Použitie techník strojového učenia na dataset vín - Projekt

Lucia Lahučká

31. decembra 2024

Úvod

Tento projekt sa zameriava na analýzu dvoch datasetov týkajúcich sa portugalského vína "Vinho Verde", konkrétne na červené a biele varianty. Dátové sady obsahujú údaje o fyzikálno-chemických vlastnostiach vína a súvisiacich senzorických hodnoteniach. Tieto údaje môžu byť použité na úlohy klasifikácie alebo regresie, kde cieľovou premennou je kvalita vína, ktorá je hodnotená na stupnici od 0 do 10.

Dátové sady sú charakterizované nevyváženosťou tried, pričom väčšina vína je hodnotená ako "normálne", zatiaľ čo "vynikajúceälebo "zlé"vína sú v menšine. Tento nevyvážený charakter môže spôsobiť problémy pri aplikácii štandardných algoritmov strojového učenia, preto je dôležité vykonať analýzu neobvyklých vzorcov a detekciu odľahlých hodnôt.

Cieľom tejto práce je aplikovať rôzne techniky strojového učenia na tento dataset a analyzovať ich schopnosť predikovať kvalitu a typ vína na základe fyzikálno-chemických vlastností. Okrem toho bude vykonaná analýza dôležitosti jednotlivých vlastností vína a posúdenie výkonu rôznych algoritmov na úlohy klasifikácie a regresie.

V rámci tohto projektu sa zameriame na nasledujúce techniky strojového učenia:

- **Zhlukovanie (Clustering)**: Budeme skúmať, či sa vína dajú rozdeliť do zhlukov na základe ich fyzikálno-chemických vlastností. Táto technika môže odhaliť skryté vzory v dátach, ktoré nie sú priamo viditeľné v rámci klasifikácie.
- **Analýza hlavných komponent (PCA)**: Pomocou PCA sa pokúsime znížiť dimenzionalitu dát a identifikovať hlavné komponenty, ktoré naj-

lepšie vystihujú variabilitu v dátach. Táto technika nám pomôže lepšie pochopiť vzťahy medzi rôznymi vlastnosťami vína.

- **Klasifikácia typu vína**: Na základe fyzikálno-chemických vlastností sa budeme snažiť klasifikovať vína do dvoch kategórií: červené a biele. Tento úkol predstavuje binárnu klasifikáciu.
- **Klasifikácia podľa kvality vína**: Na základe senzorických dát a kvalittívnych tried budeme predikovať kvalitu vína, ktorá je hodnotená na stupnici od 0 do 10.
- **Regresia s optimalizáciou hyperparametrov (Grid Search)**: Na predikciu kvality vína použijeme regresné modely, pričom na optimalizáciu hyperparametrov týchto modelov využijeme metódu grid search.
- **Selektívna analýza dôležitých vlastností**: Vzhľadom na to, že nie všetky vstupné premenné môžu byť relevantné, použijeme techniky selekcie vlastností, aby sme identifikovali najdôležitejšie premenné pre predikciu kvality a typu vína.

Obsah

0 Informácie o atribútoch

Dataset pre červené víno obsahuje 1599 inštancií, zatiaľ čo pre biele víno je to 4898 inštancií. Každý dataset obsahuje 11 vstupných premenných, ktoré sa týkajú rôznych chemických vlastností vína.

Vstupné premenné (na základe fyzikálno-chemických testov) sú:

- Fixná kyslosť
- Volatilná kyslosť
- Kyselina citrónová
- Zvyškový cukor
- Chloridy
- Voľný oxid siričitý
- Celkový oxid siričitý
- Hustota
- pH
- Síran
- Alkohol

Výstupná premenná (na základe senzorických údajov) je kvalita vína (skóre medzi 0 a 10). Dané atribúty sú totožné ako aj pre biele, tak aj červené víno.

