Univerzita Pardubice

Fakulta ekonomicko-správní

Predikce popularity videí na sociálních sítích

Diplomová práce

2025 Bc. Tomáš Valášek

Univerzita Pardubice
Fakulta ekonomicko-správní
Akademický rok: 2024/2025

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Tomáš Valášek**
Osobní číslo:	**E23057**
Studijní program:	**N0613A140041 Aplikovaná informatika – Data Science pro business**
Téma práce:	**Predikce popularity videí na sociálních sítích**
Zadávající katedra:	**Ústav systémového inženýrství a informatiky**

Zásady pro vypracování

Cílem práce je shrnutí metod predikce popularity videí na sociálních sítích, analýza trendů videí na sociálních sítích, sběr a zpracování získaných dat, návrh modelu predikce popularity videí na vybrané sociální platformě a zhodnocení výsledků.
Osnova:

- Metody predikce popularity videí na sociálních sítích
- Přístupy k analýze trendů videí na sociálních sítích
- Sběr dat a predikce popularity videí na sociální platformě
- Zhodnocení výsledků predikce

Rozsah pracovní zprávy: **cca 50 stran**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

BARKER, Melissa S.; BARKER, Donald; BORMANN, Nicholas F.; ROBERTS, Mary Lou a ZAHAY, Debra L. *Social media marketing: a strategic approach*. Second edition. Boston: Cengage Learning, 2017. ISBN 978-1305502758.
CHEN, Guandan; KONG, Qingchao; XU, Nan; MAO, Wenji. NPP: A neural popularity prediction model for social media content. *Neurocomputing*, 2019, 333: 221-230.
CHEN, Yen-Liang; CHANG, Chia-Ling. Early prediction of the future popularity of uploaded videos. *Expert Systems with Applications*, 2019, 133: 59-74.
LI, Chenyu; LIU, Jun; OUYANG, Shuxin. Characterizing and predicting the popularity of online videos. *IEEE Access*, 2016, 4: 1630-1641.
ROUSIDIS, Dimitrios; KOUKARAS, Paraskevas; TJORTJIS, Christos. Social media prediction: a literature review. *Multimedia Tools and Applications*, 2020, 79.9: 6279-6311.

Vedoucí diplomové práce: **prof. Ing. Petr Hájek, Ph.D.**
Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce: **1. dubna 2024**
Termín odevzdání diplomové práce: **30. dubna 2025**

L.S.

prof. Ing. Jan Stejskal, Ph.D. v.r.
děkan

prof. Ing. Petr Hájek, Ph.D. v.r.
garant studijního programu

V Pardubicích dne 1. dubna 2024

PROHLÁŠENÍ AUTORA

Práci s názvem Predikce popularity videí na sociálních sítích jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 30. 4. 2025 Tomáš Valášek v. r.

PODĚKOVÁNÍ:

Rád bych tímto poděkoval panu prof. Ing. Petru Hájkovi, Ph.D. za odborné konzultace, ochotu a trpělivost v průběhu zpracování této diplomové práce.

ANOTACE

Práce je věnována predikci popularity videí na sociálních sítích s využitím analytických metod a strojového učení. Sleduje a popisuje klíčové faktory ovlivňující potenciál růstu sledovanosti videa, věnuje se tvorbě vhodného predikčního modelu a jeho vyhodnocení.

KLÍČOVÁ SLOVA

Strojové učení, predikce, analytika, sociální sítě, videotvorba, popularita, YouTube, Python

TITLE

Prediction of video popularity on social media

ANNOTATION

The thesis focuses on predicting the popularity of videos on social media using analytical methods and machine learning. It examines and describes key factors influencing the potential growth of video viewership, develops an appropriate predictive model, and evaluates its performance.

KEYWORDS

Machine learning, prediction, analytics, social media, video creation, popularity, YouTube, Python

OBSAH

SEZNAM OBRÁZKŮ A TABULEK

SEZNAM ZKRATEK

API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
BERT	Bidirectional Encoder Representations from Transformers
CSV	Comma Separated Values
CTR	Click-Through Rate
GRU	Gated Recurrent Unit
ID	Identifikátor
IVR	Impression-to-View Rate
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NLP	Natural Language Processing
R^2	Koeficient determinace
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
SHAP	SHapley Additive exPlanations
SVR	Support Vector Regression
TF-IDF	Term Frequency – Inverse Document Frequency
XGBoost	Extreme Gradient Boosting

ÚVOD

Sociální sítě jsou stále rostoucím fenoménem, který využívá většina dnešního světa. Za to může především jednoduchá možnost komunikovat s lidmi nebo sdílet či konzumovat různorodý obsah. Mezi ten patří i videotvorba, která se poslední roky stává dominantní součástí i těch sociálních sítí, které se původně na videoobsah nezaměřovaly. Sociální sítě jako YouTube, Facebook, Instagram nebo TikTok dnes generují miliardy zhlédnutí, které díky monetizaci a propagační síle umožňují velmi dobrou obživu úspěšným tvůrcům a společnostem. Z tohoto důvodu je schopnost odhadnout potenciál videa klíčové nejen pro tvůrce videí při plánování obsahu, ale i pro firmy investující do online reklamy.

Predikce popularity videí se stala významným tématem datové analytiky, která se pomocí zejména metod strojového učení snaží zjistit, jakými faktory je tato popularita ovlivněna a dokázat na jejich základě co nejpřesněji odhadnout úspěšnost nových videí. Mezi nejčastěji sledované faktory patří například název a délka videa, doba zveřejnění nebo aktivita diváků. Úspěch na sociální sítích je ovšem složitý a jeho predikce je velkou výzvou, jelikož je ovlivňován i vnějšími faktory, s kterými často nelze počítat.

Cílem této práce je proto shrnout metody použitelné k predikci popularity videí, identifikovat klíčové atributy ovlivňující růst počtu zhlédnutí videa pro platformu YouTube a navrhnutí vlastního predikčního modelu, který dokáže úspěšnost vybraného videa odhadnout.

Za účelem dosažení tohoto cíle bude nejprve představen teoretický rámec spojený se sociálními sítěmi, faktory ovlivňujícími popularitu videí a metodami predikce. Následně bude provedena analýza vlastních reálných dat platformy YouTube a pomocí několika predikčních modelů bude ověřena jejich účinnost při odhadu počtu zhlédnutí. Nejlepší model bude následně navrhnut k implementaci v praxi.

1 SOCIÁLNÍ SÍTĚ A VIDEOOBSAH

Tato kapitola je zaměřena na seznámení s vývojem sociálních sítí a jak na nich všechen videoobsah šíří. Sociální sítí je jakákoliv internetová služba, která umožňuje svým zaregistrovaným členům mít svůj veřejný, či pouze částečně veřejný profil a navazovat skrz něj vztahy s uživateli na stejné síti [1].

V posledních letech popularita sociálních sítí po celém světě vysoce vzrostla. Podle údajů z roku 2024 činí počet uživatelů sociálních sítí po celém světě přibližně 5,2 miliardy lidí. [2] V České republice je zhruba 7,2 milionu uživatelů sociálních sítí a očekává se, že do roku 2029 číslo vzroste na téměř 8,7 milionu [3].

Lidé na sociálních sítích komunikují a sdílí obsah, který chtějí. Mezi takový obsah patří jejich vlastní myšlenky, ale i články, fotografie nebo videa. Videotvorba se stala nedílnou součástí digitálního světa a pro mnoho tvůrců představuje významné generování zisku. Úspěšní tvůrci často ukazují své vysoké příjmy, což motivuje další lidi k pokusu o úspěch na těchto platformách.

1.1 Vývoj videotvorby na internetu

Videotvorba na internetu prošla za poslední dvě desetiletí velmi razantním vývojem. S přístupem k rychlému internetu v běžné společnosti bylo umožněno sledovat videa ostatních, ale i přidávání těch vlastních.

1.1.1 První desetiletí (2000–2010)

V tomto směru byla průkopníkem sociální síť **YouTube**. Ta byla založena v roce 2005 a umožnila velmi jednoduše videa sledovat i nahrávat, ale také s nimi interagovat pomocí tlačítka „To se mi líbí“, komentářů a sdílení na dalších platformách. YouTube se tak stal centrem videotvorby na internetu [4].

YouTube je hybridní mediální platforma, kde dochází k prolínání participativní kultury běžných uživatelů s komerčními zájmy velkých mediálních subjektů [5]. V roce 2007 YouTube udělal zásadní krok pro svoji platformu, když zavedl YouTube Partner Program, umožňuje tvůrcům generovat příjmy z obsahu, který vytvářejí, a zároveň poskytuje nástroje pro ochranu autorských práv, jako je systém Content ID. To velmi motivovalo tvůrce k pravidelné tvorbě kvalitnějších videí [6].

Algoritmus YouTube se v této době zaměřoval primárně na počet zhlédnutí jako klíčový ukazatel úspěšnosti videa, což postupem času vedlo k růstu clickbaitových titulků a trendu virálních videí [4].

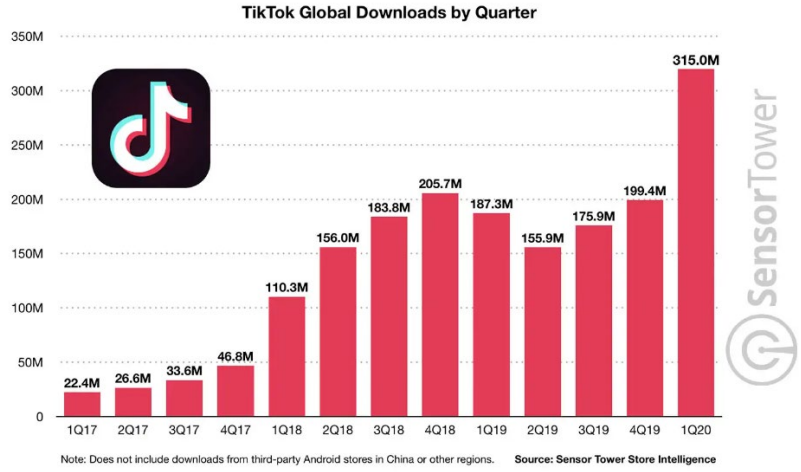
1.1.2 Druhé desetiletí (2010–2020)

Druhé desetiletí 21. století bylo charakterizováno výrazným rozvojem videotvorby a její profesionalizací na digitálních platformách. YouTube se profesionalizoval a začal být vnímán jako reálná kariérní příležitost. Přelom nastal kolem roku 2013, kdy populární tvůrci jako PewDiePie začali veřejně sdílet své příjmy z YouTube [7]. Tento trend přilákal nové autory a rozšířil videotvorbu i na další platformy. Instagram v roce 2013 zavedl krátká videa jako konkurenci Vine, později přidal Stories a Facebook automaticky přehrávaná videa ve feedu. Tato inovace vedla k růstu tzv**. influencer marketingu** – spolupráce firem s tvůrci na základě jejich videoobsahu a autenticity.

1.1.3 Třetí desetiletí (2020+)

Pandemie COVID-19, která začala 31. prosince 2019, ovlivnila nejen běžný život kolem nás, ale i digitální prostředí. Podle údajů společnosti Statista dosáhl v roce 2020 celkový počet uživatelů sociálních médií 4,2 miliardy, což představuje o 490 milionů uživatelů vyšší nárůst oproti roku 2019. Pandemie tak výrazně urychlila digitalizaci, zintenzivnila aktivitu stávajících uživatelů sociálních sítí a přivedla na ně i ty, kteří je dosud téměř nevyužívali [2].

Platforma TikTok, zaměřená na krátká videa, zaznamenala rekordní růst (Obrázek 1). Na tento trend reagoval Instagram (Reels) a YouTube (Shorts). Studie **Shorts on the Rise: Assessing the Effects of YouTube Shorts on Long-Form Video Content** z roku 2024 ukazuje, že Shorts vedly k poklesu sledovanosti dlouhých videí na stejných kanálech [10]. To signalizuje proměnu uživatelského chování a nutnost přizpůsobit obsah novým preferencím. Trend zkracování formátu ovlivňuje tvorbu i doporučovací algoritmy a mění pojetí popularity videí na všech hlavních platformách.



Obrázek 1: Graf růstu počtu stažení platformy TikTok, 2020 **Zdroj**: [9]

1.2 Trendy & popularita

Trendy ve videoobsahu odrážejí aktuální vývoj uživatelských preferencí na sociálních sítích. V běžné mluvě se pojmem „trend“ často myslí populární téma, internetový mem nebo virální hudba, avšak v širším kontextu se ale jedná o populární způsoby sledování videí nebo vývoj algoritmu jednotlivých platforem.

Vysoká popularita videí na sociálních sítích dle studie **Characterizing and Predicting the Popularity of Online Videos** není velmi častá, většina má spíše podprůměrné zhlédnutí, pouze pár z nich vybouchne. Životnost videí je navíc také velmi nízká, a to především u těch, které se nikdy neprosadí [11].

1.2.1 Virální formáty a témata

Dlouhá videa (long-form), tedy videa s délkou přesahující několik minut, se stala populární již při prvních fázích rozvoje videoobsahu na internetu, zejména po spuštění platformy YouTube v roce 2005. Ten je dodnes nejpopulárnějším centrem long-form videí. Dlouhá videa často nabízí komplexnější ponoření do tématu, detailnější zpracování a možnost lépe zapojit diváky. S delší stopáží je mnohem větší prostor pro obsáhlejší scénář videa, což je důvodem, proč tato videa přitahují více lidí s cílem sledovat obsah s větší informační hodnotou. Na druhé straně stojí krátká videa, které mají spíše tendenci přitahovat diváky hledající rychlou zábavu.

Relevantní long-form platformy: YouTube, Vimeo, Twitch

Krátká videa (Short-form) byla na internetu přítomna i dříve, popularitu však získala později než long-form obsah, a to až v roce 2013 s nástupem platformy Vine s pouze pár sekundovými vtipnými videi. Short-form zažívá největší nárůst od roku 2020 díky platformě TikTok, která dovoluje rozsah od 3 sekund do 10 minut. Tento fenomén začaly přebírat i ostatní sociální sítě.

Důvodem oblíbenosti short-form obsahu je jeho jednoduchá konzumace. Není potřeba velká pozornost ani čas k zhlédnutí. vyhledávání nových videí navíc nezabere ani sekundu. Nová videa jsou navíc vybírána velmi pečlivým algoritmem, který se naučí preference uživatele a na jejich základě doporučuje podobný obsah. Videa jsou navíc často velmi dynamická, vizuálně poutavá nebo vtipná. Právě tyto faktory vedou také k problémům se závislostí na digitálním obsahu, jelikož TikTok zkrácením délky videí aktivně zkracuje i délku pozornosti uživatelů a v kombinaci se svým jednodušším obsahem má spíše negativní vliv na kognitivní schopnosti jedince [12].

Relevantní short-form platformy: TikTok, Instagram Reels, YouTube Shorts

První **virální formáty** byly long-form a patřilo mezi ně komentované hraní her (tzv. „Let's Play") nebo videa o módě a lifestylu. Tyto dva formáty významně přispěly k růstu sledovanosti a zapojení diváků na YouTube. Postupem času začaly být populární sociální experimenty, které podněcovaly diskusi publika, dále také pranky, jejichž cílem bylo vyvolání emoční reakce, nebo dokumentární videa, která nabízela hlubší analýzu různých témat.

S postupným rozvojem long-form videa dnes mezi ty nejpopulárnější patří rozhovory a podcasty. Výrazným virálním formátem jsou také reakční videa. Ty většinou recyklují již jiné populární video pouze s přidáním názoru člověka, který na video reagoval. Na YouTube a dalších sociálních sítích mají long-form trendy delší životní cyklus než short-form trendy, které se vyznačují rychlým střídáním populárních témat [4].

Obrázek 2: Příklad populárního Let's Play videa: PedrosGame **Zdroj**: Screenshot, YouTube

Právě tato neustálá rotace trendů přispěla k popularitě short-form videí na TikToku, Shorts, nebo Reels. Některé trendy vydrží na těchto platformách v rámci několika dní, někdy dokonce i pouhých pár hodin.

Viralita v short-form obsahu je založena na jednodušších tématech, která nevyžadují složitou produkci ani hlubší pozornost diváků. Mezi prvními populárními formáty se objevila krátká vtipná videa (sketche) a lipsync. Postupem času se přidala rychlá taneční videa, pranky s krátkou stopáží nebo internetové memy. Dnes se v short-form obsahu objevují i témata běžná v long-form videích, ale upravená tak, aby odpovídala požadavkům rychlé a krátké konzumace obsahu (například sestřihy z Let's Playe (Obrázek 2) nebo mnohem více stříhané vlogy) [4].

1.2.2 Doporučovací algoritmy

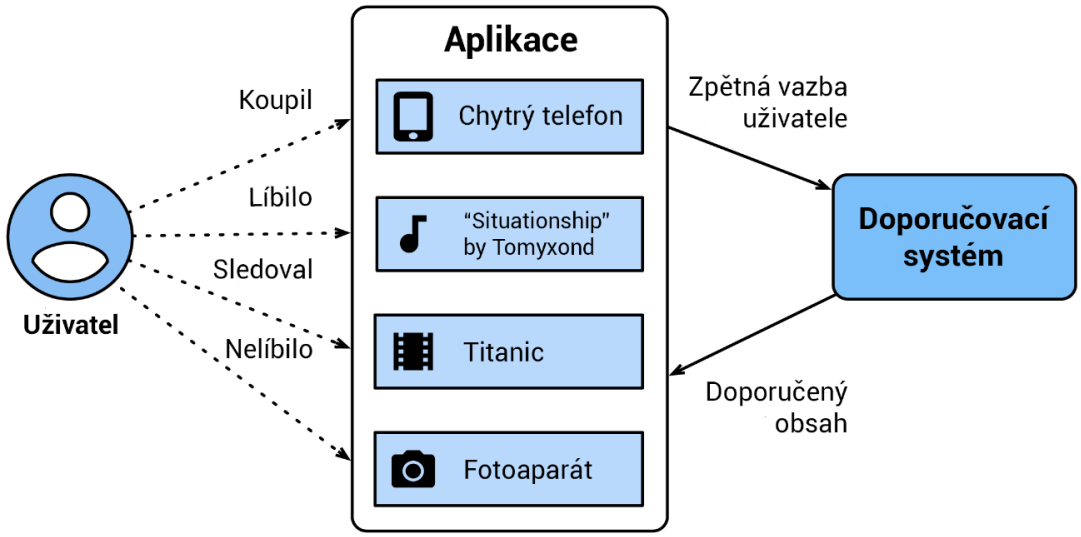
Z technického hlediska je algoritmizace stavebním kamenem všech moderních sociálních sítí včetně videoobsahu, jelikož výrazně zjednodušuje jejich funkci a umožňuje efektivní správu obrovského množství dat. Odborně jsou nazývány doporučovacími algoritmy (Obrázek 3), které analyzují chování uživatelů a optimalizují distribuci obsahu s cílem zajistit co nejrelevantnější zážitek pro každého jednotlivce [13].

Studie **NPP: A Neural Popularity Prediction Model for Social Media Content** uvádí, že doporučovací algoritmy hrají klíčovou roli při určování popularity obsahu na sociálních sítích [13].

Mezi hlavní aspekty, které tyto algoritmy analyzují, patří:

- Historie interakcí (To se mi líbí, sdílení, komentáře atd.)
- Obsah příspěvků (Klíčová slova, vizuální prvky)
- Časové faktory (Čas zveřejnění, rychlost získávání interakcí)

Předpověď popularity obsahu je stále výzvou, jelikož algoritmy musí kombinovat různé zdroje dat a adaptovat se na měnící se vzorce chování uživatelů.



Obrázek 3: Zjednodušená vizualizace procesu doporučování **Zdroj**: upraveno dle [13]

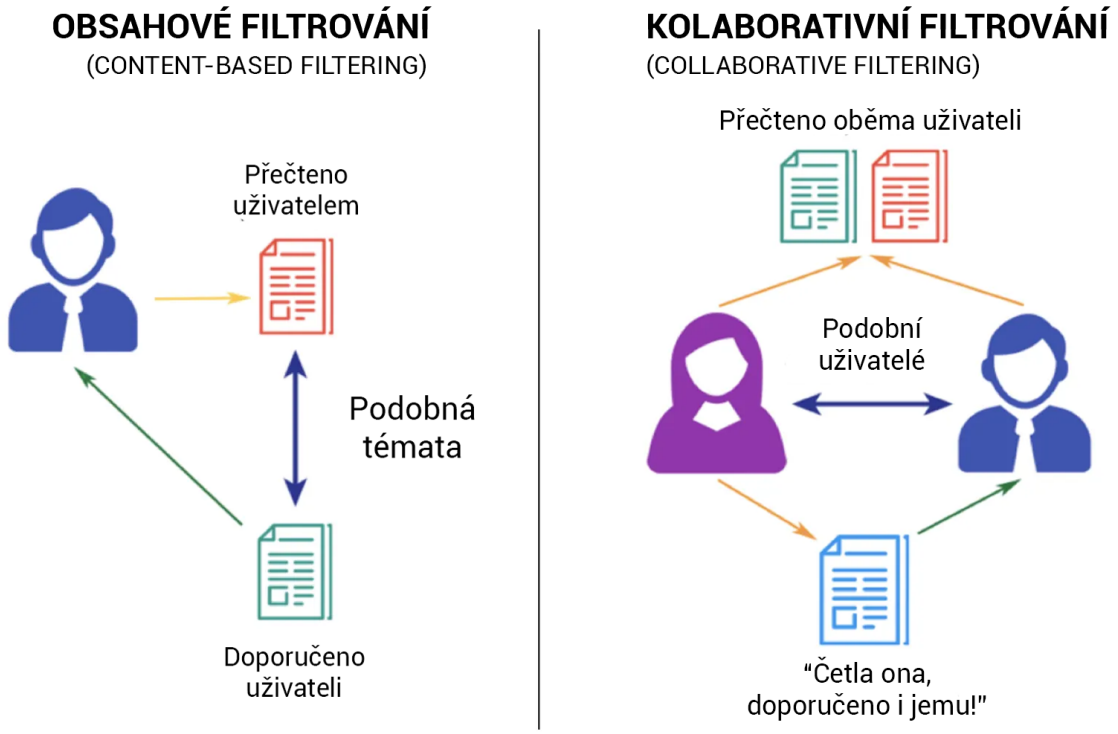
Sociální sítě se navíc snaží nejen zobrazovat relevantní obsah, ale také maximalizovat dobu, po kterou diváci dávají pozornost. Tento cíl vedl k vývoji pokročilejších modelů strojového učení, které *„nejen okamžitě přizpůsobují obsah na základě uživatelských preferencí, ale zároveň se učí z historických vzorců chování a predikují budoucí interakce“* [13].