1 Odklon od pôvodného návrhu projektu

Pôvodný návrh projektu sa zameriaval na vytvorenie modelov strojového učenia pre predikciu a klasifikáciu kvality vína. Tento problém mal byť riešený dvoma hlavnými spôsobmi:

- **Regresia**: Predikcia kontinuálnej hodnoty kvality vína na stupnici od 0 do 10.
- **Klasifikácia**: Rozdelenie kvality vína do diskrétnych tried (napr. veľmi zlé, priemerné, vynikajúce).

Hoci tento návrh poskytoval pevný základ pre analýzu, počas realizácie projektu sme identifikovali príležitosti na jeho rozšírenie. Po hlbšej analýze dát a preskúmaní aktuálnych trendov v strojovom učení sme sa rozhodli integrovať do projektu ďalšie zaujímavé techniky a rozšíriť pôvodný rámec o nové prístupy, ktoré môžu priniesť hlbšie poznatky a zvýšiť presnosť modelov.

1.1 Rozšírenie návrhu projektu

Zatiaľ čo pôvodný návrh bol orientovaný primárne na regresiu a klasifikáciu, v rozšírenej verzii projektu sme sa rozhodli zahrnúť aj nasledujúce prístupy:

- **Zhlukovanie (Clustering)**
- **Analýza hlavných komponent (PCA)**
- **Selektívna analýza vlastností**
- **Experimentovanie s pokročilými algoritmami**

Toto rozšírenie projektu nám umožní nielen splniť pôvodné ciele, ale aj preskúmať nové perspektívy v analýze dát. Integrácia zhlukovania a redukcie dimenzionality pridáva exploratívny aspekt do projektu, zatiaľ čo selekcia vlastností a experimentovanie s pokročilými modelmi zlepšujú jeho praktickú hodnotu a robustnosť. Táto zmena zamerania reflektuje našu snahu nielen splniť požiadavky projektu, ale zároveň využiť potenciál dát na maximum a prispieť k lepšiemu pochopeniu problému predikcie kvality vína.

2 Regresia pre predikciu kvality vína

Cieľom regresnej analýzy bolo predikovať kvalitu vína ako spojitú hodnotu na stupnici od 0 do 10 na základe jeho fyzikálno-chemických vlastností. Použili sme model `RandomForestRegressor` v rámci pipeline, ktorý zahŕňal aj škálovanie vstupných dát a optimalizáciu hyperparametrov pomocou `GridSearchCV`. Celý proces pozostával z niekoľkých krokov:

2.1 Príprava dát

Datasety červeného a bieleho vína boli skombinované do jedného datasetu, pričom cieľová premenná (`quality`) bola oddelená od vstupných atribútov. Následne bol dataset rozdelený na trénovaciu (80%) a testovaciu (20%) množinu.

```
red_wine = pd.read_csv("wine+quality/winequality-red.csv", sep=
    ↪ ';')
white_wine = pd.read_csv("wine+quality/winequality-white.csv",
    ↪ sep=';')
wine_data = pd.concat([red_wine, white_wine], axis=0).
    ↪ reset_index(drop=True)

X = wine_data.drop(columns='quality') # Features
y = wine_data['quality'] # Target variable (quality)

X_train, X_test, y_train, y_test = train_test_split(X, y,
    ↪ test_size=0.2, random_state=42)
```

Listing 1: Príprava dát

2.2 Pipeline a optimalizácia hyperparametrov

Pipeline zahŕňal dva hlavné kroky:

- **Škálovanie atribútov**: Použitím `StandardScaler` boli všetky vstupné atribúty štandardizované na nulový priemer a jednotkovú odchýlku.
- **Regresný model**: Použitý bol `RandomForestRegressor`, pričom jeho hyperparametre boli optimalizované pomocou mriežkového vyhľadávania (`GridSearchCV`) na základe 5-násobnej krížovej validácie.

```
pipeline = Pipeline([
    ('scaler', StandardScaler()), # Scaling the features
    ('regressor', RandomForestRegressor(random_state=42)) #
        ↪ Regression model
])

# The parameter grid to optimize hyperparameters
param_grid = {
    'regressor__n_estimators': [100, 200, 300],
    'regressor__max_depth': [10, 20, None],
    'regressor__min_samples_split': [2, 5, 10],
    'regressor__min_samples_leaf': [1, 2, 4],
}

# Perform hyperparameter tuning
```

```
grid_search = GridSearchCV(pipeline, param_grid, cv=5, scoring=
    ↪ 'neg_mean_squared_error', n_jobs=-1, verbose=2)
grid_search.fit(X_train, y_train)
```