Doporučovací algoritmy jsou rozdělovány do tří hlavních kategorií, a těmi jsou obsahové filtrování, kolaborativní filtrování a hybridní metody. Každý z těchto přístupů má své výhody a nevýhody, a volba vhodné metody závisí na kontextu a dostupných datech [15].

Obsahové filtrování (Content-Based Filtering) doporučuje položky na základě jejich vlastností, například klíčových slov nebo metadat, a hledá podobnosti mezi obsahem, který uživatel již preferoval. U filmů lze analyzovat žánr, režiséra či herecké obsazení. Často se využívají techniky jako TF-IDF (Term Frequency – Inverse Document Frequency) nebo vektorové reprezentace textu [15].

Kolaborativní filtrování (Collaborative Filtering) vychází z předpokladu, že diváci s podobnými preferencemi budou mít v budoucnu podobné zájmy. Tento přístup se dále dělí na **paměťové (Memory-based) metody**, které využívají historická data o interakcích bez nutnosti složitých modelů. Často se používá k-nejbližších sousedů (k-NN), který hledá podobné uživatele a doporučuje položky na základě jejich hodnocení. Druhá metoda se nazývá **modelová (Model-based)**, ta aplikuje strojové učení ke škálování doporučení.

Rozdíl mezi obsahovým a kolaborativním filtrováním je zjednodušeně ilustrován na následujícím Obrázku 4.



Obrázek 4: Obsahové vs. kolaborativní filtrování

Zdroj: upraveno dle [15]

1.2.3 Engagement metriky

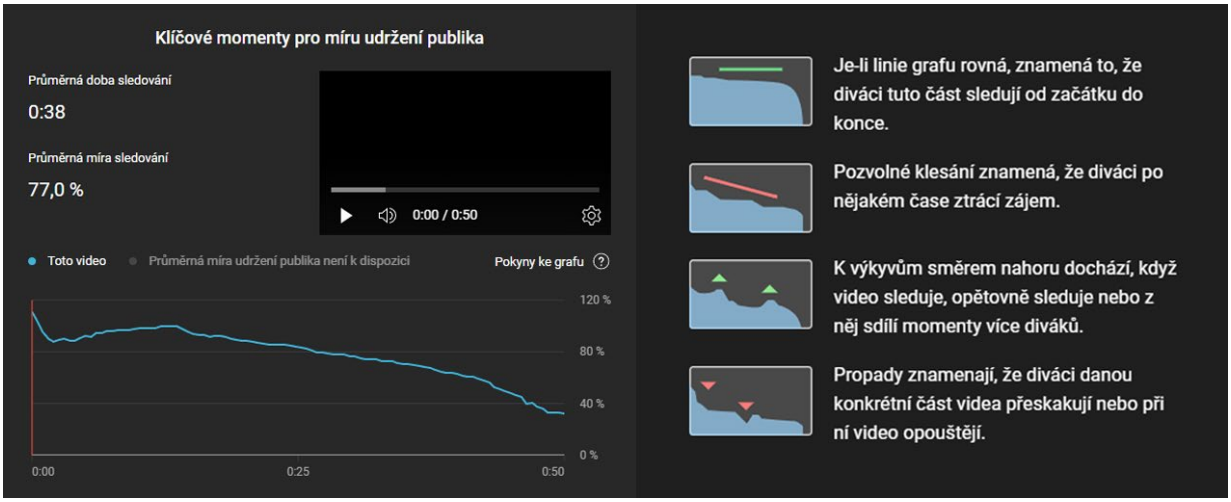
Pro hodnocení úspěchu videa na sociální síti jsou využívaný tzv. engagement metriky. Jedná se o ukazatele, které měří interakci uživatelů sociální sítě s videem. Tyto metriky představují pokročilejší způsob hodnocení výkonu videa oproti samotnému počtu zhlédnutí. Na rozdíl od tradičního měření popularity na základě zhlédnutí poskytují engagement metriky hlubší vhled do chování diváků [16].

Metriky sledovanosti

Počet zhlédnutí již dnes není tolik relevantní metrikou, zastoupil ho ale mnohem důležitější **Watch Time**, což je celkový čas, který diváci stráví sledováním videa. Tato metrika pomáhá zjistit, jak je video poutavé a zda u něj diváci vydrží až do konce [16].

Pro všechny sociální sítě je velmi důležitou metrikou **Retention Rate**, tedy udržení diváků. Retention Rate měří podíl průměrné délky sledování vůči celkové délce videa (v procentech).

Vysoké hodnoty u obou metrik naznačují, že diváci video považují za dostatečně zajímavé na to, aby u nich trávili svůj čas. Vizualizace Retention Rate u videa na YouTube a jednoduché vysvětlení, co různé vývoje v grafu znamenají, je vidět na následujícím Obrázku 5 [17].



Obrázek 5: Retention graf a vysvětlivky **Zdroj**: Screenshot, YouTube

Za pomocí metriky Watch Time je získán další důležitý ukazatel **Average View Duration** tedy průměrná délka sledování, která určuje, jak dlouho diváci video v průměru sledují. To stejně jako Watch Time nebo Retention Rate umožňuje zjistit, zda je video dostatečně zajímavé, aby udrželo pozornost diváků [17].

Důležitou komponentou při měření úspěšnosti videa je, kolikrát bylo video dokoukáno. K tomu slouží **Completion Rate**, tedy procento diváků, kteří se při sledování videa dostali až na konec. Vysoké procento algoritmům naznačuje, že je video natolik zajímavé, aby ho divák sledoval po celý čas [17].

Metriky interakce

Počet kladných hodnocení ve formě tlačítka „To se mi líbí“, komentářů, sdílení a další formy reakcí na video. To jsou ty nejznámější metriky pro běžné uživatele známé také jako interakce, které ale i v hlubší analytice sledují a určují podle nich potenciál daného videa.

Engagement Rate je jednou z klíčových metrik interakce a její výpočet může být přizpůsoben podle konkrétní analytické potřeby. Tím nejvíce využívaným je celkový součet těchto interakcí vydělen celkovým počtem zhlédnutí a následně vynásoben stokrát pro získání procentuálního výsledku. Dalšími možnostmi je měření na základě dosahů videa nebo počtu sledujících účtu, který video nahrál. Vzorec zůstává stejný, počet interakcí je pouze vydělen zmíněnými metrikami [17].

Platformy jako TikTok a YouTube Shorts upřednostňují videa, která mají vysokou míru sdílení, jelikož to značí, že jejich obsah je zajímavý a stojí za to jej doporučit dalším uživatelům. K tomu je využívána metrika interakce **Share Rate**. Její běžnou variantou je podíl sdílení na zhlédnutích. Čím vyšší Share Rate, tím vyšší pravděpodobnost, že se video stane virálním. Například YouTube Shorts videa s vysokým Share Rate posílá do doporučených videí na hlavní stránku.

Metriky kvality distribuce

Míra prokliků, tedy **Click-Through Rate (CTR)**, je dalším důležitým ukazatelem výkonu videa. CTR vyjadřuje poměr mezi počtem kliknutí na video a celkovým počtem jeho zobrazení v různých distribučních kanálech platformy. Vysoké CTR dává najevo, že náhledový obrázek a název videa úspěšně upoutaly pozornost diváků a přiměly je ke kliknutí. Z tohoto důvodu je optimalizace náhledu a názvu videa klíčovou součástí tvorby videí.

Autoři často využívají vizuálně kontrastní miniatury a lákavé, někdy až zavádějící názvy k maximalizaci proklikovosti u jejich videa. Tato metrika ovšem vede k zneužívání a vytváření klamných názvů nebo náhledů, tedy clickbaitů.

Druhou klíčovou metrikou kvality distribuce je **Impression-to-View Rate (IVR)**, která se od CTR liší svým zaměřením na odlišné aspekty diváckého chování. Zatímco CTR měří atraktivitu náhledového obrázku a názvu videa, tedy kolik uživatelů na video klikne v poměru k počtu jeho zobrazení, IVR vyjadřuje podíl uživatelů, kteří si video skutečně pustili poté, co jim bylo zobrazeno (Tabulka 1).

Tato metrika se nejčastěji používá v prostředí, kde se videa automaticky přehrávají, například u krátkého vertikálního obsahu na TikToku, YouTube Shorts nebo Instagram Reels, kde náhledy a názvy hrají menší roli. Vysoká IVR hodnota signalizuje, že video dokáže upoutat pozornost i bez nutnosti aktivního kliknutí ze strany diváků.

Vzorce engagement metrik

Tabulka 1: Vzorce engagement metrik **Zdroj**: [18]

Název metriky	Vzorec
Watch Time	$\sum(\text{Délka sledování jednotlivých diváků})$
Retention Rate (%)	$\frac{\text{Průměrná délka sledování}}{\text{Celková délka videa}} \times 100$
Average View Duration	$\frac{\text{Průměrná délka sledování}}{\text{Celková délka videa}}$
Completion Rate (%)	$\frac{\text{Počet diváků, kteří video dokoukali}}{\text{Celkový počet zobrazení}} \times 100$
Engagement Rate (%)	$\frac{\text{Celkový počet interakcí}}{\text{Celkový počet zobrazení}} \times 100$
Share Rate (%)	$\frac{\text{Celkový počet sdílení}}{\text{Celkový počet zobrazení}} \times 100$
Click-Through Rate (%)	$\frac{\text{Celkový počet kliknutí}}{\text{Celkový počet zobrazení}} \times 100$
Impression-to-View Rate (%)	$\frac{\text{Celkový počet zhlédnutí}}{\text{Počet impresí}} \times 100$

2 METODY PREDIKCE POPULARITY

Jedním z hlavních výzkumných směrů v oblasti analýzy sociálních sítí je definitivně predikce popularity. V prostředí, kde je denně nahráno ohromné množství videoobsahu, je efektivní identifikace potenciálně úspěšných videí nezbytnou schopností platformy. Tyto predikce hrají zásadní roli v optimalizaci doporučovacích algoritmů, které je využívají k tomu, aby upřednostnily a distribuovaly videa s vyšším potenciálem popularity mezi své uživatele [19].

Predikce popularity videí se opírá o širokou škálu metod a modelů, které kombinují různé faktory – od historických vzorců engagement metrik až po obsahové rysy videa či časové aspekty zveřejnění. Výzkumy ukazují, že neexistuje univerzální přístup k predikci, nýbrž to, že efektivní modely využívají kombinaci několika přístupů, a to tradičních statistických metod, strojového učení nebo hlubokých neuronových sítí [19].

Metody predikce jsou obecně rozdělovány na supervizované a nesupervizované, v českém znění učení „s učitelem“ a „bez učitele“. **Supervizované (s učitelem) metody** vycházejí z historických dat, ve kterých jsou známy jak vstupní proměnné, tak odpovídající výstupy. Modely se učí na těchto datech a následně odhadují hodnoty pro nová data [20]. Naopak **nesupervizované (bez učitele) metody** nepracují s předem definovanými výstupy, ale hledají skryté struktury a vzory v datech [20].

Vedle těchto dvou hlavních kategorií existují také **hybridní metody**, které kombinují prvky obou přístupů. Typicky jsou dle literatury o hlubokém učení využívány nesupervizované metody ke zpracování dat, například k identifikaci vzorců nebo segmentaci, a následně aplikují supervizované metody k přesnější predikci [40].

Vedle faktorů souvisejících s chováním diváků hraje v predikci popularity zásadní roli také dynamika šíření obsahu mezi uživateli. Není to jen o tom, jestli je obsah zajímavý, ale také o tom, jak dobře ho lidé sdílí nebo na něj reagují influenceři. Výzkumy ukazují, že právě dynamika šíření může rozhodnout o tom, zda video získá tisíce, nebo miliony zhlédnutí [17]. Proto dnes nejúspěšnější modely kombinují analýzu uživatelského engagementu se síťovou analýzou a metodami strojového učení, aby co nejpřesněji předpověděly, jaký obsah bude trendovat [19].

2.1 Vstupní proměnné predikci popularity

Před samotnou aplikací predikčních metod je klíčové být obeznámen o tom, jaké typy proměnných jsou běžně využívány. Všechny vycházejí z konkrétních atributů, které popisují video, jeho šíření nebo chování uživatelů. Aby bylo možné vytvořit predikční model s dobrým výkonem, je důležité porozumět, jaké typy dat se k tomuto účelu běžně využívají a co přesně v praxi reprezentují.

2.1.1 Číselné a kategoriální proměnné

Číselné proměnné jsou nejčastěji využívanou kategorií. Obsahují kvantitativní data, se kterými lze přímo počítat. Patří sem například počet zhlédnutí, délka sledování (Watch Time), míra prokliků (CTR), počet sdílení nebo engagement rate.

Kategoriální proměnné pro změnu zachycují informace kvalitativní. Mezi ty patří například žánr videa, jazyk, kategorie obsahu nebo typ zařízení, ze kterého bylo video sledováno. Pro jejich využití v modelech je často potřeba provést transformaci na binární nebo číselnou podobu.

Všechny tyto proměnné jsou tvořeny různými typy dat. Každý typ dat přináší specifické vlastnosti a určuje, jakými analytickými metodami s nimi lze dále pracovat. Číselná data lze dále rozlišit na **spojitá**, která mohou nabývat jakýchkoliv hodnot v určitém intervalu, a **diskrétní**, která jsou vždy celá čísla. Kategoriální data rozlišujeme na **nominální**, která označují různé třídy bez přirozeného pořadí a **ordinální**, u kterých lze jednotlivé kategorie uspořádat [21]. Kategoriální data se často rozdělují na binární proměnné pomocí metody „**one-hot encoding**“ [22].

Tabulka 2 uvádí přehled těchto základních typů dat, jejich charakteristiky, podtypy a konkrétní příklady proměnných využitelných při predikci popularity.

Tabulka 2: Typy číselných a kategoriálních dat **Zdroj**: [19]

Typ dat	Charakteristika	Podtyp	Příklady
Číselná	Kvantitativní hodnoty, umožňují výpočty	Spojité	Watch Time, CTR
		Diskrétní	Počet zhlédnutí, komentářů nebo sdílení
Kategoriální	Kvalitativní hodnoty, nelze sčítat	Nominální	Žánr, typ, název
		Ordinální	Hodnocení (1 až 5 hvězd)

2.1.2 Časové řady

Jedná se o kategoriální i číselná data, které neslouží pouze k popisu aktuálního stavu, ale umožňují analyzovat, jak se hodnoty mění v čase a zda vykazuje určité vzory, trendy nebo sezónnost. Využívají se především v modelech časových řad nebo rekurentních neuronových sítích.

V případě predikce popularity zachycují vývoj sledovanosti, interakcí nebo jiných metrik. Jedná se například o datum zveřejnění videa, denní přírůstky počtu zhlédnutí, tempo nárůstu engagementu nebo aktivitu uživatelů v jednotlivých časových obdobích. To dokáže pomoct zejména v studiích, které se snaží popularitu predikovat na základě vývoje dat v prvních hodinách po vydání [23].

2.1.3 Textové prvky

Tyto proměnné nejsou primárně určeny k měření v kvantitativní formě, ale často poskytují cenné informace o tématu, tónu a záměru textu. Mezi textové proměnné v predikci popularity patří například název videa, popis, komentáře nebo hashtagy. Pomáhají tak například odhalit, jaká témata nebo výrazy přispívají k vyšší míře sdílení a interakce. Přestože nejde o číselná data, lze je převést do numerické podoby pomocí technik zpracování přirozeného jazyka a dále využít v predikčních modelech.

Pro zpracování takových textových dat se v oblasti datové analytiky běžně využívají různé metody. V rámci **základní textové statistiky** lze pracovat s jednoduchými numerickými charakteristikami textu, jako je například délka, počet slov, výskyt interpunkčních znaků nebo poměr velkých písmen. Tyto údaje mohou sloužit jako indikátory tónu, naléhavosti nebo struktury sdělení.

Jednou z pokročilejších metod je **TF-IDF,** která dává jednotlivým slovům váhu podle jejich četnosti v konkrétním dokumentu a zároveň jejich vzácnosti v celém korpusu. Tímto způsobem lze identifikovat klíčová slova, která jsou pro daný text specifická a informativní [24]. Často nachází své využití v obsahovém filtrování (Kapitola 1.2.2).

Sofistikovanější přístup představují **jazykové modely typu BERT (Bidirectional Encoder Representations from Transformers)**. Tyto modely jsou trénovány na rozsáhlých textových datech a dokáží zachytit význam slov v kontextu celého sdělení. Díky obousměrnému čtení textu (zleva doprava i zprava doleva) poskytují detailnější porozumění významu jednotlivých vět a umožňují využití například pro úlohy analýzy sentimentu či rozpoznávání významu [25].

2.1.4 Vizuální prvky

Vizuální prvky hrají v online prostředí zásadní roli při rozhodování uživatelů o tom, zda si daný obsah zobrazí, otevře nebo s ním interaguje. V kontextu videoobsahu na platformách, jako je YouTube, patří mezi hlavní vizuální faktory například náhledový obrázek (thumbnail), barevná paleta, kontrast, lidské tváře, výrazné prvky nebo text v obrázku. Tyto prvky mohou přímo ovlivnit popularitu videa.

Z hlediska datové analýzy lze vizuální prvky převést do numerické podoby prostřednictvím metod počítačového vidění. Jedním z nejpoužívanějších přístupů je využití **konvolučních neuronových sítí (CNN – Convulational Neural Networks)**, které jsou schopny automaticky extrahovat významné vizuální znaky z obrazového vstupu [40]. Tyto modely dokážou detekovat například tváře, texty v obrázcích, objekty nebo dominantní barevná schémata. Výstupem může být například přítomnost konkrétních prvků (např. člověka v záběru), úroveň kontrastu nebo barevný tón, které se poté využívají jako prediktory v modelech popularity. Výhodou těchto metod je schopnost zachytit i jemné vizuální nuance, které mohou mít vliv na rozhodování uživatelů.

2.2 Tradiční statistické metody predikce popularity

Mezi nejčastěji používané tradiční statistické metody patří základní regresní modely, a to především **vícenásobná lineární regrese** [31], které modelují vztah mezi historickými daty a budoucí popularitou. Využívané jsou i modely využívající **časové řady** jako model ARIMA [32]. Ačkoliv nejde o predikční model, tak sem patří i **korelační analýza**, kterou je zjišťováno, které proměnné spolu nejvíce korelují [26].

Tyto modely jsou ve většině supervizované a díky své jednoduchosti jsou tyto metody oblíbené pro základní predikční úlohy, avšak mají omezenou schopnost identifikovat složitější vzory v datech. Jak ukazují studie, kde bylo využito i strojového učení, tradiční statistické modely nejsou vždy dostatečně přesné při práci s nelineárními vztahy mezi proměnnými a často ignorují síťové efekty, které hrají klíčovou roli v šíření virálního obsahu [19].

2.2.1 Korelační analýza

Pro zjišťování vztahů mezi proměnnými v datech se využívá korelační analýzy. V oblasti predikce popularity videí je totiž potřeba pochopit, jak spolu jednotlivé metriky souvisí. Nejčastěji používanou metodou pro měření síly těchto vztahů bývá **Pearsonova korelační analýza**, která sleduje lineární závislost mezi dvěma spojitými proměnnými [26]. V mnoha případech však tyto vztahy nejsou lineární, což je typické právě u dat ze sociálních sítí.

V takových případech se používá **Spearmanova korelační analýza**, kterou využila například studie **Using Visual Features and Early Views to Classify the Popularity of Facebook Videos** k zjištění korelací vizuálních rysů s popularitou videa [27]**.** Spearman pracuje s pořadím hodnot namísto jejich skutečných číselných hodnot, tedy porovnává, zda vyšší hodnota jedné proměnné odpovídá vyšší (nebo nižší) hodnotě druhé proměnné v rámci jejich vzájemného pořadí. Díky tomu dokáže lépe zachytit monotónní vztahy, které nejsou nutně lineární [28].

Výsledek výpočtu se pohybuje v rozmezí 0 až 1 a určuje, jak silná korelace mezi proměnnými panuje. Pokud je v tomto případě hodnota vysoká (například **r** = 0.85), znamená to, že videa, která jsou často sdílena, mají zároveň vyšší počet zhlédnutí. Negativní hodnota (například **r** = -0.85) značí inverzní vztah mezi proměnnými – pokud jedna roste, druhá klesá. Pokud je hodnota korelace nulová (**r** = 0), znamená to, že mezi proměnnými neexistuje žádný významný vztah.

Obecná interpretace je v následující Tabulce 3.

Tabulka 3: Interpretace korelací **Zdroj**: [27]

r	Síla vztahu	Interpretace
> 0.6	Silná pozitivní korelace	Růst jedné proměnné je silně spojen s růstem druhé.
⟨0.2, 0.6⟩	Střední pozitivní korelace	Růst jedné proměnné mírně doprovází růst druhé.
⟨-0.2, 0.2⟩	Slabá nebo žádná korelace	Vztah mezi proměnnými je minimální nebo neexistuje.
⟨-0.6, -0.2⟩	Střední negativní korelace	Růst jedné proměnné bývá spojen s poklesem druhé.
< -0.6	Silná negativní korelace	Růst jedné proměnné silně snižuje hodnotu druhé.

2.2.2 Vícenásobná lineární regrese

Základní regresní modely patří mezi běžné nástroje pro predikci popularity online obsahu. Sledují a hodnotí vztahy mezi nezávislými proměnnými (prediktory), jako jsou engagement metriky, časové faktory a charakteristiky obsahu, a závislými proměnnými, jako je počet zobrazení, sdílení nebo poměr „líbí/nelíbí“. Z tohoto důvodu se dají klasifikovat jako supervizované metody.

Vícenásobná lineární regrese (Multiple Linear Regression – MLR) je jedním z nejběžnějších modelů používaných k predikci popularity obsahu na sociálních sítích. Tento model je založen na základní lineární regrese, ale dokáže určit vliv více jednotlivých metrik najednou na popularitu videa a kvantifikovat jejich vzájemné vztahy. V dnešním světě datové analytiky se používá spíše jako „**baseline**“ model – tedy základní model, který následně slouží k porovnání výkonu s těmi složitějšími.

Matematický vzorec vícenásobné lineární regrese:

$$\mathbf{Y} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1}\mathbf{X_1} + \boldsymbol{\beta_2}\mathbf{X_2} + \cdots \boldsymbol{\beta_n}\mathbf{X_n} + \boldsymbol{\varepsilon}$$

V tomto vzorci je $\mathbf{Y}$ hodnota, kterou model předpovídá, tedy například počet zhlédnutí videa. $\mathbf{X_n}$ jsou jednotlivé vstupní faktory, které model bere v úvahu, například engagement metriky, čas zveřejnění videa nebo jiné prvky, které mohou ovlivnit jeho popularitu. $\boldsymbol{\beta_0}$ je tzv. intercept, což znamená, že představuje základní hodnotu $\mathbf{Y}$, když všechny ostatní faktory $\mathbf{X_n}$ mají nulovou hodnotu. $\boldsymbol{\beta_1}$ ... $\boldsymbol{\beta_n}$ jsou koeficienty, které určují, jak moc každý z faktorů ovlivňuje výsledek – tedy o kolik se $\mathbf{Y}$ zvýší nebo sníží, pokud se příslušný $\mathbf{X_n}$ změní o jednu jednotku, přičemž ostatní proměnné zůstávají stejné. $\boldsymbol{\varepsilon}$ je chybový člen, který zachycuje vše, co model nevysvětlí – tedy vlivy, které v datech nejsou zahrnuté nebo jsou čistě náhodné [30].