Listing 2: Optimalizácia hyperparametrov a fitovanie modelu

Optimalizované hyperparametre zahŕňali:

- Počet stromov (`n_estimators`),
- Maximálnu hĺbku stromov (`max_depth`),
- Minimálny počet vzoriek na rozdelenie uzla (`min_samples_split`),
- Minimálny počet vzoriek na vytvorenie listu (`min_samples_leaf`).

2.3 Výsledky a vyhodnotenie

Po optimalizácii modelu boli predikcie na testovacej množine porovnané s reálnymi hodnotami kvality vína. Model bol vyhodnotený pomocou nasledujúcich metrík:

- **Priemerná absolútna chyba (MAE)**: MAE = 0.435,
- **Priemerná kvadratická chyba (MSE)**: MSE = 0.369,
- **Koeficient determinácie ($\mathbf{R^2}$)**: $R^2 = 0.499$.

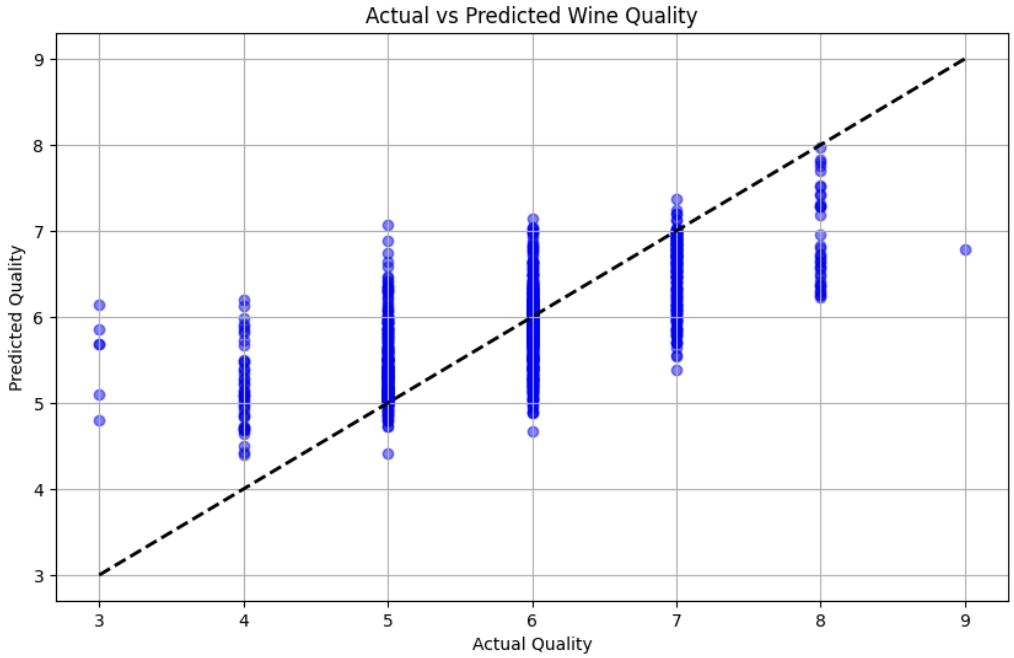
Najlepšie hyperparametre identifikované pomocou `GridSearchCV` boli:

- Počet stromov: 200,
- Maximálna hĺbka: None,
- Minimálny počet vzoriek na rozdelenie: 2,
- Minimálny počet vzoriek na list: 1.