Studie z roku 2019 zkoumala efektivitu vícenásobné lineární regrese při predikci popularity videí na platformě YouTube v porovnání se silnějšími modely [31]. Ačkoliv výsledek pokročilých metod byl lepší, tak výzkum i pomocí této metody ukázal, že kombinace Watch Time, CTR a míry sdílení je dobrým prediktorem celkového počtu zhlédnutí. Tím se dokázalo, že i jednoduchý model dokáže poskytnout užitečné predikce, zejména v základních analýzách. Hlavní nevýhoda MLR ale spočívá v tom, že předpokládá lineární vztah mezi proměnnými, což není vždy realistické.

2.2.3 Časové řady ARIMA

Časové řady jsou data uspořádaná v čase, kde každé pozorování odpovídá určitému časovému okamžiku. V kontextu sociálních sítí jsou stejně jako v regresních metodách analyzovány metriky jako počet zhlédnutí, engagement nebo šíření obsahu, které se vyvíjejí v čase. Predikce těchto metrik umožňuje lépe porozumět dynamice popularity videí a optimalizovat strategie jejich propagace [32].

Pro zpracování predikčních časových řad se běžně používají modely **ARIMA (Autoregressive Integrated Moving Average).** Tyto modely se skládají ze tří hlavních částí, které dohromady tvoří efektivní predikční model. Tyto části se nazývají integrace, autoregrese a klouzavý průměr.

Studie **Forecasting Social Media Engagement: An ARIMA-based Approach** aplikovala tento model na predikci engagement rate na platformách Facebook a Instagram [32]. V této analýze byl model ARIMA využit k zachycení časových vzorců v datech, což umožnilo předpovědět míru zapojení uživatelů na základě dvouletých historických údajů.

2.2.4 Souhrn statistických metod

Společným problémem statistických metod jsou jejich přísné předpoklady, které nemusí odpovídat realitě sociálních sítí. Lineární regrese vyžaduje přímý vztah mezi proměnnými a modely ARIMA předpokládají stacionaritu časových řad, což v prostředí, kde trendy a algoritmy neustále mění podmínky, není vždy splněno.

Zásadní limitací je tak neschopnost efektivně pracovat s nelineárními vzory, dynamikou engagementu nebo sezónními vlivy. Tradiční modely sice umožňují analyzovat základní vztahy mezi proměnnými, ale nejsou dostatečně flexibilní, aby zachytily složitější chování uživatelů, interakce mezi metrikami nebo změny v algoritmech. Tyto navíc neumí automaticky přizpůsobit svůj výpočet, což je zásadní nevýhoda tam, kde obliba obsahu často závisí na krátkodobých virálních efektech.

Právě kvůli těmto nedostatkům se dnes stále více využívají metody strojového učení, které na rozdíl od tradičních přístupů nepotřebují předem stanovené vztahy mezi proměnnými, ale učí se je automaticky z dat. Díky tomu dokážou lépe pracovat s nelineárními vzory, reagovat na trendy a přizpůsobit se měnícím se algoritmům platforem. V další části budou představeny pokročilejší metody, které tyto limity statistických modelů překonávají a umožňují přesnější predikci popularity videí.

2.3 Strojové učení a hluboké neuronové sítě

Predikce popularity videí na sociálních sítích není díky své dynamice a neustálým změnám velmi jednoduchá. Omezenost statistických metod vede k využití sofistikovanějších metod, které se dokážou na základně získaných dat učit a objevit nové vzory, které využijí k predikci.

Strojové učení (Machine Learning) je oblastí umělé inteligence, která umožňuje modelům se učit ze zkušeností a adaptovat se na nové situace. V kontextu predikce popularity videí modely strojového učení analyzují obrovské množství engagement metrik. Existuje několik přístupů, jak k této predikci přistoupit. Tato kapitola proto shrnuje ty nejpoužívanější metody strojového učení, které jsou využívány k predikci popularity videí na sociálních sítích.

Dle provedené literární rešerše bylo zjištěno, že nejvíce využívané **supervizované metody** zahrnují pokročilé regresní modely, jako je **SVR**, který dokáže modelovat složité nelineární vztahy mezi prediktory popularity, avšak je méně vhodný pro práci s velkými objemy dat [34]. Výkonnější alternativou jsou **gradientně posilované stromy**, konkrétně **XGBoost**, který patří mezi nejpřesnější metody predikce popularity a je běžně využíván ve výzkumech i průmyslových aplikacích [39]. Další metoda, která využívá rozhodovací stromy a nabízí stabilní predikční výkon, je **Random Forest** [37].

Z **nesupervizovaných metod** mají největší význam **neuronové sítě**, především hluboké modely využívající nelineární reprezentace dat [41]. Naproti tomu shluková analýza (např. **K-means clustering**) a asociační pravidla (**Apriori algoritmus**), které se běžně používají k segmentaci nebo hledání vzorů v datech, nejsou pro přímou predikci popularity vhodné, a tak není potřeba je v této práci blíže popisovat.

V této kapitole se proto zaměříme pouze na ty metody, konkrétně na **SVR, Random Forest, XGBoost a neuronové sítě**, které se ukázaly jako nejúčinnější při predikci popularity videí na sociálních sítích.

2.3.1 Pokročilá regrese Support Vector Regression (SVR)

SVR je metoda strojového učení určená k predikci spojitých hodnot, která vychází z algoritmu **Support Vector Machines (SVM)**. Oproti běžným regresním modelům je SVR robustnější vůči extrémním hodnotám a odlehlým datům, protože ignoruje drobné chyby uvnitř tzv. **epsilon-insenzitivní zónu** (ε). Díky tomu se model zaměřuje na klíčové trendy v datech a méně podléhá šumu.

Princip modelu spočívá v hledání optimální **hyperroviny**, která s rezervou odděluje data, a přitom minimalizuje chybu. Parametr **C** určuje kompromis mezi přesností a generalizací, přičemž odchylky mimo zónu penalizují tzv. **slack proměnné**. ξ_i **a** ξ_i^* **.** V oblasti predikce online videí se často používá tzv. **Gaussian Radial Basis Function (RBF)** [34]. Tato jádrová funkce nahrazuje skalární součin v podmínkách modelu a umožňuje SVR lépe modelovat složité nelineární vztahy. Celý trénovací vzorec s aplikací jádra v podmínkách je v následující Tabulce 4.

Tabulka 4: Účelová funkce a omezující podmínky SVR **Zdroj**: Vlastní zpracování

Účelová funkce SVR:	**Omezující podmínky (+RBF jádro):**
$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$	$y_i - K(x_i, x_j) - b \le \epsilon + \xi_i$ $K(x_i, x_j) + b - y_i \le \epsilon + \xi_i^*$ $\xi_i, \xi_i^* \ge 0; i = 1, \dots, N$

Využití Je velmi dobře vidět v článku "**Predicting Popularity of Online Videos using Support Vector Regression**", kde je uvedena metoda **Popularity-SVR** [34]. Autoři využili SVR s jádrem **RBF** k predikci popularity na základě metrik jako počet zhlédnutí, engagement a vizuální charakteristiky. Trénovací dataset obsahoval 24 000 videí z YouTube a Facebooku. Studie potvrdila, že SVR dosahuje velmi dobrých výsledků zejména při predikci krátce po zveřejnění videa (např. <12 hodin). Mezi ty nejvýznamnější proměnné v tomto modelu patřily engagement metriky, CTR, Engagement Rate, zhlédnutí v prvních hodinách a rychlost nárůstu sledovanosti.

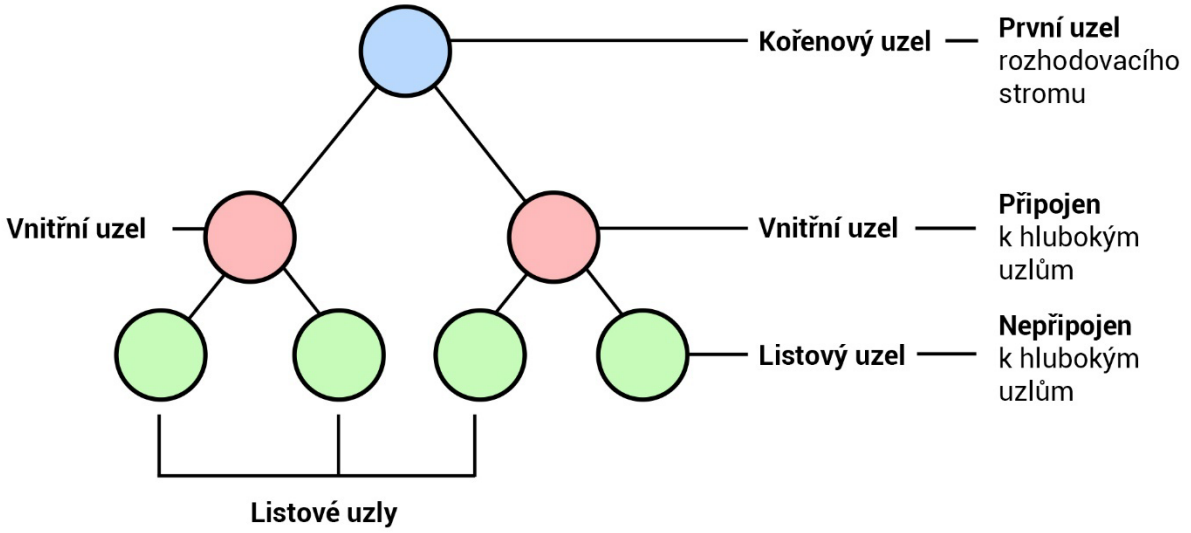
V porovnání s ostatními modely, které byly testovány, měl nejlepší výsledky predikce zejména při nižších hodnotách **tr** (<12 dní od zveřejnění videa), kde je trendování zásadní a predikce nejobtížnější [34]. Studie také zjistila, že tato metoda dominuje spíše u malých datasetů. Pro dlouhodobější predikce se proto doporučuje zvážit robustnější metody strojového učení.

2.3.2 Rozhodovací stromy a Bagging metoda Random Forest

Jednou z nejlépe interpretovatelných metod strojového učení jsou **rozhodovací stromy**, které jsou často využívány jako základní stavební blok pro pokročilejší kombinace modelů [35]. Princip rozhodovacího stromu spočívá ve vytvoření hierarchické struktury, ve které jsou jednotlivá rozhodnutí reprezentována větvením na základě hodnot vstupních proměnných.

Struktura rozhodovacího stromu připomíná skutečný strom – začíná u **kořenovým uzlem** (**root node**), který reprezentuje počáteční rozhodnutí na základě určité vlastnosti. Odtud se strom větví do **vnitřních uzlů** (**internal nodes**) podle hodnot jednotlivých vlastností, tyto vrstvící se uzly se nazývají hlubokými uzly (**deeper nodes**), které větví až k tzv. **listovým uzlům** (**leaf nodes**), které představují výsledné predikce [35]. Model se učí tak, že rozděluje trénovací data do větví na základě podmínek, které maximalizují rozdíly mezi třídami, nebo v případě regrese co nejvíce snižují chybu predikce.

Vizualizace struktury rozhodovacího stromu je na následujícím Obrázku 6.



Obrázek 6: Struktura rozhodovacích stromů **Zdroj**: Vlastní zpracování

Při konstrukci rozhodovacích stromů jsou využívány různé metody pro dělení uzlů, které hodnotí kvalitu rozdělení dat. Nejčastější je **Giniho index**, který měří míru nečistoty uzlu – čím nižší, tím lepší. Alternativou je **Information Gain**, který hodnotí zisk informace na základě snížení entropie po dělení. Pro vyvážení tohoto přístupu se využívá i **Gain Ratio**, který zohledňuje různorodost rozdělení [35]. Tyto metody pomáhají určit, podle kterého atributu strom dále větví data.

Velmi používanou metodou, která využívá rozhodovací stromy v predikci popularity, je **Random Forest**. Ta je založena na tzv. **ensemble modelování**, které spočívá v kombinování více jednoduchých modelů do jednoho složitějšího, což zvyšuje přesnost a snižuje náchylnost k chybám [36].

V tomto případě je využívána technika „**Bagging**“, která funguje na principu „**bootstrap vzorkování**“, které z původního trénovacího datasetu o velikosti **N** náhodně vybírá vzorky (s opakováním) a vytváří **B** nových datasetů, každý opět o velikosti **N**. Na každý z těchto datasetů se natrénuje jeden rozhodovací strom. Tyto stromy jsou díky rozdílnosti dat na sobě nezávislé, což zvyšuje rozmanitost modelu a tím i jeho celkovou stabilitu a přesnost. V případě Random Forest je navíc využíváno náhodného výběru podmnožiny prediktorů v rozhodovacím uzlu, ze kterých je následně vybrán nejlepší split pomocí např. Giniho Indexu [36]. Tímto způsobem je zlepšena schopnost generalizace. Výsledná predikce se získává kombinací výstupů jednotlivých stromů. V regresních úlohách se průměrují (Tabulka 5) zatímco v klasifikaci se používá nejčastější hodnota.

Tabulka 5: Bagging – regrese a klasifikace **Zdroj**: [28]

Regrese – průměr predikcí:	**Klasifikace – modus (nejčastější hodnota):**
$\hat{f}(x') = \frac{1}{B}\sum_{i=1}^{B} f^{(b)}(x')$	$\hat{f}(x') = \mathbf{modus}\{\mathbf{f}^{(1)}(x'), \mathbf{f}^{(2)}(x'), \ldots \mathbf{f}^{(B)}(x')\}$

Random Forest se ukázal být velmi přesný v případě predikce popularity ve studii **„Predicting Popularity of YouTube Videos Using Viewer Engagement Features“** z roku 2022, která na platformě YouTube pomocí několika klasifikačních algoritmů predikovala na základě získaných engagement metrik (Lajky, komentáře atd.), jestli bude video populární. Jeden z těchto algoritmů byl právě Random Forest, který se ukázal být nejlepším prediktorem, a to s **úspěšností 91 %** [37]**.**

Nevýhodou Random Forest metody je její špatná interpretovatelnost, celý „les“ se totiž velmi obtížně interpretuje a výsledný model je tak „černou skřínkou“, tzn. výsledky jsou známy, ale cesta k nim není snadno interpretovatelná. Problémy jsou i s výpočetním výkonem, jelikož je model plný stromů v rámci stovek až tisíců. To vede také k pomalé predikci v reálném nasazení.

2.3.3 Boosting metoda XGBoost

Pokročilou metodou strojového učení je XGBoost, tedy **Extreme Gradient Boosting**, jenž patří mezi nejvýkonnější a nejrozšířenější algoritmy pro predikční úlohy v oblasti datové vědy, a to zejména díky své vysoké přesnosti, rychlosti trénování a možnosti ladění hyperparametrů [38].

Stejně jako v předešlé kapitole je použito **ensemble modelování**, ale tentokrát se jedná o využití techniky **gradient boostingu**, což znamená, že nové modely (standardně se jedná o rozhodovací stromy) se učí sekvenčně a každý nový strom zlepšuje model tím, že se zaměřuje na chyby stromů předchozích [38]. Výsledná predikce se získává kombinací výstupů všech stromů.

Matematicky je cíl trénování metody XGBoost vyjádřen pomocí **účelové funkce**, která obsahuje dvě klíčové složky. První složkou je **ztrátová funkce l**, která měří rozdíl mezi skutečnou a predikovanou hodnotou. Druhou složkou je **regulizační člen Ω**, který penalizuje složité modely, čímž zabraňuje přeučení [38]. Vzorec této účeloví funkce je:

$$\boldsymbol{L^{(t)} = \sum_{n=1}^{n} l\left(\mathbf{y}_i, \hat{\mathbf{y}}_i^{(t-1)} + f_t(x_i)\right) + \Omega(\mathbf{f}_k)}$$

Tento přístup umožňuje dosahovat vysoké generalizace, což v kombinaci s možností paralelizace výpočtů a regularizací L1 a L2 (penalizace absolutních a čtvercových hodnot vah) vytváří z metody XGBoost efektivní model pro predikční úlohy, a to i v případě popularity na sociálních sítích.

XGBoost se totiž ukázal jako efektivní model ve studii **„Optimizing Prediction of YouTube Video Popularity Using XGBoost“** z roku 2021**,** která zkoumala jeho výkon při odhadu počtu zhlédnutí a engagementu YouTube videí. Testování probíhalo na rozsáhlém datasetu, který obsahoval metriky jako počet zhlédnutí v prvních hodinách (early Views), míra sdílení a engagement rate, které zároveň patřily mezi ty nejvýznamnější. Výsledky studie ukázaly, že XGBoost dosáhl průměrné přesnosti predikce až **88 %** [39], čímž překonal metody regrese a Random Forest.

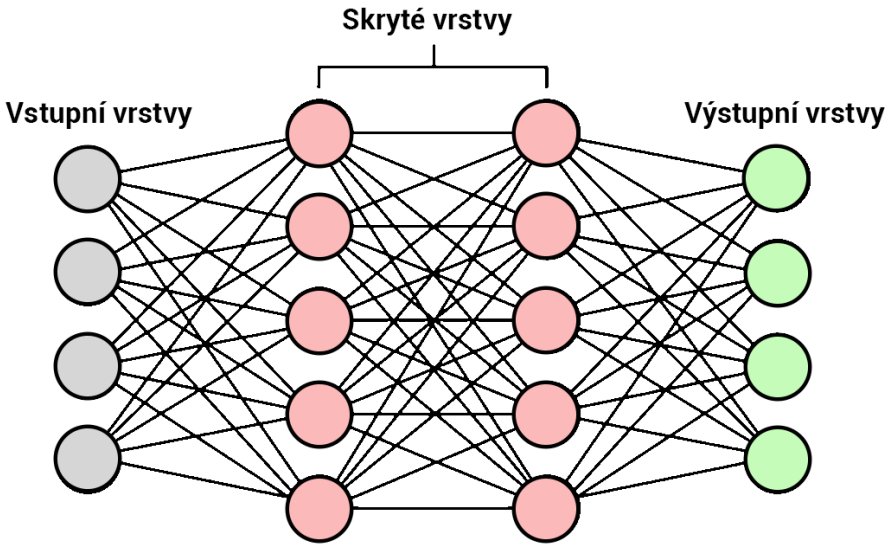
XGBoost se tak ukazuje jako velmi silný nástroj pro predikci popularity, a to i u videí na sociálních sítích. Jeho nevýhodou však může být nutnost pečlivého zpracování vstupních proměnných u komplexních dat riziko přeučení, které je potřeba řešit pomocí regularizací [38].

2.3.4 Neuronové sítě

Neuronové sítě představují další pokročilý přístup v oblasti predikce popularity, jelikož dokážou efektivně modelovat složité nelineární vztahy a zachytit specifické vzory v datech. Tyto modely jsou inspirované strukturou lidského mozku a vycházejí z myšlenky, že informace jsou zpracovávány prostřednictvím propojení jednotlivých neuronů [40]. Umělé neuronové sítě proto provádí výpočet pomocí propojených uzlů (neuronů), které přenášejí a transformují vstupní informace.

Základní architektura neuronových sítí se skládá z několika vrstev – **vstupní vrstvy**, **skryté vrstvy** a **výstupní vrstvy**. Každý neuron přijímá vstup, aplikuje váhy, přičte bias a výstup následně transformuje pomocí aktivační funkce. Právě tento princip umožňuje zachytit i složité nelineární vztahy v datech.

Nejjednodušší typ neuronové sítě je dopředná (**feed-forward)** síť (Obrázek 7). Neobsahuje žádné cykly a zpětné propojení, které je možné u složitějších architektur. Každý neuron předává výstup jen do neuronů ve vrstvě následující, nikdy zpět [40]. Tuto architekturu využívá například **Multilayer Perceptron (MLP),** která byla využita v studii **Multilayer Perceptron Based on Joint Training for Predicting Popularity** a umožnila zachytit významné klíčové vlastnosti a predikovat popularitu [41]. Úspěšnost MLP byla **82 %**, což předčilo ostatní použité metody strojového učení.



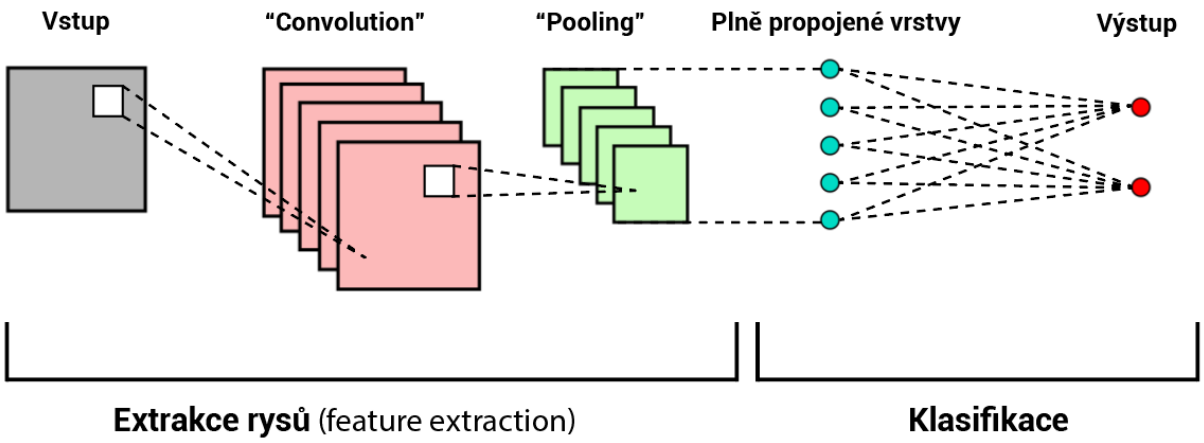
Obrázek 7: Základní vizualizace neuronové sítě **Zdroj**: Vlastní zpracování

Mezi složitější architektury, které jsou využívané i v predikci popularity na sociálních sítích, patří **konvoluční** neuronové sítě (CNN) a **rekurentní** neuronové sítě (RNN). Každá z těchto metod se zaměřuje na jiný typ vstupních dat. V praxi bývají tyto architektury často kombinovány pro nejlepší predikční výkon [40].

Konvoluční síť (CNN) jsou jednou z nejpoužívanějších architektur hlubokého učení, která je efektivní zejména při práci s obrazovými a časově závislými daty. Jejich klíčovým principem je využití konvolučních vrstev, které aplikují malé filtry (tzv. „**receptivní pole**“) na vstupní data. Tyto filtry se při tréninku modelu učí rozpoznávat různé charakteristiky, a to s menším počtem parametrů než plně propojené vrstvy. Díky tomu jsou CNN sítě výpočetně efektivní a dobře škálovatelné [40].

Proces konvoluční sítě se dá rozdělit do dvou částí – **extrakce rysů (feature extraction)** a **klasifikace**. V první části jsou na vstupní data aplikovány konvoluční vrstvy, které pomocí receptivních polí vyhledávají lokální vzory (textury, hrany a další složitější charakteristiky). Následně se přechází k fázi „**pooling“**, který zmenšuje dimenzi dat a zachovává pouze klíčové informace.

Výsledky této extrakční fáze následně putují do **fully connected layer,** tedy plně propojené vrstvy, která již slouží ke klasifikaci výstupu [40]. Celý proces je vyobrazen na následujícím Obrázku 8.



Obrázek 8: Princip konvoluční sítě **Zdroj**: Vlastní zpracování

CNN našel své využití v studii **„Multi-branch LSTM Encoded Latent Features with CNN-LSTM for Youtube Popularity Prediction**“ z roku 2025, která zpracovala obsáhlý dataset o počtu více jak 5,3 milionu videí z YouTube [42]. Ten byl získán pomocí YouTube API a dalších externích nástrojů třetích stran, což umožnilo získat detailnější informace o videích. V této studii byla nejdříve samostatně využita složitější rekurentní sít LSTM, která zpracovala časové atributy videí. Ta byla následně podpořena architekturou CNN s cílem extrahovat vizuální vlastnosti náhledových obrázků videí. Toto zohlednění vizuálních prvků pomocí CNN zvýšilo úspěšnost predikce tohoto modelu o 50 %.

Rekurentní sítě (RNN) pracují se sekvenčními daty, jako jsou texty nebo časové řady. Umožňují uchovávat informace z předchozích kroků pomocí skrytých stavů (**paměť**), čímž modelují závislosti v čase [40]. Základní princip je tak založen na neuronech, které ve skryté vrstvě přijímají nejen vstupní data z aktuálního kroku, ale i vstupy z kroku předchozího. Tak zohledňuje minulé stavy a dynamicky modeluje.

Standardní RNN architektura bohužel trpí problémy s mizícími a explodujícími gradienty, tedy se jim tzv**. rozpadá paměť** – buď zapomíná důležité informace, nebo je naopak přeceňuje, což komplikuje učení dlouhodobých závislostí.

Pro překonání tohoto problému byla vyvinuta architektura **Long Short-Term Memory (LSTM)**, která byla využita v předešlé studii [42]. Ta obsahuje tři brány – zapomínací (Forget gate), **vstupní** (Input gate) a **výstupní** (Output gate), které regulují, jaké informace se uchovají, přidají a odešlou do výstupu.

Z důvodu výpočetní náročnosti LSTM vznikla její zjednodušená varianta zvaná **Gated Recurrent Unit (GRU)**. Ta spojuje brány na pouhé dvě – která používá pouze dvě brány – **resetovací** (Reset gate) a **aktualizační (**Update gate). GRU šetří parametry a čas, ale přesto dosahuje podobné přesnosti jako LSTM, zejména u menších datasetů**.**

Využití metody GRU pro jeho nižší náročnost, ale větší výkonnost, potvrzuje studie **NPP: A Neural Popularity Prediction Model for Social Media Content**, která se zaměřila na predikci popularity textových příspěvků na platformě Twitter. [13]. GRU bylo aplikováno v několika oblastech – od enkodéru textového obsahu po uživatele a časové vzory – a dosáhlo nejlepší přesnosti ze všech testovaných metod (přibližně 87 %). Tyto výsledky byly lepší i v porovnání s XGBoost a složitějším LSTM.

Neuronové sítě tak představují velmi výkonný nástroj schopný zachytit velmi složité a nelineární vztahy v datech, a to jak v podobě klasických vrstev, tak v pokročilých sekvenčních architekturách jako RNN, LSTM či GRU.

2.3.5 Souhrn metod strojového učení

V porovnání s tradičními statistickými metodami, které často předpokládají lineární vztahy a mají omezenou schopnost pracovat s komplexními daty, nabízí strojové učení flexibilnější a výkonnější přístupy. Tyto metody dokážou zachytit nelineární vztahy, kombinace proměnných a skryté vzory v datech, což z nich činí vhodný nástroj pro predikci popularity videí. Výsledná přesnost však výrazně závisí na kvalitě dostupných dat a jejich důkladném předzpracování.

Z hlediska jednoduchosti a výpočetní efektivity zůstává **SVR** vhodnou volbou. Vyniká přehlednou strukturou a solidní přesností zejména u menších datasetů a krátkodobých predikcí (např. vývoj zhlédnutí do 24–48 hodin po zveřejnění videa). S růstem objemu dat a časového horizontu však jeho účinnost klesá.

Bagging metoda **Random Forest** oproti tomu nabízí o něco robustnější model a lépe škáluje na větší datasety a nevyžaduje složité ladění hyperparametrů. Díky technice baggingu je stabilní i bez složitého ladění a nabízí dobrou přesnost. Využívá se pro klasifikaci i predikci metrik jako engagement rate, ale i klasické počty zhlédnutí, lajky nebo komentářů.

S navyšováním složitosti datasetů a požadavků je ale více doporučována spíše boosting metoda **XGBoost**. Ty jsou již náročnější především skrze optimalizaci a porozumění modelu. Jsou ideální pro predikci popularity kombinací vícero vstupních atributů – např. celkových zhlédnutí během několika týdnů s využitím číselných i kategorických proměnných.

Neuronové sítě, zejména **konvoluční (CNN)** a **rekurentní (RNN)**, poskytují nejvyšší flexibilitu a schopnost modelovat komplexní nelineární vztahy. CNN se uplatní při analýze vizuálních prvků (např. náhledových obrázků), zatímco RNN efektivně pracují s časovými daty, například při sledování vývoje sledovanosti v čase nebo analýze komentářů.

Jednotlivé metody je vždy nutné mezi sebou porovnat a najít vhodnou pro daný typ dat. Proto je důležité je vyhodnotit a porovnat pomocí standardizovaných metrik, které umožní objektivní porovnání jejich výkonu. Metody, které se k tomuto vyhodnocení využívají, jsou rozebrány v následující kapitole.

2.4 Metody zhodnocení výkonu modelu

Hledání těch nejlepších predikčních modelů je nedílnou součástí procesu tvorby modelu. Samotná schopnost modelu generovat výstupy totiž ještě neznamená, že jsou kvalitní nebo užitečné. Jejich úspěšnost je proto potřeba změřit a pokud vytváříme několik různých predikčních modelů zároveň, tak následně i mezi sebou porovnat. K tomu slouží hned několik metrik zhodnocení výkonu modelu, pomocí kterých se dá objektivně určit, jak doopravdy přesný model je.

2.4.1 Regresní metriky hodnocení

Slouží k vyhodnocení přesnosti modelů, které předpovídají spojitou číselnou hodnotu (například počet zhlédnutí videa nebo engagement rate). Jejich hlavním cílem je změřit, jak moc se predikované hodnoty liší od skutečných. Každá z metrik hodnotí chybu trochu jiným způsobem – některé zdůrazňují průměrnou odchylku, jiné penalizují větší chyby víc než ty menší.

Mezi nejpoužívanější patří střední kvadratická chyba **MSE**, její odmocněná verze **RMSE,** průměrná absolutní chyba **MAE** a koeficient determinace **R^2**, který ukazuje, jak velkou část variability ve výstupech model dokáže vysvětlit [43]. Volba konkrétní metriky závisí na tom, co je v dané úloze důležitější – zda absolutní přesnost, stabilita, nebo robustnost vůči extrémům. Přehled je popsán v Tabulce 6.

Tabulka 6: Hodnocení metod: přehled regresních metrik **Zdroj**: [35]

Metrika	Popis funkce	Vzorec
MSE	Měří průměr čtvercových rozdílů mezi predikovanými a skutečnými hodnotami, čím nižší MSE, tím přesnější model	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
RMSE	Odmocněná verze MSE. Lépe interpretovatelná, protože má stejnou jednotku jako cílová proměnná	$\sqrt{MSE}$
MAE	Měří průměrnou absolutní odchylku mezi predikcí a skutečností. Je méně citlivý na extrémy než MSE/RMSE	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$
R^2	Vyjadřuje, kolik procent variability cílové proměnné dokáže model vysvětlit. Hodnota blížící se 1 značí velmi přesný model, hodnota 0 znamená, že model nevysvětluje nic.	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(y_i - \bar{y}_i)}$

2.4.2 Klasifikační metriky hodnocení

U klasifikačních úloh je cílem správně určit kategorii, do které daný výstup patří, tedy například rozpoznat, zda video bude populární, nebo ne. Na rozdíl od regresních úloh zde nejde o předpověď přesné hodnoty, ale o správné rozhodnutí mezi předem definovanými třídami. Hodnocení modelů je proto založeno na tom, kolik rozhodnutí bylo správných a kolik chybných. K určení úspěšnosti modelů se využívá tzv. matici záměn (**Confusion Matrix)**, která rozlišuje čtyři scénáře (Tabulka 7).

Tabulka 7: Hodnocení metod: matice záměn **Zdroj**: [35]

Název scénáře	Popis
True Positive (TP)	Model správně předpověděl pozitivní výsledek
False Positive (FP)	Model chybně označil výsledek jako pozitivní
True Negative (TN)	Model správně předpověděl negativní výsledek
False Negative (FN)	Model chybně označil výsledek jako negativní

Pro lepší představu lze říct, že pokud model předpověděl video na sociální síti jako populární, ale nebylo, jedná se o stav „False positive“. Pokud model predikoval, že video nepřesáhne zhlédnutí a opravdu nepřesáhlo, jedná se o stav „True negative“.

Tyto scénáře se využívají jako výstup k výpočtu vzorců klasifikačních metrik. Mezi nejčastěji používané klasifikační metriky patří **Accuracy** (přesnost), **Precision** (preciznost), **Recall** (úplnost) a **F1-score**. Přehled je uveden v následující Tabulce 8.

Tabulka 8: Hodnocení metod: přehled klasifikačních metrik **Zdroj**: [35]

Metrika	Popis funkce	Vzorec
Accuracy	Určuje, kolik klasifikací bylo správných ze všech. Často používaná, ale může být zavádějící u nevyvážených dat.	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	Kolik z predikovaných pozitivních případů bylo skutečně pozitivních. Důležitá, pokud je cílem minimalizovat falešné poplachy.	$\frac{TP}{TP + FP}$
Recall	Kolik skutečně pozitivních případů model zachytil. Důležité, pokud nechceme přehlédnout žádný pozitivní případ.	$\frac{TP}{TP + FN}$
F1-score	Harmonický průměr preciznosti a úplnosti. Vhodné, pokud je třeba vyvážit obě metriky.	$2 \times \frac{Precision \times Recall}{Precision + Recall}$

3 PREDIKCE POPULARITY VIDEÍ NA YOUTUBE

S využitím všech zjištěných informací bylo možné se přesunout k vlastní realizaci predikčního modelu. Cílem této kapitoly bylo ukázat celý proces inspirovaný rámcem **CRISP-DM**, jenž poskytuje jak obchodní porozumění celému problému, tak logickou strukturu pro zpracování datových projektů od sběru až po vyhodnocení a doporučení do příštích analýz [44]. Všechny analytické postupy byly provedeny v programovacím jazyce **Python** využívající knihovny jako **pandas** nebo **scikit-learn** pro práci s daty.

3.1 Obchodní porozumění

V první fázi projektu bylo důležité si ujasnit, proč to celé vlastně děláme a jaký reálný cíl nás motivuje k vytvoření predikčního modelu. Každý projekt by měl být užitečný ve skutečném světě a naplnit tak požadavek, který může něčemu pomoct. Proto se tato kapitola zabývá obchodní stranou celého problému a pomáhá udržet směr celého procesu tvorby predikčního modelu popularity.

3.1.1 Kontext a definování cíle

Sledování videí na internetu se stalo natolik populární, že motivovalo velké množství lidí tvořit svůj vlastní obsah. Pro mnohé, zejména mladší lidi, se dnes jedná o reálnou kariérní vidinu. **YouTube** patří mezi největší hráče v tomto prostředí a denně na něj proudí obrovské množství nového obsahu. V takto přehlceném prostoru je čím dál obtížnější zaujmout publikum a dosáhnout takového počtu zhlédnutí, který by jejich videu zajistil viditelnost a reálný dopad.

Cílem tohoto projektu se tak stalo vytvoření **predikčního modelu,** který bude schopen odhadnout počet zhlédnutí videa na platformě YouTube ještě před jeho zveřejněním, a to na základě reálných dat, které mohou ovlivnit popularitu. Úspěšnost modelu bude měřitelná z hlediska analytiky a zpětné vazby po nasazení.

Různorodé chování uživatelů, rozdílné formáty a velmi rychlé změny trendů výrazně ztěžují predikci všech videí rovnoměrně, proto bylo rozhodnuto se zaměřit pouze na klasické **long-form** videa a pouze jednu kategorii – **herní videa**. Tato kategorie je na YouTube stále velmi populární, a tak existuje dostatek potřebných videí, které je možné využít k datové analýze.

Model se tak měl stát **podpůrným nástrojem**, který by mohl být využit tvůrci nebo firmami k rozhodnutím v rámci tvorby a propagace obsahu, jelikož by předem věděli, jakého počtu zhlédnutí video může zhruba dosáhnout.

3.1.2 Analytická interpretace

Z hlediska datové analytiky bylo nutné vše správně interpretovat do analytického světa. Jelikož bylo cílem získat odhad počtu zhlédnutí videa, bylo vhodné dívat na problém jako na **regresní úlohu**, jejímž výstupem je číselná hodnota reprezentující očekávaný výkon videa.

Aby bylo možné takový úkol realizovat, bylo potřeba se zamyslet i nad tím, jaká data budou k dispozici a jaké informace by mohly být využitelné jako vstupy. Klíčové se tak stalo i sehnat **kvalitní dataset**, který obsahuje užitečné atributy ovlivňující úspěšnost videa. Předpokládalo se využití základních informací o videu jako název videa, den vydání nebo vložené tagy. Konkrétní výběr atributů byl ale upřesněn až na základě reálné struktury dat a jejího zpracování.

Regresní predikci bylo možné provést hned několika přístupy. Pro tento typ datové analytiky bylo tak vhodné vyzkoušet základní lineární **regresi**, ale také pokročilé modely jako jsou **rozhodovací stromy** nebo **neuronové sítě**. Každý z těchto přístupů má své výhody, ale i omezení. Z tohoto důvodu se počítalo s porovnáním jejich výkonnosti a výběrem nejvhodnějšího modelu.

3.1.3 Měřitelnost výkonnosti

Aby se dalo zjistit, jak přesně model fungoval, bylo potřeba nastavit jasný způsob, jak jeho úspěšnost změřit. K tomu byly použity běžně používané regresní metriky,

- konkrétně střední absolutní chyba (**MAE**),
- střední kvadratická chyba (**RMSE**),
- koeficient determinace ($\mathbf{R^2}$).

Díky nim je možné snadno porovnat, jak moc se predikovaný počet zhlédnutí liší od skutečného výsledku. To umožňuje vyhodnotit, zda model plní stanovený cíl a jak velkou odchylku vykazuje při aplikaci na reálná data. Touto formou měření výkonnosti tak bylo možné zjistit celkovou výkonnost navrženého nástroje.

Výkonnost je vhodné měřit i umožněním uživatelům zanechat své hodnocení, jelikož zpětná vazba je důležitým prvkem pro zdokonalování nástroje.

3.1.4 Jaká jsou omezení?

Při návrhu predikčního modelu bylo nutné zohlednit několik předem známých **omezení**, která mohla ovlivnit kvalitu a rozsah výsledného řešení. Úspěšnost videa například ovlivňuje několik faktorů, které nemusí být v datech dostupné nebo se obtížně zpracovávají. Bohužel právě tyto atributy často hrají významnou roli v tom, zda video upoutá pozornost publika.

Významné vlivy jsou navíc těžko předvídatelné nebo velmi proměnlivé. Mezi ty patří například virální sdílení známou osobností, náhodné zachycení trendu nebo silné zapojení publika. Tyto faktory mohou způsobit výrazné odchylky v predikovaných hodnotách a není možné s nimi dopředně počítat.

Omezující je i již zmíněná široká škála kategorií a typu obsahu, kvůli které bylo nutné predikční model zaměřit pouze na část obsahu na YouTube. Tato volba je záměrná kvůli dostupnosti dat a konzistentnosti vstupů, ale znamená, že výsledky modelu nebylo možné bez další analýzy aplikovat na jiné typy videí (např. vlogy).

3.1.5 Možná rizika a shrnutí

I přes známá omezení existují i rizika, které mohly výrazně ovlivnit projekt k horšímu. Jednou z hrozeb je nasbírání nerelevantních dat, jelikož proměnlivost trendů na YouTube je vysoká a co je populární dnes může být brzy neúspěšné. Bylo proto žádoucí získat data z posledních týdnů.

Získaný dataset by také mohl být silně nevyvážený díky špatné technice jejich sběru nebo omezením samotného zdroje dat, což by způsobilo extrémně nevyvážený počet úspěšných a neúspěšných videí. Tento nepoměr by pak mohl naučit predikční model preferovat extrémní případy, což by negativně ovlivnilo jeho schopnost predikovat výsledek skrze nově získaná data.

S veškerými omezeními a riziky proto bylo počítáno. Následující kapitola se věnuje konkrétním krokům zpracování dat, které byly základem pro tvorbu a testování predikčního modelu.

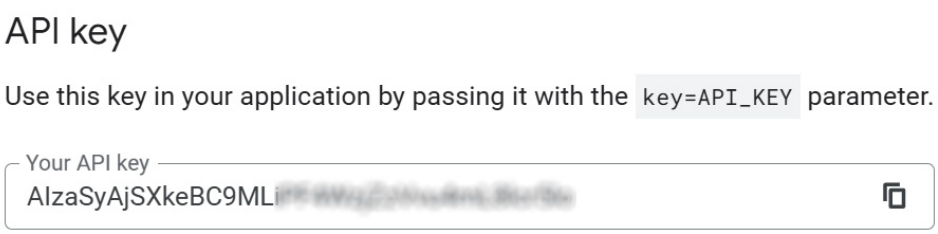
3.2 Sběr reálných dat YouTube

Pro tvorbu prediktivního modelu byla získána vlastní **reálná data** přímo od společnosti YouTube prostřednictvím jejich rozhraní **YouTube API**, přesněji jejich nejnovější verzí **Data API v3**. S tímto rozhraním lze pracovat až po získání speciálního API klíče, který se vkládá do kódu, který dokáže data sehnat a vytvořit z nich **CSV dataset**. Sběr dat z API je velmi specifický, proto je dobré si ho podrobněji představit. Kompletní kód se nachází v **Příloze A** ve složce **„Sběr dat".**

3.2.1 Získání YouTube API klíče

Klíč do YouTube API se dá získat na webových stránkách **Google Cloud Console**, kde je potřeba se přihlásit pomocí Google účtu. Tato webová stránka nabízí širokou škálu nástrojů a služeb pro vývoj, ukládání, zpracování a analýzu dat. Jednou z funkcí je právě i vygenerování potřebného klíče.

Po přihlášení bylo nutné vytvořit nový projekt „YouTube Data". V tomto projektu bylo potřeba skrze podstránku „Knihovna" vyhledat Data API v3 a zaškrtnout povolení. Poté bylo potřeba nalézt stránku „Credentials", kde se vytvořily přihlašovací údaje, konkrétně tedy „**Klíč API**". Google Cloud neprodleně vygeneroval přístupový kód, jenž slouží k autorizaci a umožnění přístupu k práci s celou API. Částečně skrytý klíč je vidět na následujícím Obrázku 9.



Obrázek 9: Získaný YouTube API klíč **Zdroj**: Screenshot, Google Cloud Console

3.2.2 Připojení k API a deklarace parametrů vyhledávání

Po úspěšném získání přístupu k API bylo možné přejít k stažení datasetu. Python byl zvolen pro svou širokou podporu při práci s daty a dostupné knihovny umožňující přímý přístup k YouTube Data API. Tento programovací jazyk lze využívat přímo v příkazovém řádku Windows, ale z důvodu příjemnější práce s kódem jsem použil program „**Visual Studio Code**". Kód byl čerpán z oficiální dokumentace **Google API Python Client** [45].

V první řadě byly nainstalovány potřebné knihovny. V tomto případě šlo o oficiální součástí klientské Google API sady „**googleapiclient.discovery**“, jenž zajišťuje, že uživatel nemusí ručně psát HTTP požadavky. Druhou knihovnou byla „**pandas**“, která umožňuje snadnou manipulaci s tabulkovými daty a dodatečně byla použita knihovna „**isodate**“ kvůli konverzi časového formátu ISO 8601 na datetime objekt.

Po instalaci a importu knihoven byl do vytvořené proměnné „**API_KEY**“ vložen získaný unikátní API klíč. Ten zpřístupní napojení k YouTube API v importované funkci „**build()“,** která vytvoří klienta pro přístup k rozhraní YouTube, a to přesněji do verze **v3**. Celá tato funkce je vložená do proměnné „**youtube**“ (Obrázek 10).

```
from googleapiclient.discovery import build
import pandas as pd
import time
from isodate import parse_duration

API_KEY = "AIzaSyAjSXkeBC9ML
youtube = build("youtube", "v3", developerKey=API_KEY)
```

Obrázek 10: Import, připojení k YouTube API **Zdroj**: carbon.now.sh

Ve světě videí existuje **velké množství kategorií**, které lze na internetu sledovat, a každé z nich má jiný typ konzumentů, střídavost trendů atd. Proto bylo rozhodnuto vybrat pouze jednu z populárních kategorií, na které se model bude učit. Vzhledem ke střídavosti trendů bylo potřeba také vymezit, jak stará data jsou už příliš stará.

Z těchto důvodů jsem se rozhodl stáhnout základní dataset o počtu minimálně **10 000 videí**, který bude tvořen videi z kategorie „**Gaming**“, tedy herních videí. Tato kategorie je v YouTube API označena číslem „**20**“. Herní videa nemají tak silnou střídavost trendů, když pomineme vydání nových populárních videí. Proto jsem vymezil datum vydání videa na ty **nejnovější až po začátek roku 2025**.