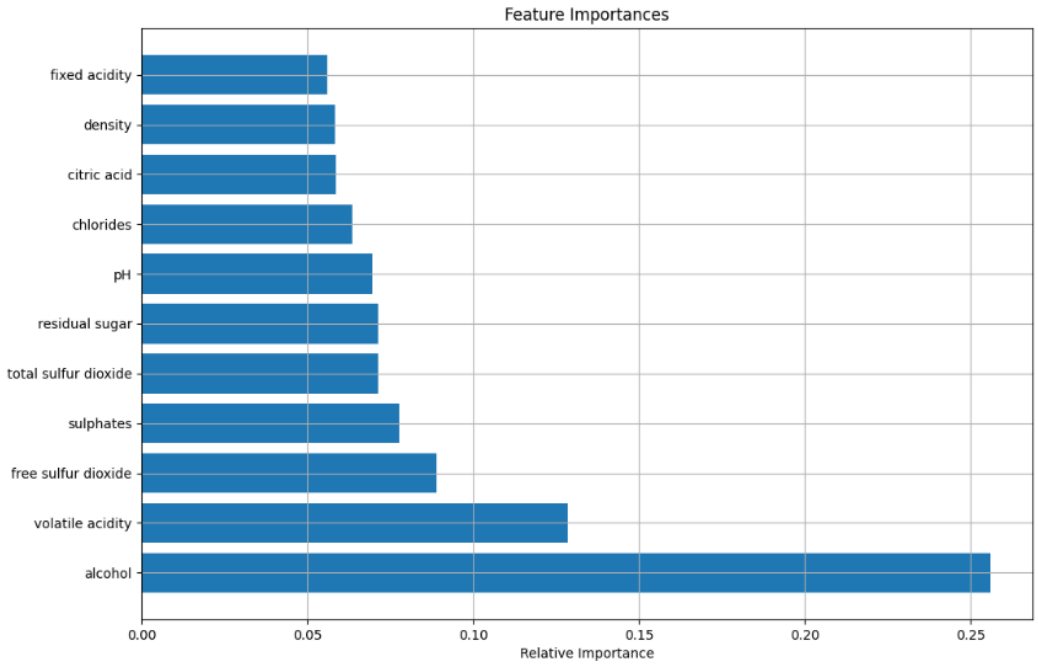
2.4 Vizualizácie

Model bol doplnený o vizualizácie pre lepšiu interpretáciu výsledkov:

- **Porovnanie reálnych a predikovaných hodnôt**: Scatter plot ukázal že väčšinu vín zaradil do kategórie kvality 5,6,7. Výsledky je možné vidieť na obrázku 1.
- **Dôležitosť atribútov**: Vizualizácia dôležitosti atribútov odhalila, že `alcohol` a `volatile acidity` mali najväčší vplyv na predikciu kvality vína, obrázok 2.



Obr. 1: Porovnanie reálnych a predikovaných hodnôt kvality vína použitím regresie.



Obr. 2: Dôležitosť atribútov vína pre metódu regresie.

3 Klasifikácia typu vína

Táto sekcia sa zaoberá klasifikáciou typu vína (červené alebo biele) na základe jeho fyzikálno-chemických vlastností. Na riešenie tejto úlohy bol použitý model `RandomForestClassifier`, ktorý poskytuje robustné a presné výsledky pre binárne klasifikačné problémy. Na základe výsledkov sme sa rozhodli použiť selekciu dát.

3.1 RandomForestClassifier

3.1.1 Príprava dát

Datasety červeného a bieleho vína boli skombinované, pričom bol pridaný nový atribút `type`, ktorý označuje červené víno hodnotou 1 a biele víno hodnotou 0. Výsledný dataset bol následne rozdelený na trénovaciu (70%), validačnú (20%) a testovaciu (10%) množinu, pričom stratifikované delenie zabezpečilo rovnaký pomer tried vo všetkých množinách.

```
X_train, X_temp, y_train, y_temp = train_test_split(X, y,
    ↪ test_size=0.3, random_state=42, stratify=y)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp,
    ↪  test_size=1/3, random_state=42, stratify=y_temp)
```