Deklarované proměnné vypadají takto (Obrázek 11):

```
CATEGORY_ID = "20"
PUBLISHED_AFTER = "2025-01-01T00:00:00Z"
MAX_VIDEOS = 10000
```

Obrázek 11: Deklarace základních parametrů vyhledávání **Zdroj**: carbon.now.sh

3.2.3 Návrh procesu získávání dat z API

Každé video na YouTube má svůj vlastní unikátní identifikátor (tzv. **„videoID“**), který lze využít jako klíčový vstup pro následné dotazy na detailní informace. K jejich získání byla vytvořena samostatná funkce, která využívá jednu ze služeb YouTube API **„search().list()“**. Ta posloužila k vyhledání videí ve vybrané kategorii a také v zadaném časovém období.

Naneštěstí má YouTube API omezený počet dotazů, které za den lze pomocí jednoho API klíče vyvolat, a to **10 000 jednotek**. Funkce **„search()“** je zároveň nejvíce náročná na tento limit – každý dotaz sice vrací až 50 výsledků, ale spotřebuje 100 jednotek. To znamená, že s jedním API klíčem lze získat **nanejvýš 5 000 videoID denně**.

V dalším kroku jsem využil funkce **„video().list()“** a **„channel().list()“**, které ale jednotlivě na video využívají pouze 1 jednotku. Proto byla vytvořena Tabulka 9, která ukazuje denní spotřebu dotazů a kolik je možné denně zpracovat videí.

Tabulka 9: YouTube API: Denní limitace **Zdroj**: [38]

API dotaz	1 dotaz	Počet výsledků	Dotazů/den	Celkem
search().list()	100 jednotek	50 videoID	100	5 000 videoID
videos().list()	1 jednotka	1 video	10 000	10 000 videí
channels().list()	1 jednotka	1 kanál	10 000	10 000 kanálů

Při návrhu procesu získávání dat byly zvažovány dvě možné varianty. První možností bylo rozdělit sběr dat do dvou samostatných fází – nejprve pomocí funkce „search().list()“ získat pouze unikátní identifikátory videí (**videoID**) a ty uložit do samostatného CSV souboru. V druhé fázi by následně tento soubor sloužil jako vstup pro další zpracování, tedy pro získání detailních údajů o videích pomocí funkcí „videos().list()“ a „channels().list()“. Druhou možností bylo ponechat celý sběr dat jako jeden kód využívající všechny API dotazy zakončený kompletním CSV datasetem.

Tento přístup byl nakonec vybrán, jelikož je jednodušší a v daném počtu potřebných dat není zas tak důležité uchovávat již vygenerované videoID. Při využití limitů jednoho API klíče, který za sebou takto volá všechny metody, bylo možné zpracovat až **2 500 videí denně**. Cestou proto bylo si sběr dat rozdělit na více dní.

3.2.4 Kód pro získání ID a detailních informací

Prvním krokem bylo již zmíněné získání unikátních identifikátorů jednotlivých videí. K tomuto účelu byla vytvořena samostatná funkce **get_video_ids()**, která jakožto vstupní parametry používala již nadefinované číslo kategorie, datum publikace a počet požadovaných videí.

Uvnitř této funkce byla vytvořena proměnná „**video_ids**“, která slouží jako prázdný seznam pro postupné ukládání nově získaných ID. Dotaz **search().list()** umožňuje v jednom požadavku stáhnout maximálně 50 výsledků. Těmto shlukům výsledků dat se říká „stránky“, protože připomínají stránkování na webu.

Hlavní část tvořil cyklus „**while**“, který opakovaně posílal dotazy do API, dokud nebylo dosaženo požadovaného počtu ID. Tento dotaz je definován jako „**response**“ využívají právě funkci **search().list().execute()**.

Dovnitř list() byly vložený parametry, které filtrují nejen podle zadané kategorie nebo data, ale také podle sloupce (**part**), typu obsahu (**type**), priorita regionu (**regionCode**) a počtu výsledků v jedné stránce (**maxResults**). Hlavní část funkce „**get_video_ids**“ lze vidět na Obrázku 12.

```python
def get_video_ids(category_id, published_after, max_videos):
    video_ids = []
    next_page = None

    while len(video_ids) < max_videos:
        response = youtube.search().list(
            part="id",
            type="video",
            videoCategoryId=category_id,
            publishedAfter=published_after,
            maxResults=50,
            pageToken=next_page,
            regionCode='US'
        ).execute()
```

Obrázek 12: Hlavní funkce "get_video_ids()" **Zdroj**: carbon.now.sh

V druhé části byl do proměnné „**ids**“ z výsledků vyhledávání vytvořen seznam obsahující pouze ID jednotlivých videí. Konkrétně je pomocí příkazu uvnitř **ids** z každé položky v seznamu výsledků (**items**) získáno pouze samotné **videoId**.

Po získání identifikátorů jednotlivých videí bylo pomocí nich možné začít vyhledávat i veškeré potřebné detailní informace. K tomu byla nadefinována další samostatná funkce „**fetch_video_data**“, která využívá vytvořeného seznamu „**video_id**“.

Stejně jako v předešlé funkci bylo důležité vytvořit proměnnou, která se bude starat o komunikaci s YouTube API. Proměnná, která uchovává odpověď z API, se tentokrát nazývá „**vid_response“**. Uvnitř se nachází proměnná „**part**“, do které bylo napsáno, jaké typy informací mají být z API vyžádány. Vybranými typy jsou tyto:

- **Snippet –** základní popisné informace o videu, které se hodí pro analýzu metadat**.** Vrací data jako název videa, popis videa, datum zveřejnění atd.
- **Statistics** – kvantitativní údaje o výkonu videa, tedy počet zhlédnutí, lajků nebo komentářů.
- **contentDetails** – technické informace o videu. Patří sem například délka a rozlišení videa.

Případ, kdy pod získaným ID nejsou k nalezení informace, je ošetřen podmínkou **„if not vid_response ["items"]“,** která při zjištění absence dat aktuální ID přeskočí. Získaná data byla nahrána do proměnné „**video**“, z které se následně rozdělila do proměnných **snippet**, **stats** a **details** (Obrázek 13).

```python
def fetch_video_data(video_id):
    vid_response = youtube.videos().list(
        part="snippet,statistics,contentDetails",
        id=video_id
    ).execute()

    if not vid_response["items"]:
        return None

    video = vid_response["items"][0]
    snippet = video["snippet"]
    stats = video.get("statistics", {})
    details = video["contentDetails"]
```

Obrázek 13: První část funkce "fetch_video_data()" **Zdroj**: carbon.now.sh

YouTube nabízí i vertikální short-form kontent videí pod názvem „**YouTube Shorts**“. Tento formát se však vyznačuje odlišnými charakteristikami, které zásadně ovlivňují popularitu videí. Často o úspěšnosti rozhoduje především to, co se vizuálně odehrává přímo ve videu, což nebylo s dostupnými daty možné analyzovat.

YouTube API bohužel nenabízí přímý filtr, který by umožnil oddělit klasická videa od Shorts. Filtrace probíhala skrze dotazy na API, ale také pomocí proměnné **„duration_iso“**, do níž byla uložena délka videa získaná z API. Tato hodnota byla následně převedena do sekund v proměnné **„duration_sec“** a pomocí omezující podmínku **„if“** byla zajištěna filtrace všech videí s délkou < **60 sekund.** Tak tato videa byla zpracováním automaticky vyřazena z výsledného datasetu (Obrázek 14).

```
    duration_iso = details.get("duration")
    duration_sec = parse_duration(duration_iso).total_seconds()

    if duration_sec ⩽ 60:
        return None
```

Obrázek 14: Omezení délky videa **Zdroj**: carbon.now.sh

Po získání základních informací o videu bylo následně potřeba získat také metriky o kanálu, který video publikoval. K tomu posloužil další dotaz na YouTube API prostřednictvím funkce **channels().list()** uvnitř proměnné **„channel_response“**, která funguje na stejném principu jako **„vid_response.** Jakožto vstupní parametr proměnné **„channel_id“** bylo použito ID kanálu již získané v předešlém kódu z objektu **snippet**.

Vyžadovaný typ obsahu byl tentokrát pouze „**statistics**“, tedy kvantitativní údaje o kanálu. Všechny vyžadované detailní informace, které bylo potřeba získat, byly vypsány na konci v **„return()“** (Obrázek 15).

```
        channel_id = snippet["channelId"]
channel_response = youtube.channels().list(
        part="statistics",
        id=channel_id
    ).execute()

    if not channel_response["items"]:
        return None

    channel_stats =
       channel_response["items"][0]["statistics"]
    }
```

```
return {
        "videoId": video["id"],
        "title": snippet.get("title"),
        "description": snippet.get("description"),
        "publishedAt": snippet.get("publishedAt"),
        "channelTitle": snippet.get("channelTitle"),
        "channelId": channel_id,
        "views": stats.get("viewCount"),
        "likes": stats.get("likeCount"),
        ... #Zbytek řádků v Příloze A
    }
```

Obrázek 15: Druhá část funkce "fetch_video_data()" **Zdroj**: carbon.now.sh

3.2.5 Hlavní funkce a vytvoření CSV souboru

Obě předešlé funkce byly součástí hlavní nadefinované funkce „**main()**“. Ta funguje jako řídící panel celého kódu, jelikož si podle potřeby volá funkci získání ID a následné doplnění detailních informací. V přesném pořadí byla tato funkce napsána tak, že nejdříve zavolá funkci „**get_video_ids()**“, která následně dle parametrů od API získá seznam identifikátorů videí.

Cyklus „**for**“ následně pro každé ID v seznamu zajistil i získání detailních informací pomocí volané funkce „**fetch_video_data()**“ v proměnné „**data**“. Pokud získání veškerých informací bylo úspěšné, tak byl vzorek zařazen do proměnné „**results**“, která funguje jako finální pole se seznamem všech dat.

Nakonec bylo potřeba vytvořit soubor, ve kterém se všechny data uchovají. Využito bylo knihovny pandas (**pd**) a funkce „**DataFrame**()“, která nakonec celý seznam „results“ vyexportoval do jednoduchého textového souboru CSV, který je standardem v ukládání textových dat (Obrázek 16).

```python
def main():
    ids = get_video_ids(CATEGORY_ID, PUBLISHED_AFTER, MAX_VIDEOS)

    results = []
    for i in ids:
        data = fetch_video_data(i)
        if data:
            results.append(data)
        time.sleep(0.3)

    df = pd.DataFrame(results)
    df.to_csv("gaming_videos_YT.csv", index=False)
    print("Výsledky uloženy do gaming_video_YT.csv")
```

Obrázek 16: Main() funkce **Zdroj**: carbon.now.sh

Doděláním této finální funkce, která vše řídí, bylo dosaženo kódu, který dokáže stahovat data, a to dle vlastních kritérií. Celý kód je detailně popsaný v **příloze A**. Sběr dat je téměř plně automatizovaný, jelikož získání ID a přiřazení detailních informací funguje iteračně.

3.2.6 Extrakce datasetů

Získávání dat bylo náročné kvůli omezenému počtu dotazů na API a omezených možností náhodného vyhledávání, jelikož i bez jakýchkoliv filtračních parametrů má API při každém spuštění často vrací podobná videa, což komplikuje sběr různorodých vzorků. Problémem byla také dominance Shorts, která i při získání 2000 videí často znamenala, že 80–90 % záznamů muselo být odstraněno.

Na tyto problémy bylo zareagováno pomocí využití kombinací filtračních příkazů v dotazu na **search().list()**. Tyto změny dokázaly vyřešit problémy s opakujícími se daty, jelikož při jednotlivých změnách API hledá videa odlišně. Také tak bylo dosaženo větší diverzifikace dat a stažení dat v rozumném časovém období. Tyto filtrace proběhly pomocí parametrů „**order**“, „**q**“ a „**videoDuration**“ (Obrázek 17).

```
order= #Vyhledávání podle popularity videí (Relevance, viewCount)
q= #Vyhledávání pomocí klíčových slov (například "Let's Play")
videoDuration= #Vyhledávání podle délky videa (Short, Medium, Long)
```

Obrázek 17: Filtrační parametry **Zdroj**: carbon.now.sh

Kombinace vyobrazené v Tabulce 10 umožnily stažení 18 datasetů pokrývajících různorodá videa. Počet vzorků v jednotlivých kombinacích byl zvolen tak, aby byl výběr co nejvyváženější. Větší zastoupení mají videa bez specifického řazení („order“) a videa delší než 1 minuta, jelikož bylo potřeba získat co nejvíce různorodých dat včetně kratších klasických videí. S těmito datasety bylo následně pracováno v následující kapitole zabývající se celkovým zpracovaní dat.

Tabulka 10: Kombinace stahovaných dat **Zdroj**: Vlastní zpracování

Délka videa	**q ?**	order =			**Celkem**
		-	**relevance**	**viewCount**	
Medium (4-20 min)	Ano	~ 800	~ 400	~ 400	~ 1600
Medium (4-20 min)	Ne	~ 800	~ 400	~ 400	~ 1600
Long (20+ min)	Ano	~ 800	~ 400	~ 400	~ 1600
Long (20+ min)	Ne	~ 800	~ 400	~ 400	~ 1600
If < 60 (1+ min)	Ano	~ 1200	~ 400	~ 400	~ 2000
If < 60 (1+ min)	Ne	~ 1200	~ 400	~ 400	~ 2000
				Celkem:	**~ 10 400 videí**

3.3 Zpracování dat

Získaná data z YouTube API bylo nutné před samotným použitím nejprve zpracovat. Tento proces zahrnoval úkony jako přiřazení nového sloupce dle názvu původního datasetu, spojení těchto jednotlivých datasetů v jeden, odstranění zbytečných a neúplných záznamů nebo rozšíření o nové proměnné (například den v týdnu, hodina publikace nebo engagement rate). Byly využity především knihovny pandas a glob, které jsou běžně používány pro práci s daty v jazyce Python. Veškeré postupy zmíněně v následujících podkapitolách jsou realizovány pomocí Python kódů a jsou obsaženy v složce „**Zpracování dat" (Příloha A)** pojmenované podle podkapitoly, v které byl kód použit.

3.3.1 Sloupec „source" a spojení souborů

Každému datasetu byl z důvodu kvalitnějšího rozdělení trénovacích a testovacích dat dodatečně přidán sloupec „**source**" vyplněný názvem právě příslušného vzorku dat jako například „MediumRandomSloupec". Tímto způsobem bylo při rozdělení testovacích a trénovacích dat umožněno využití stratifikace, tedy rozdělení dat rovnoměrně podle poměru původních datasetů.

Po přidání sloupce došlo k spojení všech datasetů do jednoho. K tomu byla využita také knihovna **glob**, které jsou využívány k jednodušší práci se zadáváním souborů. Pokud by tato knihovna chyběla, bylo by potřeba napsat všechny CSV ručně, což není optimální vzhledem k počtu spojovaných souborů. Také došlo k smazání duplikátů pomocí „**drop_duplicates()",** která detekovala stejné hodnoty v sloupci **„videoId"**.

3.3.2 Odstranění duplikátů a prázdných hodnot

Již při spojování souborů proběhlo i odstranění duplikátů pomocí porovnání hodnot videoID. Následně při základní analýze bylo pomocí **„df.isnull().sum()"** zjištěno, že soubor obsahuje několik prázdných buněk (Obrázek 18).

```
Chybějící hodnoty v sloupcích:
videoId                 0     likes                 155
title                   0     comments               70
description           335     duration                0
publishedAt             0     definition              0
channelTitle            0     subscribers             0
thumbnail               0     channelTotalVideos      0
channelId               0     channelTotalViews       0
views                   0     source                  0
```

Obrázek 18: df.isnull().sum() **Zdroj**: Screenshot, Visual Studio Code

Prázdná pole byla detekována v sloupci **description**, **likes** a **comments** (Obrázek 18). Záznamy, které měly prázdné jedno z těchto míst, byly smazány. Celkově tak bylo odstraněno přibližně 500 záznamů. Dodatečně byly smazány ještě řádky s **0 views**, jelikož se s velkou pravděpodobností jedná o neveřejná nebo jinak nedostupná videa.

Description sice není povinná hodnota, jelikož tvůrci na internetu nemusí vyplňovat popis videa, ale po hlubším porozumění datům bylo zjištěno, že většina videí, které nemají popis, jsou ve formátu Shorts, které byly delší jak jedna minuta a prošly přes filtr. Z tohoto důvodu byly smazány i tyto záznamy.

3.3.3 Zpracování času a výpočet engagement rate

Pro analýzu byly i dopočítány nové hodnoty. Datum publikace bylo rozloženo na sloupce „**Day_of_week**“ a „**Hour_of_day**“. Ty byly vytvořeny s cílem testování, do jaké míry popularitu videa ovlivňují den a hodina vydání. Tyto sloupce byly zjištěny již z existujícího sloupce „**PublishedAt**“, který obsahoval datum ve formátu ISO 8601. Hodnoty v něm byly překonvertovány a dále dle potřeby rozděleny na den v týdnu a hodinu dne.

Dále byl vytvořen i sloupec „**Engagement_Rate**“, který byl vypočítán pomocí vzorce z tabulky engagement metrik () a zaokrouhlením tak, aby nezasahoval do 10 desetinných čísel. Díky absenci kvalitnějších dat, které z YouTube API nelze získat, nebylo možné dopočítat další důležité metriky, které by pomohly predikci (například Retention Rate).

Sloupec „**duration**“ následně za cílem zjednodušení prošel konverzí na pouhé sekundy. Toho bylo dosaženo funkcí z knihovny **isodate**, která byla využita již při sběru dat při eliminaci videí pod jednu minutu.

3.3.4 Eliminace extrémních hodnot

Po přidání nových sloupců byla pomocí „**df.describe()**“ vytvořen výpis veškerých statistických informací o jednotlivých sloupcích (Obrázek 19). Bylo tak jednoduše zjištěno, jak velká rozmanitost kvantitativních dat v datasetu vlastně je.

Statistické shrnutí číselných sloupců:

	hour_of_day	views	likes	comments	duration	subscribers	channelTotalVideos	channelTotalViews	engagement_rate
count	6854.000000	6.854000e+03	6854.000000	6854.000000	6854.000000	6.854000e+03	6854.000000	6.854000e+03	6854.000000
mean	13.142399	2.477042e+05	7733.051357	513.671725	4647.849723	1.609299e+06	2454.030493	6.723533e+08	5.190495
std	6.264642	1.003802e+06	34323.617581	2302.904595	9184.138259	4.421750e+06	8830.380613	2.291573e+09	8.463831
min	0.000000	1.000000e+00	0.000000	0.000000	62.000000	0.000000e+00	1.000000	2.000000e+00	0.000000
25%	9.000000	7.875000e+02	20.000000	1.000000	721.250000	5.452500e+03	246.000000	1.140591e+06	1.550000
50%	14.000000	2.401750e+04	524.000000	28.000000	1275.000000	1.285000e+05	780.500000	3.268709e+07	3.290000
75%	18.000000	1.945732e+05	4507.500000	302.750000	3881.500000	1.090000e+06	2250.750000	3.739948e+08	5.880000
max	23.000000	3.977608e+07	828225.000000	87456.000000	246292.000000	5.580000e+07	169576.000000	3.893943e+10	200.000000

Obrázek 19: df.describe() **Zdroj**: Screenshot, Visual Studio Code

Ačkoliv data vypadala na první pohled v pořádku, některé maximální hodnoty vykazovaly až příliš odchýlené hodnoty od kvartálu 75 % videí. V novém sloupci „**engagement_rate**“ dosahovaly nerealistických čísel. Engagement Rate se dle zdrojů většinou pohybuje v rozsahu **0 až 10 %**, extrémně populární videa se dokážou dostat přes 10 **%,** ale jedná se o velmi vzácné vzorky [46]. Hodnoty nad tyto procenta jsou pravděpodobně zkreslené hodnoty díky videím s nízkými zhlédnutími, ale velkým počtem interakcí.

Za cílem minimalizace zkreslujících hodnot byly smazány záznamy, které v sloupci „engagement_rate“ přesahovaly hodnotu 15 %, jelikož bylo potřeba zachytit i extrémní případy, které nemusí být artefakty, ale výsledkem velmi aktivní komunity.

3.3.5 První korelační analýza

Po úpravách byla provedena první analýza vztahů mezi proměnnými a zhlédnutím. Kvůli předpokladu nelineárních vztahů mezi proměnnými byl použit **Spearmanův korelační koeficient** (Obrázek 20). Nejvyšší korelace vykázaly proměnné **likes** a **comments**, ty však nejsou dostupné před zveřejněním videa, a proto nejsou vhodné pro predikci. Nebylo možné uznat ani **engagement_rate**, jelikož se skládá z nevhodných proměnných.

Z proměnných známých dopředu silně korelovaly **subscribers** (počet odběratelů) a **channelTotalViews** *(*celkový počet zhlédnutí kanálu*)*. Vysoké hodnoty korelace potvrzovaly předpoklad a tvrzení, že populární kanál bude mít pravděpodobněji populární video než kanál nepopulární. Naopak proměnné jako **duration** nebo **hour_of_day** korelovaly jen minimálně.

Z důvodu tohoto slabého počtu kvalitních prediktorů bylo potřeba využít metod k vytvoření dalších prediktorů, které mohly predikci zpřesnit. K tomu byla využita textová analytika.

Proměnná	Spearman korelace s views
likes	0.943800
comments	0.746028
subscribers	0.708781
channelTotalViews	0.656138
engagement_rate	-0.092309
duration	0.038628
hour_of_day	0.034961
channelTotalVideos	0.023620

Obrázek 20: Korelační analýza (Spearmanův koeficient) **Zdroj**: Screenshot, Visual Studio Code

3.3.6 Textová analytika

Dataset obsahoval nevyužité proměnné obsahující textové řetězce. Na první pohled těmi nejvhodnějšími k predikci popularity videa byly především názvy videí (**title**) a seznamy klíčových slov (**tags**). K jejich zpracování byly využity jednoduché operace textové statistiky jako například počty emotikonů nebo unikátních slov. Také byly vytvořené binární sloupce, které říkají, zdali se v textu objevují otazníky nebo čísla., ale také pokročilejší přístupy TF-IDF a BERT.

TF-IDF byla využita k nalezení 20 nejčastěji se vyskytujících slov v proměnných **title** a **tags** a vytvoření čtyř sloupců, které ukazují, zdali se některé z těchto slov v záznamu vyskytuje. Následně byly i vytvořeny sloupce, které pouze ukazují, zdali se jedno z těchto slov vyskytuje a jestli se jedná o slova, která s cílovou proměnnou korelují kladně nebo záporně.

BERT je jazykovým modelem, který nabízí škálu využití v různých úlohách textové analytiky [25]. V této práci byl využit jazykový model **Multilangual sentiment BERT**, který pokročilým způsobem analyzuje náladu (sentiment) textu a hodnotí ho na škále 0 až 5. Jeho využití bylo důležité i kvůli tomu, že dokáže sám překládat texty a v datasetu se i přes filtraci jazyku objevovala například španělština nebo čeština.

Využitím těchto technik vznikly nové sloupce vypsané v následující Tabulce 11.

Tabulka 11: Sloupce textové analýzy **Zdroj**: Vlastní zpracování

Název sloupce	Popis	Datový typ
title_emoji_count	Počet emotikonů v názvu videa	Int64
title_word_count	Počet slov v názvu videa	Int64
title_token_count_unique	Počet unikátních slov v názvu videa	Int64
tag_count	Počet tagů pod videem	Int64
title_exclamations	Počet vykřičníků v názvu videa	Int64
title_has_number	Výskyt čísla v názvu (Ano/ne)	Int64
title_has_question	Výskyt otazníku v názvu (Ano/ne)	Int64
title_ratio_uppercase_letters	Poměr velkých slov v názvu	Float64
TF-TDFTitle_pos/neg	Výskyt populárního slova v názvu	Int64
TF-TDFTag_pos/neg	Výskyt populárního slova v tagu	Int64
Title_contains_*slovo*	Výskyt daného populárního slova	Int64
bert_sentiment_numeric	Hodnocení sentimentu (0-5)	Int64
day_*název dne*	Vydáno v daný den (Ano/ne)	Int64

3.3.7 Dny v týdnu a One-hot encoding

Po aplikaci textové analytiky bylo využito ještě metody „**One-hot encoding**“, který každou kategoriální proměnnou reprezentuje jako samostatný binární sloupec. Byla tak zpracována doposud pro korelační analýzu i predikční modely neviditelná proměnná „**day_of_week**“, která obsahovala nestrukturované textové kategoriální proměnné označující den vydání.

Aplikace One-hot encoding metody tak vytvořila sedm nových sloupců s binárními hodnotami, které říkají, jestli daný záznam byl vytvořen právě v tento den. Sloupce tak mají stejnou logiku jako sloupce „title_contains_*slovo*“.

3.3.8 Zakřivení hodnot a logaritmizace

Zakřivení hodnot popisuje asymetrii rozložení dat vzhledem k jejich střední hodnotě. Záporná hodnota značí levošikmé rozdělení (více hodnot nad průměrem), kladná hodnota naopak pravošikmé rozdělení (většina hodnot je nízkých a několik extrémně vysokých). Hodnoty zakřivení větší než ±1 se obecně považují za silně šikmé a indikují, že rozložení není symetrické.

Míra zakřivení všech číselných atributů byla vyhodnocena automatizovaně. Všechny proměnné s hodnotou zakřivení (**skewness)** > 1 nebo < -1 byly identifikovány jako silně pravošikmé (hodnoty přesahovaly 3.5) a doporučeny k logaritmické transformaci pro zajištění vyváženějšího rozložení vstupních dat.

Logaritmická transformace byla z důvodu pravošikmosti, tedy dominance vysokých hodnot, využita u sloupců **views, likes, comments, subscribers, channelTotalViews,** a také proměnné **duration**, čímž došlo k eliminaci extrémně dlouhých videí. čímž se eliminoval vliv extrémně dlouhých videí.

Ačkoliv mezi hodnoty s vyšším zakřivením patřila i proměnná **engagement_rate**, jakožto poměrový ukazatel, a **title_exclamations**, jakožto diskrétní početní údaj, byly ponechány v původní podobě kvůli jejich lepší interpretovatelnosti a absenci výrazné deformace rozložení.

3.3.9 Druhá korelační analýza a datový slovník

Po rozšíření datasetu o nové proměnné došlo k výraznému zvýšení počtu atributů, které alespoň slabě korelovaly s počtem zhlédnutí videa. Mezitím, co v původní sadě bylo identifikováno jen několik relevantních proměnných, po přidání znaků založených na textové analýze názvu videa a tagů, zejména TF-TDF skóre a dalších jazykových atributech došlo ke zlepšení a získání dalších prediktorů (Obrázek 21).

Proměnná	Spearman korelace s views
subscribers	0.708781
log_subscribers	0.708781
channelTotalViews	0.656138
log_channelTotalViews	0.656138
TF-TDFTitle_negative	-0.460038
title_contains_play	-0.387632
title_contains_game	-0.276808
title_contains_longplay	-0.243120
title_contains_review	-0.239297
tags_contains_walkthrough	-0.224306
TF-TDFTag_positive	0.220282
title_contains_gameplay	-0.211750
title_ratio_uppercase_letters	0.202024
tags_contains_play	-0.195847
tags_contains_roblox	0.195249
TF-TDFTitle_positive	0.192818
title_word_count	-0.180577

Obrázek 21: Druhá korelační analýza **Zdroj**: Visual Studio Code, Screenshot

Tím bylo ukončeno zpracování dat a byl vytvořen datový slovník (Tabulka 12) ve formě tabulky, která je v celé své verzi v **příloze B**. V další kapitole je pak již možné realizovat predikční modely.

Tabulka 12: Datový slovník **Zdroj**: Vlastní zpracování

Název proměnné	**Popis proměnné**	**Datový typ**	**Příklad obsahu**
title	Název videa, titulek	Object	„ROBLOX ART!“
views	Počet zhlédnutí	Int64	124451, 145, 10445
log_views	Zlogaritmovaný počet zhlédnutí	Float64	13.5, 3.2, 1
day_Friday	Den vydání v pátek	Int64/Bool	0, 1
tag_count	Počet použitých tagů	Int64	11, 3, 0

3.4 Regresní modelování

Kapitola modelování se zabývá samotným návrhem několika regresních modelů. Ty byly jednotlivě testovány a laděny tak, aby docílily své nejvyšší predikční výkonnosti. Důraz byl kladen na to, aby modely netrpěly přeučením, které by se projevilo ztrátou schopnosti generalizovat na nová data.

Modely byly trénovány na transformované proměnné **log_views**, jelikož původní proměnná views vykazovala výrazný rozptyl hodnot a logaritmická transformace pomohla stabilizaci. Kvůli vysoké korelaci proměnných obsahující informace o kanálu byly modely pro účely porovnání trénovány zvlášť na datech s těmito proměnnými i bez nich. Veškeré kódy využity k modelování se také nachází v **příloze A**.

3.4.1 Rozdělení testovacích a trénovacích dat

Při tvorbě predikčních modelů bylo potřeba dataset rozdělit na trénovací a testovací podmnožinu, aby bylo možné objektivně vyhodnotit výkonnost jednotlivých modelů. Rozdělení bylo provedeno ve shodě s existujícími studiemi [47] v tradičním poměru 80 / 20 % pro trénovací a testovací data (Obrázek 22).

Kromě poměru bylo využito také rozdělení pomocí metody **stratifikace** na základě sloupce **source**. Tento sloupec označuje původ konkrétního záznamu, tedy z jaké části zdrojového sběru videí pochází. Jelikož byl dataset složen z více menších sběrů, bylo důležité zachovat jejich poměrné zastoupení v obou částech dat.

Pro zachování opakovatelnosti byl přidán i řádek **random_state**, který zajišťuje, že se rozdělená data nemění při každém spuštění predikce. Hodnota proměnné je irelevantní, ale musí zůstat stejná, aby rozdělovala data stále stejným způsobem.

```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
    stratify=df["source"]
)
```

Obrázek 22: Rozdělení trénovacích a testovacích dat **Zdroj**: carbon.now.sh

3.4.2 Výběr vhodných modelů a jejich ladění

Pro regresní predikci byly použity v předchozích studiích často používané typy modelů. Jako základní „baseline" model byla použita jednoduchá **lineární regrese**, která sloužila jako příklad predikční schopnosti toho nejjednoduššího přístupu.

Vyzkoušena také byla pokročilá regrese **SVR s RBF jádrem**, která dokáže díky své minimalizaci strukturálního rizika získat stabilnější výstupy než ostatní metody založené na pouhé minimalizaci chyby na trénovacích datech. Nejvyšší přesnost se na základě provedené literární rešerše očekávala od ensamble metod **Random Forest** a **XGBoost**, které jsou díky své robustnosti standardem ve oblasti predikce popularity na internetu. Vyzkoušena byla nakonec i neuronová síť **Multilayer Perceptron (MLP).** Interpretace nejlepších výsledků byla provedena až po dokončení modelování.

Každá z metod obsahuje své **hyperparametry,** které určují, jakým způsobem se bude učit. Je možné například měnit počet rozhodovacích stromů v ensemble metodách, počet skrytých vrstev neuronové sítě nebo vyzkoušet jiné jádro pro SVR. Manipulace s hyperparametry slouží k vyladění výkonnosti modelu.

Ladění bylo částečně automatizováno pomocí knihovny **ParameterGrid**, která dle libovolně sepsaných hodnot hyperparametrů vytvořil modely se všemi kombinacemi a našel ten nejvhodnější model (Obrázek 23).

Ten byl s cílem dosažení co nejlepší generalizace dat vyhodnocen pomocí pomocného výpočtu **rozdílu R^2** na trénovacích a testovacích datech. Této proměnné byl dán název **r2_diff**. Pro zachování výkonu byl následně vytvořen i pomocný výpočet **score**, který vznikl odečtením **r2_diff** od výsledku **R^2** na testovacích datech a zajišťoval vybrání modelu, který vyvažuje predikční výkon a generalizaci.

Výsledky všech modelování byly nakonec hodnoceny a porovnávány právě metrikou koeficientem determinace **R^2**. Tento přístup se stal jednotným rámcem pro celou fázi modelování.

```
param_grid = {
    'hyperparametr_1': [1, 2, 5, 10],
    'hyperparametr_2': [0.1, 0.2, 0.05],
    'hyperparametr_3: [ 'auto' ,0.01, 0.1],
    'hyperparametr_4': [5, 10, 150]
}
```

```
r2_train = r2_score(y_train, preds_train)
r2_test = r2_score(y_test, preds_test)
r2_diff = abs(r2_train - r2_test)
score = r2_test - r2_diff
```

Obrázek 23: ParameterGrid a hodnoty pro ladění r2_diff a score **Zdroj**: carbon.now.sh

3.4.3 Základní model: Lineární regrese

Model lineární regrese očekává čistě lineární vztah, což v praxi není ideální pro predikci popularity na sociálních sítích. Z tohoto důvodu sloužila jako základ pro porovnání s pokročilejšími modely.

Model dle výsledků učení poskytl jen omezenou schopnost predikce. Na testovacích datech bez informací o kanálu dosáhl hodnoty **$R^2 = 0.394$**, tedy dokázal vysvětlit méně než polovinu variability cílové proměnné.

Po zahrnutí doplňujících proměnných došlo ke zlepšení predikční schopnosti modelu. Na testovacích datech bylo dosaženo hodnoty **$R^2 = 0.728$**, což značilo dobrou schopnost vysvětlit variabilitu cílové proměnné a ukázalo, že vztah mezi daty o kanálu a počtem zhlédnutí má lineární povahu.

3.4.4 Ladění Support Vector Regression (SVR)

SVR je díky své schopnosti ignorovat malé chyby v rámci epsilon-insenzitivní zóny mnohem vhodnější pro práci s reálnými daty z YouTube, kde často dochází ke zkreslení dat. Díky použití **RBF jádra** navíc dokáže modelovat i nelineární vztahy mezi proměnnými, čímž se stal vhodným modelem pro predikci popularity.

SVR je **citlivé** na měřítko jednotlivých vstupních proměnných, tedy situaci, kdy například jedna proměnná má hodnoty v tisících a jiná v rozmezí 0 až 1. Dataset byl těmito rozdíly přeplněn, proto bylo potřeba provést osvědčené **škálování dat**, které převádí všechny proměnné do podobného číselného rozsahu. Tím zabrání tomu, aby model některé proměnné preferoval jen kvůli jejich větší číselné hodnotě.

V tomto případě byl využit **StandardScaler** (Obrázek 24), která každou proměnnou převede tak, aby průměr celého sloupce byl 0 a směrodatná odchylka 1. To provede vzorcem, který od původní hodnoty v záznamu odečte průměr sloupce a vydělí směrodatnou odchylkou. Tento typ škálování se nazývá **standardizace**.

```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = pd.DataFrame(scaler.fit_transform(X_train), columns=X.columns)
X_test_scaled = pd.DataFrame(scaler.transform(X_test), columns=X.columns)
```

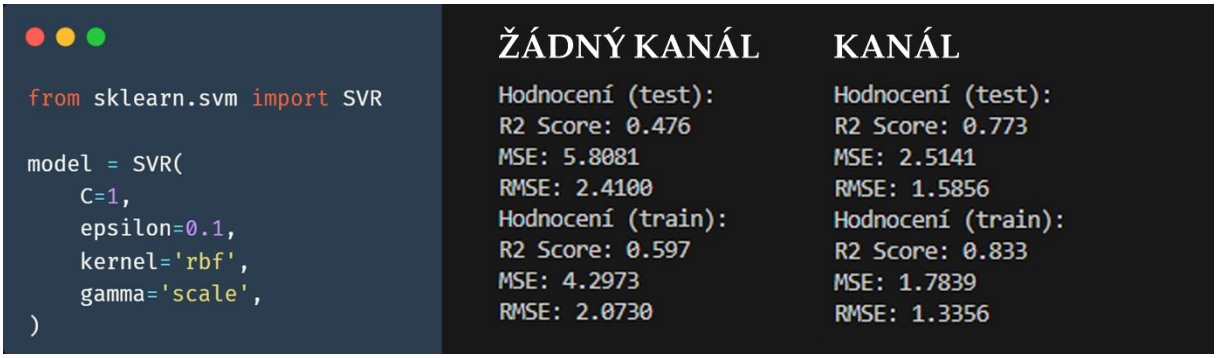
Obrázek 24: Škálování dat pomocí StandardScaler **Zdroj**: carbon.now.sh

Model SVR má několik nastavitelných hyperparametrů, které ovlivňují především jeho práci s penalizací chyb a jádrovými funkcemi. V následující Tabulce 13 jsou uvedeny nejpoužívanější hyperparametry v rámci ladění SVR modelu.

Tabulka 13: SVR – Hyperparametry **Zdroj**: Vlastní zpracování

Hyperparametr	Popis	Typy hodnot
C	Míra penalizace chyb.	Kladné číslo (1, 10, 100)
Epsilon	Šířka zóny, kde se chyba nepočítá.	Malé kladné číslo (0.1)
Gamma	Šířka RBF jádrové funkce	"scale“,“auto“, 1, 0.1
kernel	Výběr jádra	"rbf“,“linear“,“poly“

Model SVR byl z počátku testován v jeho základním nastavení, tedy s výchozími hodnotami nejdůležitějších hyperparametrů (Obrázek 25).



Obrázek 25: Výsledky SVR – První model **Zdroj**: Visual Studio Code, Screenshot + carbon.now.sh

Při testování na datech bez informací o kanálu dosáhl model pouze průměrných výsledků – na testovacích datech bylo dosaženo hodnoty $\mathbf{R^2 = 0.476}$, což ukazovalo spíše slabší predikční schopnost. Trénovací data dosáhly vyšší hodnoty $\mathbf{R^2 = 0.597}$, což v poměru znamenalo, že model už v tomto nastavení zvládal generalizovat.

Ladění SVR ale nepřineslo výrazně lepší výsledky. Přestože byly vyzkoušeny různé kombinace hyperparametrů, výsledné zlepšení predikce bylo jen mírné. V případě modelu, který neměl informace o kanálu, se mírně snížila predikční síla, ale bylo dosaženo lepší generalizace dat. Model, který vnímal informace o kanálu, mírně navýšil svoji schopnost predikce i zlepšil generalizaci.

Tabulka 14 shrnuje veškeré kombinace hyperparametrů SVR modelu a ukazuje nejlepší po ladění pomocí metody ParameterGrid. V převodu **R^2** na procenta model „**Žádný kanál**“ tak vysvětlil přibližně **47 %** variability v počtu zhlédnutí. Lepších výsledků dosáhl model „**Kanál**“, který vysvětlil **77.7 %** variability.

Tabulka 14: SVR – ParametricGrid a nejlepší modely **Zdroj**: Vlastní zpracování

SVR / ParameterGrid					Nejlepší kombinace:	
Hyperparametr	**Kombinované hodnoty**				**ŽÁDNÝ KANÁL**	**KANÁL**
C =	1	2	5	10	1	1
epsilon =	0.1	0.2	0.05		0.2	0.05
gamma =	scale	0.01	0.1		0.01	0.01
kernel =	rbf				rbf	rbf
-				**R^2_train:**	0.551	0.814
				R^2_test:	**0.466**	**0.777**
				R^2_diff:	0.085	0.036
				Score:	0.381	0.741

Tabulka 14 také ukazuje, že modelu při práci s oběma datasety nejvíce vyhovoval jen základní parametr **C = 1**, tedy když penalizace chyb nebyla příliš razantní a model byl jednoduchý. Gamma v obou případech nejlépe fungovala na nízkých 0.01, díky čemuž model méně vnímá vztahy mezi daty, což vede k lepší generalizaci.

V případě modelu „**Žádný kanál**“ byla lepší větší šířka epsilon zóny 0.2, tedy když byly malé chyby ignorovány. „**Kanál**“ modelu vyhovovala mnohem menší epsilon zóna pravděpodobně kvůli lineárnějším vztahům mezi proměnnými.

3.4.5 Ladění RandomForest (RF)

Další metoda, která byla využita k predikci popularity, je **RandomForest**, která díky své robustnosti a schopnosti zachycení komplexních vzorců měla potenciál získat lepší výsledky než SVR. Její odolnost vůči extrémům se navíc velmi hodila na práci s datasetem z YouTube API. Nejdůležitější hyperparametry Random Forest určují především o maximální hloubku stromů, jejich počtu rozhodovacích stromů a uzlech, které určují jejich rozdělování (Tabulka 15).

Tabulka 15: RF – Hyperparametry

Hyperparametr	Popis	Typy hodnot
n_estimators	Počet rozhodovacích stromů.	100, 200, 500
max_depth	Maximální hloubka stromů.	None, 5, 10, 20
min_samples_leaf	Minimální počet vzorků v listovém uzlu.	1, 2, 5
bootstrap	Použít bootstrap vzorkování?	True, False
min_samples_split	Minimální počet vzorků pro rozdělení vnitřního uzlu.	2, 5, 10
max_features	Počet proměnných zvažovaných při rozdělení uzlu.	'None', 'sqrt', 'log2'

Zdroj: Vlastní zpracování

Při základním nastavení hyperparametrů bylo v případě predikce bez informací o kanálu objeveno silné **přeučení** modelu Zatímco na trénovacích datech bylo dosaženo téměř dokonalé hodnoty $\mathbf{R^2 = 0.944}$, výsledek na testovacích datech byl pouhých $\mathbf{R^2 = 0.580}$ (Obrázek 26).

Obrázek 26: Výsledky RF – První model **Zdroj**: Visual Studio Code, Screenshot + carbon.now.sh

Ladění hyperparametrů tady oproti SVR bylo nutností s cílem lepší generalizace. Zaměřeno bylo především na variace počtů rozhodovacích stromů a jejich hloubku. Pomocí ParameterGrid byla nalezena kombinace hyperparametrů, která dokázala značně snížit rozdíl mezi **R^2** a dosáhnout výborné generalizace (Tabulka 16).

U modelu „**Žádný kanál**“ to ale vedlo k snížení predikčního výkonu pod výkon SVR a v převodu **R^2** na procenta dokázal zachytit jen **44.5 %** variability v počtu zhlédnutí. Naopak lepších výsledků dosáhl model „**Kanál**“, který vysvětlil **83.8 %** variability.

Tabulka 16: RF – ParametricGrid a nejlepší modely **Zdroj**: Vlastní zpracování

Random Forest / ParameterGrid						Nejlepší kombinace:	
Hyperparametr	**Kombinované hodnoty**					**ŽÁDNÝ KANÁL**	**KANÁL**
n_estimators	100	200	300	350		200	100
max_depth	None	3	4	5	10	5	10
min_samples_leaf	1	2	5			2	1
bootstrap	True	False				True	True
min_samples_split	2	5	10			10	10
max_features	sqrt	Log2	None			None	None
-				**R^2_train:**		0.484	0.910
				R^2_test:		**0.445**	**0.838**
				R^2_diff:		0.038	0.072
				Score:		0.406	0.766

Z hlediska hyperparametrů každému modelu sedělo trochu jiné nastavení. „**Žádný kanál**“ potřeboval 200 stromů s maximální hloubkou 5, mezitím co modelu „**Kanál**“ stačilo stromů pouze 100, ale využil lépe maximální hloubku 10.

3.4.6 Ladění XGBoost

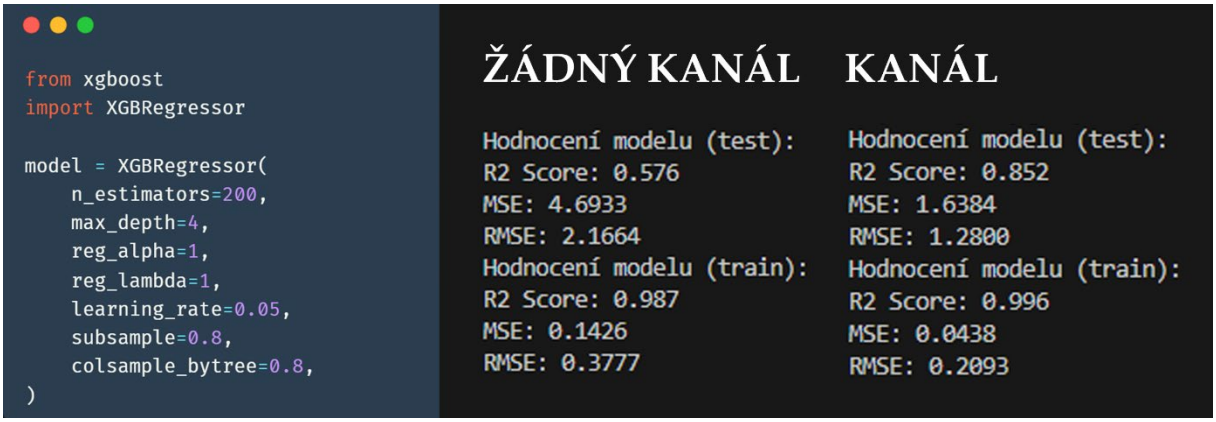
Výkonný boosting algoritmus **XGBoost** je brán jako standardní přístup k predikci popularity zejména na sociálních sítích díky své vysoké přesnosti, rychlosti trénování a optimalizaci skrze paralelizace výpočtů a regulace složitosti modelu. Ve studiích dosahuje dobrých výsledků [39], a právě proto byl použit jako další predikční model.

Klíčové hyperparametry (Tabulka 17) jsou u metody XGBoost díky jejímu základu v rozhodovacích stromech podobné jako u Random Forest. Také se ladí pomocí počtu rozhodovacích stromů a jejich maximální hloubce. Bonusem je například nastavitelná L1 a L2 regularizace nebo rychlost učení modelu z jednotlivých stromů.