Listing 3: Príprava dát

3.1.2 Tréning modelu

Na trénovanie bol použitý model `RandomForestClassifier` s predvolenými hyperparametrami. Tréning prebehol na trénovacej množine a model bol následne vyhodnotený na validačnej množine.

```
# Train a Random Forest Classifier
clf = RandomForestClassifier(random_state=42, n_estimators=100)
clf.fit(X_train, y_train)

# Evaluate the model on validation set
y_val_pred = clf.predict(X_val)
```

Listing 4: Trénovanie a vyhodnotenie modelu na validačnej množine.

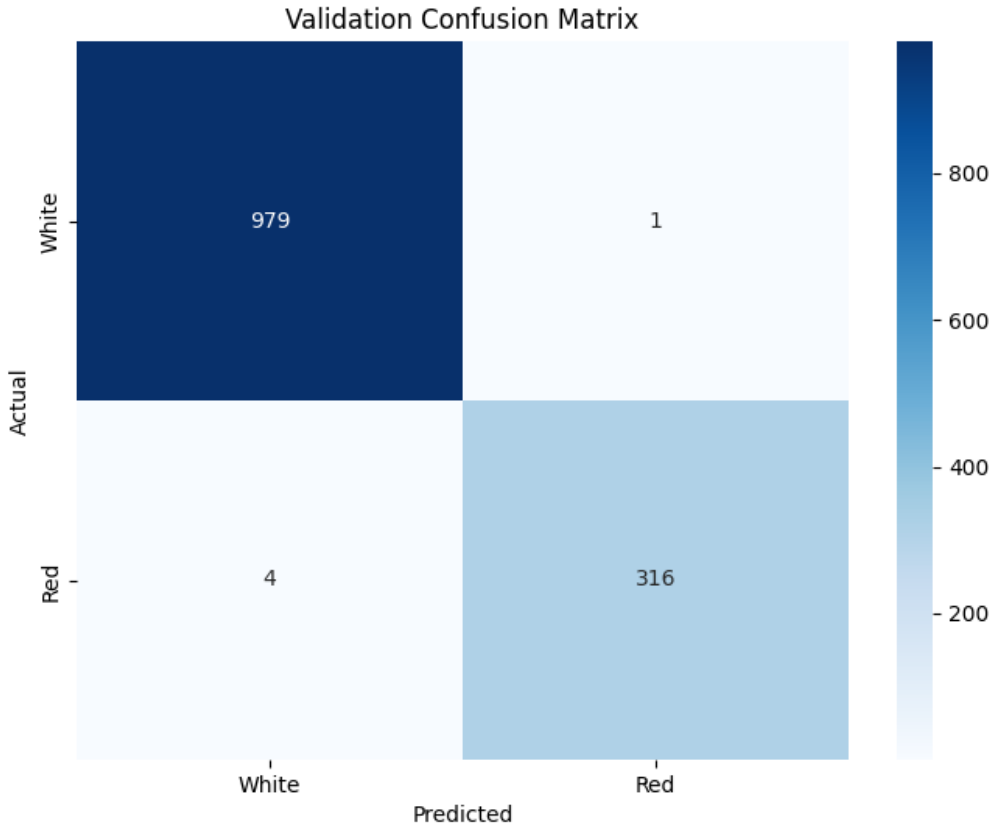
3.1.3 Výsledky a vyhodnotenie

Validačná množina

Model bol vyhodnotený na validačnej množine, pričom dosiahol nasledujúce výsledky:

- **Presnosť (Precision)**: Pre biele víno (trieda 0) bola presnosť 1.00 a pre červené víno (trieda 1) bola presnosť 1.00.
- **Návratnosť (Recall)**: Návratnosť pre biele víno bola 1.00 a pre červené víno 0.99.
- **F1-skóre**: F1-skóre pre biele víno bolo 1.00 a pre červené víno 0.99.
- **Celková presnosť**: Model dosiahol presnosť 1.00 na validačnej množine.