Tabulka 17: XGBoost – Hyperparametry **Zdroj**: Vlastní zpracování

Hyperparametr	Popis	Typy hodnot
n_estimators	Počet stromů (iterací boosting procesu)	200, 300, 350, 400
max_depth	Maximální hloubka stromů.	None, 5, 10, 20
reg_alpha	L1 regularizace	0, 1
reg_lambda	L2 regularizace	0, 1
learning_rate	Jak moc se má model z každého stromu učit	0.01, 0.1, 0.3
subsample	Podíl trénovacích vzorků použitých pro každý strom	0.5, 0.8, 1.0
colsample_bytree	Podíl vstupních proměnných použitých pro každý strom	0.5, 0.8, 1.0

Po vzoru RandomForest i XGBoost ve svém základním nastavení hyperparametrů vykazoval výrazné přeučení na datech bez informací o kanálu. Zatímco na trénovacích datech bylo dosaženo perfektní hodnoty $\mathbf{R^2 = 0.987}$, se výsledek na testovacích datech dostal pouze na $\mathbf{R^2 = 0.576}$ (Obrázek 27).



Obrázek 27: Výsledky XGBoost – První model **Zdroj**: Visual Studio Code, Screenshot + carbon.now.sh

Ladění hyperparametrů probíhalo podobně jako RandomForest. Mimo počet stromů a jejich hloubky byly vyzkoušeny i kombinace rychlého a pomalého učení, využití regularizací a podílů trénovacích vzorků a vstupních proměnných v každém stromu. Tak byla nalezena kombinace hyperparametrů, která dokázala značně snížit rozdíl mezi **R^2** a dosáhnout vynikající generalizace (Tabulka 18).

Metoda XGBoost si vedla velmi dobře nejen z hlediska predikčního výkonu, ale také ve schopnosti generalizace. U obou modelů došlo k výraznému zlepšení ve srovnání s modely SVR a Random Forest. Model „**Žádný kanál**" dokázal v převodu **R^2** na procenta zachytit více jak polovinu variability v počtu zhlédnutí, a to **54.7 %**. Generalizace byla sice horší, ale přeučení nebylo nijak výrazné. Model „**Kanál**" měl identický výkon s Random Forest a vysvětlil **83.9 %** variability.

Kombinované hodnoty byly vybrány tak, aby dokázaly zachytit větší spektrum možností. Počet stromů byl navýšen oproti Random Forest z důvodu **learning_rate**, který v případě pomalejšího učení potřebuje vyšší počet stromů.

Tabulka 18: XGBoost – ParametricGrid a nejlepší modely **Zdroj**: Vlastní zpracování

XGBoost / ParameterGrid						Nejlepší kombinace:	
Hyperparametr	**Kombinované hodnoty**					**ŽÁDNÝ KANÁL**	**KANÁL**
n_estimators	200	300	350	400	500	200	200
max_depth	None	3	4	5	10	3	3
learning_rate	0.01	0.1	0.3			0.1	0.1
subsample	0.8	1				0.8	0.8
colsample_bytree	0.8	1				0.8	1.0
reg_alpha	1	0				0	1
reg_lambda	1	0				1	1
-				**R^2_train:**		0.639	0.870
				R^2_test:		**0.547**	**0.839**
				R^2_diff:		0.092	0.032
				Score:		0.454	0.807

Z hlediska hyperparametrů se v obou sloupcích osvědčila kombinace průměrného počtu stromů (**n_estimators = 200**) relativně nízké hloubky stromů (**max_depth = 3**), a průměrná rychlost učení modelu (**learning_rate = 0.1**). Regularizace (**reg_alpha** a **reg_lambda**) také nakonec pomohly k lepším výsledkům. Model tímto nastavením dokázal zabránit přeučení a zároveň efektivně využít vztahy v datech.

3.4.7 Ladění neuronové sítě (MLP)

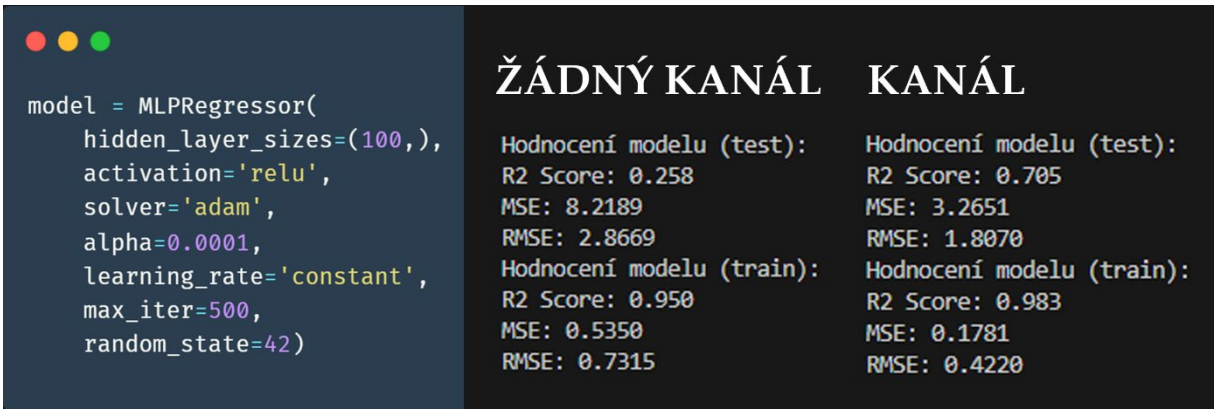
Jako poslední metoda byla zařazena predikce metodou **Multilayer Perceptron**. Jedná se o základní typ neuronové sítě, který je v predikci popularity používán skrze svoji flexibilitu a schopnost zachytit velmi složité a nelineární vztahy mezi vstupními proměnnými. MLP je považován za silný predikční model, avšak zároveň vyžaduje pečlivé ladění hyperparametrů [41].

MLP je stejně jako SVR citlivé na různorodost dat, proto i v tomto případě bylo důležité nejdříve data škálovat pomocí **StandardScaler**. Hyperparametry jsou jiné než u předešlých metod a ty nejzásadnější pro ladění se soustředí především na počet skrytých vrstev, aktivační funkce a počet iterací (epoch) viz Tabulka 19.

Tabulka 19: MLP – Hyperparametry **Zdroj**: Vlastní zpracování

Hyperparametr	Popis	Typy hodnot
hidden_layer_sizes	Počet a velikost skrytých vrstev (např. (100,) = 1 vrstva s 100 neurony)	(50,), (100,)
activation	Aktivační funkce ve skrytých vrstvách	'relu', 'tanh'
max_iter	Maximální počet iterací (epoch)	100, 300, 1000
solver	L2 regularizace	0.0001, 0.001, 0.01
learning_rate	Typ adaptace rychlosti učení v průběhu tréninku	0.01, 0.1, 0.3
alpha	L2 regularizace (čím vyšší, tím větší penalizace složitých vah)	0.5, 0.8, 1.0

U základního nastavení MLP se projevilo nejvýraznější přeučení ze všech dosud testovaných metod (Obrázek 28). Rozdíly mezi $\mathbf{R^2}$ jsou nevhodné i u modelu, který zná informace u kanálu. Rozdíl mezi testovacími a trénovacími daty bylo **r2_diff = 0.692**.



Obrázek 28: Výsledky MLP – První model **Zdroj**: Visual Studio Code, Screenshot + carbon.now.sh

Ladění hyperparametrů MLP bylo díky velké náročnosti na výkon provedeno jen menší množství kombinací. Testování proběhlo pouze na jedné skryté vrstvě o počtu 50 a 100 neuronů.

MLP v případě **„Žádný kanál“** modelu neuspěla ani po ladění. Přeučení zůstalo na vysoké úrovni, a tak byla neuronová síť pro tento dataset nevyhovující. Model „**Kanál**“ na tom byl s přeučením o trochu lépe, ale i tak nedokázal docílit odpovídající generalizace (Tabulka 20).

Tabulka 20: MLP – ParametricGrid a nejlepší modely **Zdroj**: Vlastní zpracování

Multilayer Perceptron / ParameterGrid					Nejlepší kombinace:	
Hyperparametr	**Kombinované hodnoty**				**ŽÁDNÝ KANÁL**	**KANÁL**
hidden_layer_sizes	(100,)	(50,)			(50,)	(50,)
activation	relu	tanh			relu	relu
max_iter	500	1000	1500		500	500
solver	adam				adam	adam
learning_rate	constant	adaptive			adaptive	const
alpha	0.01	0.001	0.0001		0.001	0.001
-				**R^2_train:**	0.864	0.941
				R^2_test:	**0.348**	**0.707**
				R^2_diff:	0.516	0.235
				Score:	-0.169	0.472

Tímto bylo možné najít nejlepší model predikce popularity videí, čemuž se věnuje následující kapitola.

3.5 Zhodnocení a nasazení nástroje

Po dokončení ladění hyperparametrů a vyhodnocení jednotlivých predikčních modelů následovalo závěrečné srovnání jejich výkonnosti. Porovnávány jsou obě varianty zvlášť a jsou hodnoceny nejvyšším **r2_test** s přihlídnutím na nejnižší **r2_diff**. Nejlepší modely byly srozumitelně interpretovány pomocí **R^2** a **MSE**.

Pro vyhodnocení důležitých proměnných a jejich interpretaci bylo využito techniky **SHAP (SHapley Additive exPlanations)**, která každé proměnné přidělují číslo (tzv. SHAP hodnotu), které ukazuje, o kolik daná proměnná posunula predikci směrem nahoru nebo dolů oproti průměru [48]. **SHAP graf** ukazuje na ose X sílu dopadu proměnné na predikci. Barva bodu určuje, jestli je hodnota konkrétního vzorku vysoká (červená) nebo nízká (modrá).

3.5.1 Nejlepší „Žádný kanál" predikční model

V této části byly porovnány výsledky všech testovaných algoritmů na datech bez informací o konkrétním kanálu.

Tabulka 21: Výběr nejlepších "Žádný kanál" modelů **Zdroj**: Vlastní zpracování

Metoda	R^2_train	**R^2_test**	R^2_diff
Lineární regrese	0.396	0.394	0.002
SVR	0.551	0.466	0.085
RandomForest	0.484	0.445	0.038
XGBoost	0.639	**0.547**	0.092
MLP	0.864	0.348	0.516

Z výsledků v Tabulce 21 vyplývá, že nejlepších výsledků dosáhl model **XGBoost**, který jako jediný překonal hranici **R^2 = 0.547** na testovacích datech a zároveň si udržel akceptovatelnou míru přeučení. **RandomForest** sice splňoval dobrou generalizaci, ale predikční síla byla oproti XGBoost horší.

Z druhé strany model **MLP** sice měl dobrý výkon na trénovacích datech, ale kvůli výraznému přeučení nebyla k predikci vhodná. K tomu pravděpodobně došlo z důvodu nedoladěného nastavení hyperparametrů. **SVR** sice dosáhla lepšího **R^2**, ale díky slabší generalizaci dat je dokonce horší než lineární regrese.

3.5.2 Nejlepší „Kanál“ predikční model

V této části byly porovnány výsledky všech testovaných algoritmů na datech, které již obsahovaly informace o kanálu uživatele, který video nahrál.

Tabulka 22: Výběr nejlepších "Kanál" modelů **Zdroj**: Vlastní zpracování

Metoda	R^2_train	**R^2_test**	R^2_diff
Lineární regrese	0.719	0.728	0.009
SVR	0.814	0.777	0.036
RandomForest	0.910	0.838	0.072
XGBoost	0.870	**0.839**	0.032
MLP	0.941	0.707	0.235

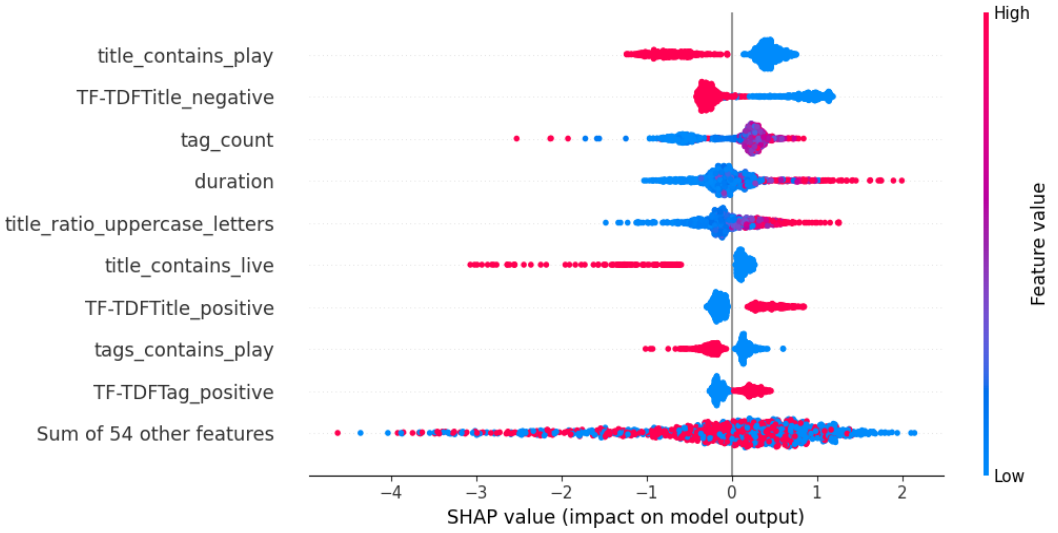
Z výsledků v Tabulce 22 vyplývá, že nejlepšího skóre opět dosáhl model **XGBoost**, který si i při vyšším predikčním výkonu na testovacích datech **R^2 = 0.839** zachoval velmi nízké přeučení, což značí výbornou schopnost generalizace.

Velmi dobře si vedl také **RandomForest**, jehož výkon na testovacích datech byl téměř stejný, ale kvůli většímu přeučení dosáhl nižšího skóre. Třetí nejlepší metodou byl **SVR**, jehož výkon byl rovněž stabilní, ale oproti stromovým metodám s nižší predikční silou.

MLP i zde dosáhlo vysokého **R^2** na trénovacích datech, ale znovu selhalo na datech testovacích, což potvrzuje problém s přeučením. **Lineární regrese** tentokrát dosáhla relativně dobrého výsledku, což může být způsobeno tím, že po přidání atributu s konkrétní informací o kanálu se některé vztahy v datech staly více lineárními a lépe predikovatelnými.

3.5.3 Důležité proměnné podle metody SHAP

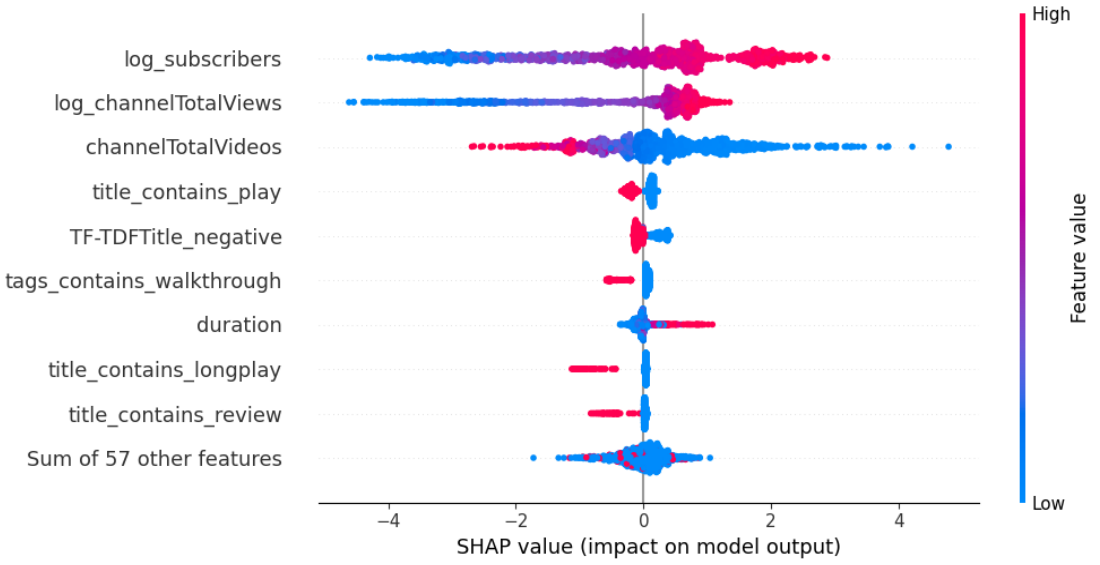
XGBoost „**Žádný kanál**“ model se dle analýzy SHAP grafu nejvíce opíral o přítomnost negativně korelujících slov, kdy jejich přítomnost často značilo menší popularitu (Obrázek 29). Také se opíral o počet tagů, délku videa a poměru velkých písmen v názvu videa, kdy se vyšší hodnoty vyskytovaly spíše u populárnějších videí.



Obrázek 29: SHAP graf (Žádný kanál)

Zdroj: Vlastní zpracování

XGBoost „**Kanál**“ model se nejvíce opíral o přítomnost právě už zpřístupněných proměnných (Obrázek 30). Dominantní proměnná byl počet odběratelů a počet celkových zhlédnutí kanálu. Obě tyto proměnné měly očekávaný dopad, kdy větší hodnota značila i větší počet zhlédnutí. Počet vydaných videí měla také velký dopad na predikci, ovšem naopak větší počet videí většinou značil menší počet zhlédnutí.



Obrázek 30: SHAP graf (Kanál)

Zdroj: Vlastní zpracování

3.5.4 Porovnání nejlepších modelů

Na základě vybraných nejúspěšnějších modelů z obou datasetů byly vypsány základní regresní metriky **R^2** a **RMSE**. Tyto hodnoty poskytují komplexnější pohled na predikční přesnost obou modelů. Původní hodnota RMSE byla vypočítána v logaritmickém měřítku vzhledem k transformaci cílové proměnné, proto byl vytvořen i sloupec s exponenciální transformací RMSE, který je lépe interpretovatelný v rámci skutečného počtu zhlédnutí.

Tabulka 23: Nejlepší modely **Zdroj**: Vlastní zpracování

Typ dat	Algoritmus	R^2	RMSE	RMSE (exp)
„Žádný kanál“	XGBoost	**0.547**	2.2459	9.49
„Kanál“	XGBoost	**0.839**	1.3359	3.82

Z výsledků v Tabulce 23 je patrné, že predikce popularity videí bez znalosti informací o kanálu je výrazně náročnější. Model „**Žádný kanál**“ dosáhl pouze **R^2 = 0.547**, což znamená, že dokázal vysvětlit jen něco málo přes polovinu variability v počtu zhlédnutí. V porovnání s modelem „**Kanál**“, který vysvětlil až **83.9 %** variability, je tento rozdíl zásadní.

Model využívající informace o kanálu tak vykazoval mnohem nižší **RMSE** oproti modelu bez těchto informací, což potvrzuje jeho vyšší predikční sílu i v reálném světě. Po exponenciální transformaci bylo zjištěno, že průměrná chyba predikce RMSE je v případě modelu „Žádný kanál“ až 2.5× vyšší. „**Kanál**“ je tak mnohem vhodnějším predikčním modelem.

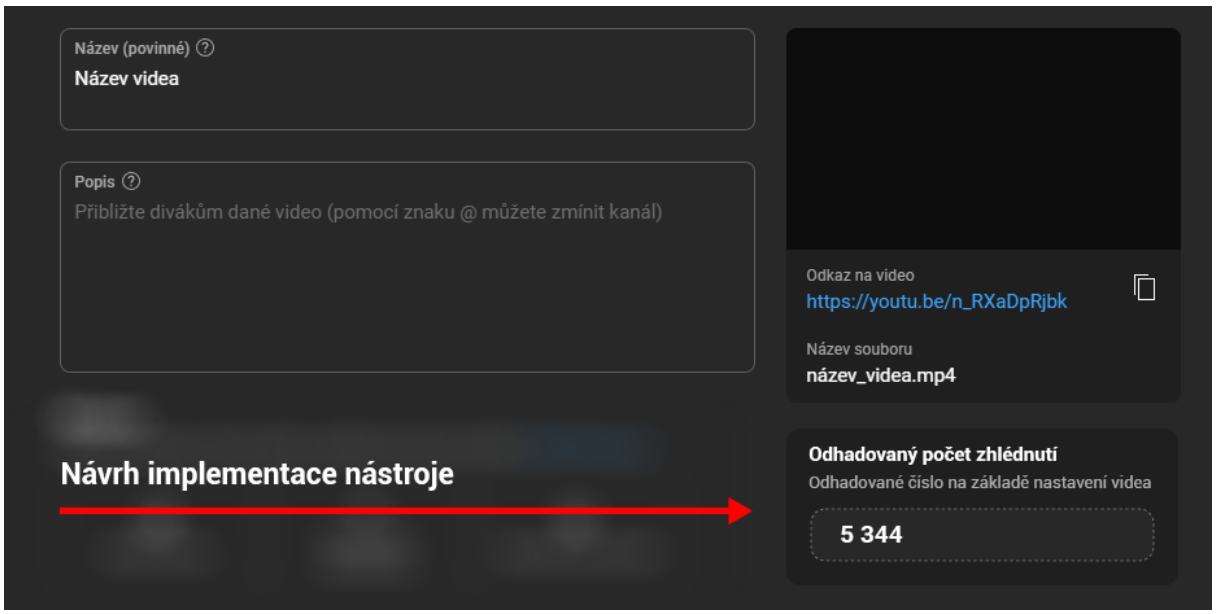
3.5.5 Příklad nasazení v praxi

V případě praktického využití by bylo možné nasadit model „**Kanál**“ přímo jako **podpůrný nástroj při nahrávání videa v rozhraní YouTube**. Tvůrci by to mohlo pomoct uvažovat nad výběrem názvu videa, ale také nad dnem, kdy ho bude vydávat nebo jaké tagy vložit k maximalizaci počtu zhlédnutí.

Výsledky predikce by bylo možné vložit přímo do hlavního rozhraní platformy (Obrázek 31), kde by se po nahrání videa zobrazila **odhadem predikovaná hodnota zhlédnutí** na základě zvolených metadat (např. názvu videa, délky nebo doby publikace).

Tato funkce by tak mohla sloužit jako **orientační ukazatel** a **motivační nástroj** pro zlepšení vstupních parametrů před samotným zveřejněním. Implementace by sebou ovšem nesla i povinnost neustálého přeučování modelů, jelikož trendy se velmi frekventovaně mění. Zároveň by bylo potřeba vytvořit několik dalších modelů, které pracují s jinými kategoriemi videí.

Ilustrativní návrh je na následujícím Obrázku 31.



Obrázek 31: Ilustrativní návrh implementace nástroje **Zdroj**: Vlastní zpracování + YouTube

4 KRITIKA A NÁVRHY K BUDOUCÍMU ZLEPŠENÍ

Nakonec byly dosaženy relativně slibné výsledky a vytrénovaný model vykazoval dobrý predikční výkon. Tato kapitola se věnuje diskusi nad omezeními navrženého modelu a návrhu možných zlepšení do budoucna. samotného projektu, uvědomění si jeho omezení a návrhu zlepšení do budoucna. V každé podkapitole jsou prezentována doporučení, jak model dále zlepšit.

4.1 Lepší zdroj dat

Veřejně dostupná **YouTube Data API** naneštěstí poskytuje jen velmi malé množství informací vhodných pro modelování. Predikční modely v rámci popularity videí často profitují například z časových řad a dostupných engagement metrik jako je CTR nebo Retention Rate, které tento zdroj běžnému analytikovi nezpřístupní.

Dalším problémem tohoto zdroje je jeho denní limit a neschopnost vyhledávat data opravdu náhodně. Tím má celý dataset tendence mít spíše pravošikmé rozdělení dat, jelikož API i bez filtrace stále vyhledává videa s jistou relevancí. Tento fakt znemožňuje použití vyrovnanějšího rozložení dat, ztěžuje stahování většího objemu vzorků a snižuje schopnost modelů učit se správně.

Navrhoval bych proto využití **třetích stran**, které mohou obsahovat podrobnější data o videích na platformě YouTube. Mezi tyto stránky může patřit například SocialBlade nebo Tubular Labs. Možností je i získání dat od velkého a různorodého množství tvůrců, kteří mají své osobní detailnější statistiky přístupné a stažitelné na svém YouTube účtu v dashboardu. Tak by bylo dosaženo lepších engagement metrik nebo dokonce časových řad.

Také je možností využití tzv. **web scraping** techniky, která funguje na principu automatizovaného sběru dat z webových stránek na základě vlastních kritérií a uložení do datasetu [49]. Tento přístup by mohl obejít omezený způsob stahovaní dat z API a zajistit lepší rozložení dat. Je však třeba brát v potaz i právní rámec, jelikož některé stránky tuto techniku výslovně zakazují.

4.2 Pokročilejší zpracování dat

Během zpracování dat nebyl využit plný potenciál získaných dat kvůli složitějšímu procesu technik, které by na to musely být použity. YouTube API například poskytla odkazy na náhledové obrázky videí, které mohly pomoct v predikci popularity. K tomu by ale bylo potřeba vytvoření automatizovaného stahování obrázků a následná extrakce vizuálních atributů pomocí analytických nástrojů. Možností bylo i použití předtrénované konvoluční neuronové sítě, která dokáže pochopit například výraznou emoci na obrázku.

V příštích analýzách by proto bylo vhodné zahrnout zpracování vizuálních prvků náhledového obrázku k vytvoření velmi zásadních proměnných, jelikož lidé se často rozhodují kliknout na video kvůli jeho vizuálnímu přesvědčení, nikoliv názvu videa.

4.3 Podrobnější ladění hyperparametrů

Ladění hyperparametrů během modelování využilo velké množství kombinací pomocí ParametricGrid. Tento přístup přinesl uspokojivé výsledky, ale bylo možné celý proces ještě více optimalizovat pomocí více testovaných hodnot v jednotlivých hyperparametrech nebo **RandomSearch**, který vyhledává kombinace zcela náhodně a může přijít na lepší kombinaci, která mohla být díky ručnímu zadávání hodnot opomenuta. Tento přístup by byl důkladnější, ale vyžadoval by mnohem více výpočetního výkonu a času, a to především v případě metody RandomForest nebo neuronové sítě.

4.4 Shrnutí

Těmito všemi návrhy by mohlo v budoucích analýzách dosaženo lepších výsledků než doposud. Kvalita datasetu je základní stavební kámen jakékoliv analýzy a jeho důkladné zpracování může ukázat vzory, které by jinak byly skryty. Špatně nastavené hyperparametry modelu ovšem můžou způsobit, že tyto vzory budou přehlédnuty.

ZÁVĚR

Cílem této diplomové práce bylo prozkoumat možnosti predikce popularity videí na sociálních sítích pomocí strojového učení. K analýze po vzoru rámce CRISP-DM byla vybrána díky dostupnějším datům platforma YouTube. Na základě porozumění cíli vytvoření podpůrného nástroje predikující přibližný počet zhlédnutí byly použity reálná data získaná přes YouTube Data API.

Získán tak byl časově relevantní dataset proměnných, kterou tvořila jak metadata o videích, tak informace o kanálech. Ty byly následně pomocí několika metod zpracovány za cílem odhalení vzorů v datech. K testování predikcí byly využity ověřené metody jako SVR, Random Forest, XGBoost a MLP, které nakonec byly hodnoceny pomocí regresních metrik R^2, RMSE a MAE.

Výsledky na získaných datech ukázaly, že bez informací o kanálu je predikce výrazně méně přesná, což naznačilo, že právě historický výkon kanálu hraje důležitou roli v určování budoucí popularity videí. Ani ty ale nebyly zcela přesné, ale v rámci videí, u kterých je popularita ovlivněna velkým množstvím vnějších faktorů, se jednalo o uspokojivý výsledek.

Nejúspěšnějším modelem se tak stal pro svojí generalizaci a predikční sílu model XGBoost, který využíval i informace o daném kanálu. Právě ten byl poté zhodnocen jako vhodný k implementaci a vybrán jako základ podpůrného nástroje. V rámci sebereflexe a celkového zhodnocení analýzy byl nakonec celý proces diskutován a byly uvedeny návrhy ke zlepšení pro budoucí obdobné projekty.

Celkově lze tak říct, že predikce popularitu videí na základě počtu zhlédnutí je možná, ale bez širšího kontextu zůstává výzvou, a to i za použití metadat o kanálu tvůrce. Do budoucna by mohlo pomoct model zlepšit použitím lepšího datasetu, který má větším počet vzorků s dostupnými časovými řadami nebo rozšířením o analýzu náhledových obrázků.

Na základě provedeného postupu lze konstatovat, že cíl práce byl splněn. Byly identifikovány klíčové atributy ovlivňující popularitu videí, vyhodnoceny vhodné predikční metody a navržen model využitelný pro predikci počtu zhlédnutí na platformě YouTube. Výsledky zároveň poukázaly na limity predikce popularity a nastínily možnosti jejího dalšího zlepšení.

SEZNAM POUŽITÉ LITERATURY

[1] HAVLOVÁ, Jaroslava. sociální síť. In: *KTD: Česká terminologická databáze knihovnictví a informační vědy (TDKIV)* [online]. Praha : Národní knihovna ČR, 2003- [cit. 2015-07-08]. Dostupné z: http://aleph.nkp.cz/F/?func=direct&doc_number=000015947&local_base=KTD.

[2] STATISTA. Number of worldwide social network users 2017-2027. In: *Statista* [online]. Hamburg: Statista, 2024 [cit. 2025-02-06]. Dostupné z: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/

[3] STATISTA. Predicted number of social network users in Czechia 2019-2029. In: *Statista* [online]. Hamburg: Statista, 2024 [cit. 2025-02-06]. Dostupné z: https://www.statista.com/statistics/568879/predicted-number-of-social-network-users-in-czechia/

[4] WAGNER, Pavel. *Vývoj youtuberství a streamerství v České republice*. Hradec Králové: Univerzita Hradec Králové, Pedagogická fakulta, 2021. Diplomová práce. Dostupné z: https://theses.cz/id/31ywcr/STAG96661.pdf.

[5] BURGRESS, Jean a Joshua GREEN. *YouTube: Online Video and Participatory Culture.* 2. vyd. Cambridge: Polity Press, 2018. ISBN 978-1-5095-2599-0.

[6] SHAH, Ayush. *YouTube Partner Program to Better Protect Content Creators.* Bangkok: Siam University, 2018. Dostupné z: https://e-research.siam.edu/wp-content/uploads/2020/07/BBA-Finance-and-Banking-2018-Coop-YouTube-Partner-Program-to-Better-Protect-Content-Creators-compressed.pdf.

[7] ČUPA, Matěj. Nenápadný youtuber vydělává stovky milionů korun a má nejvíce odběratelů v historii. *CC.cz* [online]. 2014 [cit. 2025-02-17]. Dostupné z: https://cc.cz/nenapadny-youtuber-vydelava-stovky-milionu-korun-a-ma-nejvice-odberatelu-v-historii

[8] KAYE, Daniel B. V., Laura KIM a Yu-Ri HUH. Undergraduate's Perceptions of TikTok's Effects on the COVID-19 Pandemic: A Mixed-Methods Study. *JMIR Public Health and Surveillance* [online]. 2021, roč. 7, č. 4, e29029 [cit. 2025-02-18]. DOI: 10.2196/29029. Dostupné z: https://pmc.ncbi.nlm.nih.gov/articles/PMC7989330/

[9] Chapple, C. (2020, duben). *TikTok překročil 2 miliardy stažení po nejlepším čtvrtletí pro jakoukoli aplikaci vůbec*. Sensor Tower. Dostupné z https://sensortower.com/blog/tiktok-downloads-2-billion

[10] KOLLER, Daniel, Jacek GRYCEK, Julian GÜNTHER a Björn SCHULLER. *Shorts on the Rise: Assessing the Effects of YouTube Shorts on Long-Form Video Content.* 2024. Preprint dostupný z arXiv: https://arxiv.org/pdf/2402.18208.

[11] LI, Chenyu; LIU, Jun a OUYANG, Shuxin. Characterizing and Predicting the Popularity of Online Videos. *IEEE Access* [online]. 2016, 4, s. 1630–1641 [cit. 2025-04-26]. ISSN 2169-3536. Dostupné z: https://www.researchgate.net/publication/300080441_Characterizing_and_Predicting_the_Popularity_of_Online_Videos

[12] SUCHÁ, Jaroslava, Eva AIGELOVÁ, Helena PIPOVÁ, Miroslav CHARVÁT, Martin DOLEJŠ, Nikol VÁCLAVKOVÁ a Terezie BABILONOVÁ. *Užívání internetu, sociálních sítí a digitálních her u adolescentů: Teoretická východiska, diagnostika a strategie intervence*. 1. vyd. Olomouc: Vydavatelství Univerzity Palackého, 2024. ISBN 978-80-244-6437-4. Dostupné také z: https://www.researchgate.net/publication/379143564_Uzivani_internetu_socialnich_siti_a_digitalnich_her_u_adolescentu_Teoreticka_vychodiska_diagnostika_a_strategie_intervence

[13] CHEN, Guandan; KONG, Qingchao; XU, Nan; MAO, Wenji. NPP: A Neural Popularity Prediction Model for Social Media Content. *Neurocomputing*, 2019, 333, 221-230. ISSN 0925-2312. DOI: 10.1016/j.neucom.2018.12.101

[14] ZHANG, Aston; LIPTON, Zachary C.; LI, Mu a SMOLA, Alexander J. Introduction to Recommender Systems. In: *Dive into Deep Learning* [online]. Dostupné z: https://d2l.ai/chapter_recommender-systems/recsys-intro.html [cit. 2025-02-18]

[15] AGGARWAL, Charu C. *Recommender Systems: The Textbook*. Cham: Springer, 2016. ISBN 978-3-319-29658-3.

[16] BARKER, Melissa S.; BARKER, Donald; BORMANN, Nicholas F.; ROBERTS, Mary Lou a ZAHAY, Debra L. *Social Media Marketing: A Strategic Approach*. Second edition. Boston: Cengage Learning, 2017. ISBN 978-1305502758.

[17] WU, Siqi, Marian-Andrei RIZOIU a Lexing XIE. *Beyond Views: Measuring and Predicting Engagement in Online Videos*. arXiv preprint arXiv:1709.02541 [online]. 2018 [cit. 2025-02-21]. Dostupné z: https://arxiv.org/abs/1709.02541

[18] ADVERITY. Top 10 Video Marketing Metrics you Should Actually Track [online]. *Adverity*, 2023 [cit. 2025-04-26]. Dostupné z: https://www.adverity.com/blog/top-10-video-marketing-metrics-you-should-actually-track

[19] ROUSIDIS, Dimitrios, KOUKARAS, Paraskevas a TJORTJIS, Christos. Social Media Prediction: A Literature Review. *Multimedia Tools and Applications*, 2020, 79(9), 6279-6311. DOI: 10.1007/s11042-019-08390-7.

[20] JIANG, Fei, ZHANG, Conghui, DONG, Yulan, et al. Supervised Machine Learning: A Brief Primer. *Clinical and Translational Medicine* [online]. 2020, 9(1), 1–8 [cit. 26. 4. 2025]. Dostupné z: https://doi.org/10.1186/s40169-020-00296-6

[21] PORTÁL MATEMATICKÁ BIOLOGIE. Typy dat [online]. Dostupné z: https://portal.matematickabiologie.cz/index.php?pg=aplikovana-analyza-klinickych-

a-biologickych-dat--analyza-a-management-dat-pro-zdravotnicke-obory--data-jejich-popis-a-vizualizace--typy-dat [cit. YYYY-MM-DD].

[22] ROJO-ECHEBURÚA, Ana. What Is One Hot Encoding and How to Implement It in Python [online]. *DataCamp*, 26. června 2024 [cit. 17. dubna 2025]. Dostupné z: https://www.datacamp.com/tutorial/one-hot-encoding-python-tutorial

[23] CHEN, Yen-Liang a CHANG, Chia-Ling. Early Prediction of the Future Popularity of Uploaded Videos. *Expert Systems with Applications*, 2019, 133, s. 59–74. ISSN 0957-4174. DOI: 10.1016/j.eswa.2019.05.047.

[24] MANNING, Christopher D., RAGHAVAN, Prabhakar a SCHÜTZE, Hinrich. *Introduction to Information Retrieval.* Cambridge University Press, 2008. ISBN 978-0-521-86571-5.

[25] DEVLIN, Jacob, CHANG, Ming-Wei, LEE, Kenton a TOUTANOVA, Kristina. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* [online]. 2019 [cit. 2025-04-15]. Dostupné z: https://arxiv.org/abs/1810.04805

[26] CHATZOPOULOU, Gloria, SHENG, Cheng a FALOUTSOS, Michalis. *A First Step Towards Understanding Popularity in YouTube*. Riverside: University of California, 2010.

[27] DALMORO, Bruna Martini a Soraia Raupp MUSSE. Using Visual Features and Early Views to Classify the Popularity of Facebook Videos. *Journal of the Brazilian Computer Society*, 2022, roč. 28, č. 1, s. 52–58. DOI: 10.5753/jbcs.2022.2216

[28] JAMWAL, Aman. *Comparison of Pearson and Spearman Correlation Coefficients* [online]. Analytics Vidhya, 2021 [cit. 2025-04-24]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/03/comparison-of-pearson-and-spearman-correlation-coefficients

[29] STATSTUTOR. *Spearman's Correlation* [online]. [cit. 2025-04-26]. Dostupné z: https://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf

[30] FREEDMAN, David A. *Statistical Models: Theory and Practice*. 2nd ed. Cambridge: Cambridge University Press, 2009. ISBN 978-0-521-73165-0.

[31] GUPTA, P., GOEL, A., LIN, J., SHARMA, A., WANG, D. a ZADEH, R. Wtf: The Who to Follow Service at Twitter. *Proceedings of the 22nd International Conference on World Wide Web*. 2013. s. 505-514. Dostupné z: https://web.stanford.edu/~rezab/papers/wtf_overview.pdf

[32] SHAIKH, Usman, GORAYA, Muhammad Usman, REHMAN, Muhammad Adeel a IQBAL, Tanveer. Forecasting Social Media Engagement: An ARIMA-based Approach. *International Journal of Advanced Computer Science and Applications*, 2022. ISSN 2156-5570. DOI: 10.14569/IJACSA.2022.0131013

[33] SZABO, G. a HUBERMAN, B. A. Predicting the Popularity of Online Content. *arXiv preprint*, 2008. arXiv:0811.0405. Dostupné z: https://arxiv.org/abs/0811.0405

[34] TRZCIŃSKI, Tomasz; ROKITA, Przemysław. Predicting Popularity of Online Videos using Support Vector Regression. *IEEE Transactions on Multimedia*, 2017, roč. 19, č. 11, s. 2561-2570.

[35] KOTSIANTIS, Sotiris B. Decision Trees: A Recent Overview. *Artificial Intelligence Review* [online]. 2013, roč. 39, č. 4, s. 261–283 [cit. 24. 3. 2025]. ISSN 0269-2821. Dostupné z: https://doi.org/10.1007/s10462-011-9272-4

[36] BREIMAN, Leo. Random Forests. *Machine Learning*. 2001, roč. 45, č. 1, s. 5–32. ISSN 0885-6125. Dostupné z: https://doi.org/10.1023/A:1010933404324

[37] BATTA, Himanshu a MURTHY, D.N.P. Predicting Popularity of YouTube Videos using Viewer Engagement Features. *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2022, s. 1036–1040. ISBN 978-1-6654-0286-2. Dostupné z: https://ieeexplore.ieee.org/document/9734220

[38] CHEN, Tianqi a Carlos GUESTRIN. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining* [online]. 2016, s. 785–794 [cit. 2025-03-21]. DOI: 10.1145/2939672.2939785. Dostupné z: https://arxiv.org/pdf/1603.02754.pdf

[39] NISA, M. U., MAHMOOD, D., AHMED, G., KHAN, S., MOHAMMED, M. A. a DAMASEVIČIUS, R. Optimizing Prediction of YouTube Video Popularity Using XGBoost. *Electronics* [online]. 2021, roč. 10, č. 23, s. 2962 [cit. 2025-03-21]. ISSN 2079-9292. Dostupné z: https://doi.org/10.3390/electronics10232962

[40] AGGARWAL, Charu C. *Neural Networks and Deep Learning: A Textbook*. Cham: Springer, 2018. ISBN 978-3-319-94463-0. Dostupné z: https://doi.org/10.1007/978-3-319-94463-0

[41] SHE, Wei, Li XU, Huibo XU, Xiaoqing ZHANG, Yue HU a Zhao TIAN. *Multilayer Perceptron Based on Joint Training for Predicting Popularity.* Cham: Springer, 2020. Lecture Notes in Computer Science, vol. 12240. ISBN 978-3-030-57880-0. Dostupné z: https://www.researchgate.net/publication/344010796_Multilayer_Perceptron_Based_on_Joint_Training_for_Predicting_Popularity

[42] ZHOU, Tianqi, ZHANG, Yujie a WANG, Ke. Multi-branch LSTM encoded latent features with CNN-LSTM for Youtube popularity prediction. *Scientific Reports* [online]. 2025, roč. 15, č. 1, s. 1–12 [cit. 2025-03-29]. DOI: 10.1038/s41598-025-86785-9. Dostupné z: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11747267/

[43] SCIKIT-LEARN. *Model Evaluation* [online]. [cit. 2025-03-30]. Dostupné z: https://scikit-learn.org/stable/modules/model_evaluation.html

[44] IBM Corp. *CRISP-DM 1.0: Step-by-step Data Mining Guide* [online]. IBM, 2000 [cit. 2025-04-11]. Dostupné z: https://www.ibm.com/docs/it/SS3RA7_18.3.0/pdf/ModelerCRISPDM.pdf

[45] GOOGLE. *Google API Client Library for Python Documentation* [online]. [cit. 2025-04-02]. Dostupné z: https://googleapis.github.io/google-api-python-client/docs/

[46] HYPEAUDITOR. *YouTube Engagement Rate Calculator* [online]. [cit. 2025-04-12]. Dostupné z: https://hypeauditor.com/free-tools/youtube-engagement-calculator/

[47] GHOLAMY, Afshin, KREINOVICH, Vladik a KOSHELEVA, Olga. *Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation*. El Paso: University of Texas at El Paso, 2018. Departmental Technical Report UTEP-CS-18-09. Dostupné z: https://scholarworks.utep.edu/cs_techrep/1209/

[48] LUND BERG, Scott M. a LEE, Su-In.
A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems* [online]. 2017, 30.
Dostupné z:
https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[49] GEONODE. *What is Web Scraping*? [online]. 2023 [cit. 2025-04-22]. Dostupné z: https://geonode.com/blog/what-is-web-scraping

PŘÍLOHY

Příloha A: USB Flash disk s veškerými materiály

Na paměťovém USB disku jsou uloženy veškeré materiály, které nebylo možné nebo vhodné vkládat přímo do diplomové práce.

Těmito materiály jsou soubory především Python kódů, které byly použity při sběru a zpracování dat, modelování predikčních modelů a jejich hodnocení. Na USB jsou také SHAP grafy v lepším rozlišení nebo již zpracovaná vlastní reálná data.

Příloha B: Kompletní datový slovník

Tabulka P1: Kompletní datový slovník

Název proměnné	Popis proměnné	Datový typ	Příklad obsahu
title	Název videa, titulek	Object	ROBLOX ART!
description	Popis videa	Object	Welcome at ...
hour_of_day	Hodina dne nahrání	Int64	0, 1, 2, 5, 10, 23
Log_views	Počet zhlédnutí (log)	Float64	13.39, 5.4
Log_likes	Počet lajků (log)	Float64	10.1, 8.33
Log_comments	Počet komentářů (log)	Float64	5.13, 1.14
Log_duration	Délka videa (log)	Float64	7.01, 12.0
Log_subscribers	Počet odběratelů (log)	Float64	15.3, 14.1
Log_channelTotalVideos	Počet videí na kanálu (log)	Float64	2.4, 1.54
Log_channelTotalViews	Počet zhlédnutí kanálu (log)	Float64	21.13, 11.34
source	Původní dataset vzorku	Object	1minRandom
tags	Tagy v metadatech videa	Object	Minecraft, tyler
engagement_rate	Metrika engagement rate	Float64	2.21
title_emoji_count	Počet emotikon v názvu	Int64	0, 1, 3, 6
title_word_count	Počet slov v názvu	Int64	1, 2, 5, 10
title_exclamations	Počet vykřičníků v názvu	Int64	3, 1, 0
title_has_question	Obsahuje název otazník?	Int64	1, 0
title_has_number	Obsahuje název číslo?	Int64	1, 0
tag_count	Počet tagů v metadatech	Int64	0, 2, 5, 10
title_contains_*slovo*	Je v názvu dané slovo?	Int64	1, 0
tags_contains_*slovo*	Má video daný tag?	Int64	1, 0
TF-TDFTitle_positive	Pozitivně korelující slovo?	Int64	1, 0
TF-TDFTitle_negative	Negativně korelující slovo?	Int64	1, 0
TF-TDFTag_positive	Pozitivně korelující tag?	Int64	1, 0
TF-TDFTag_negative	Negativně korelující tag?	Int64	1, 0
day_*den v týdnu*	Video nahráno v tento den?	Int64	1, 0
title_ratio_ uppercase_letters	Poměr velkých písmen v názvu videa	Float64	0.88, 0.15
title_token_count_ unique	Počet unikátních slov v názvu videa	Int64	3, 1, 5
bert_sentiment_ numeric	Sentiment názvu videa	Int64	1–5