Matica zámien pre validačnú množinu je uvedená na obrázku 3.



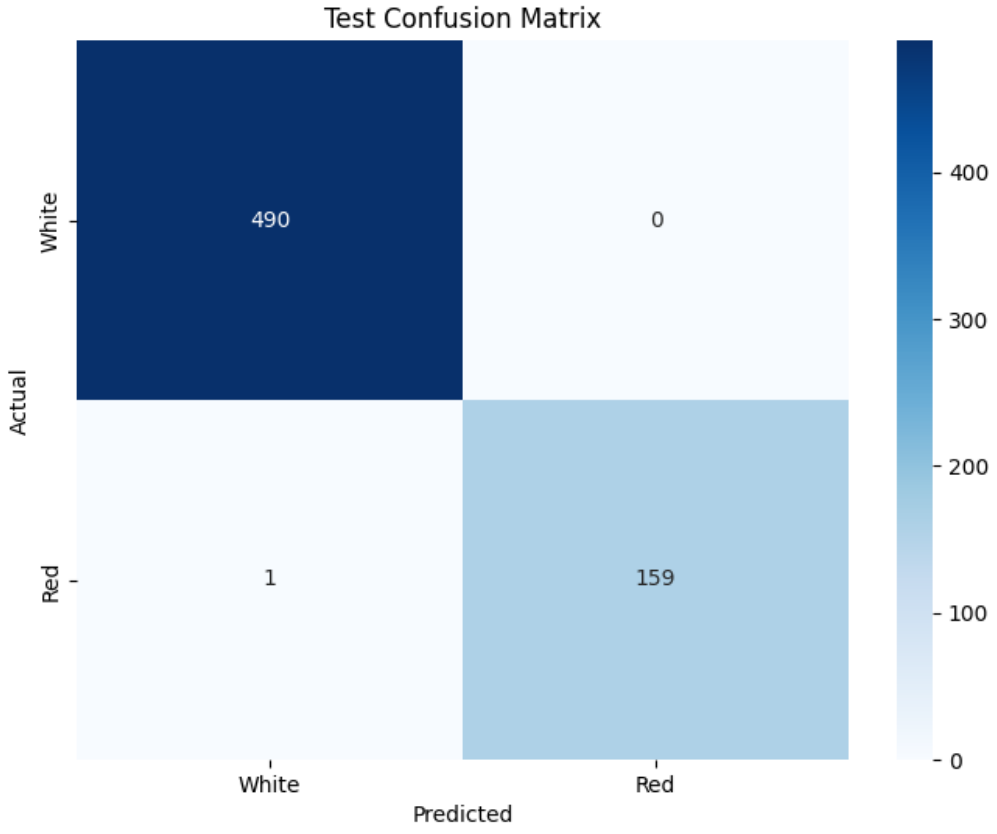
Obr. 3: Matica zámien pre validačnú množinu.

Testovacia množina

Na testovacej množine model preukázal nasledovné metriky:

- **Presnosť (Precision)**: Pre biele víno (trieda 0) bola presnosť 1.00 a pre červené víno (trieda 1) bola presnosť 1.00.
- **Návratnosť (Recall)**: Návratnosť pre biele víno bola 1.00 a pre červené víno 0.99.
- **F1-skóre**: F1-skóre pre biele víno bolo 1.00 a pre červené víno 1.00.
- **Celková presnosť**: Model dosiahol presnosť 1.00 na testovacej množine.

Matica zámien pre testovaciu množinu je uvedená na Obrázku 4.



Obr. 4: Matica zámien pre testovaciu množinu.

Dôležitosť vlastností

Najdôležitejšie vlastnosti pre klasifikáciu typu vína boli:

- **Celkový obsah oxidu siričitého (`total sulfur dioxide`)**: Tento atribút zohrával kľúčovú úlohu pri odlíšení bieleho vína od červeného, keďže jeho hladina je zvyčajne vyššia v bielych vínach.
- **Obsah chloridov (`chlorides`)**: Chloridy, ktoré sú indikátorom obsahu soli vo víne, výrazne prispeli k klasifikácii typu vína.
- **Prchavá kyslosť (`volatile acidity`)**: Táto vlastnosť, spojená s octovými tónmi vo víne, bola dôležitá najmä pri identifikácii červených vín.

Vizualizácia ukazuje, ktoré chemické vlastnosti najviac prispievajú k odlíšeniu červeného a bieleho vína. Pre ďalšie testovanie sme skúsili selektovať vstupné premenné a udržať iba tie najdôležitejšie.

3.1.4 Diskusia výsledkov

Model `RandomForestClassifier` dosiahol vynikajúce výsledky na oboch množinách, s takmer dokonalou presnosťou, návratnosťou a F1-skóre. Na validačnej množine boli zaznamenané 4 nesprávne klasifikácie červeného vína ako bieleho a 1 nesprávna klasifikácia na testovacej množine. Táto konzistentnosť naznačuje, že model nie je overfitovaný a je schopný generalizovať na nové dáta.

3.2 Gradient Boosting Classifier

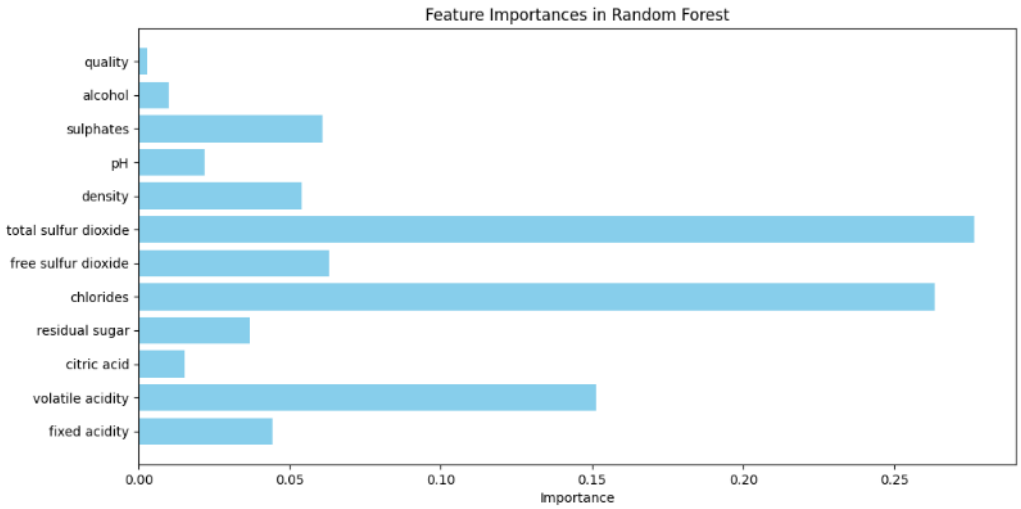
3.2.1 Príprava dát

Dáta boli načítané a predspracované rovnako ako pre RandomForetClassifier.

3.2.2 Výber vlastností a trénovanie

Na výber najdôležitejších fyzikálno-chemických vlastností sme použili model `GradientBoostingClassifier` s predvolenými hyperparametrami. Na základe tohto modelu boli vybrané vlastnosti. Tieto vlastnosti sa ukázali ako najvýznamnejšie pre odlíšenie červeného a bieleho vína.

```
# Feature Selection using Gradient Boosting
feature_selector = GradientBoostingClassifier(random_state=42,
    ↪ n_estimators=100)
```



Obr. 5: Dôležitosť vlastností pre klasifikáciu typu vína.

```
feature_selector.fit(X_train, y_train)

# Select important features
sfm = SelectFromModel(feature_selector, prefit=True)
X_train_selected = sfm.transform(X_train)
X_val_selected = sfm.transform(X_val)
X_test_selected = sfm.transform(X_test)

selected_features = X.columns[sfm.get_support()]
print("Selected Features:", selected_features)

# Train a Gradient Boosting Classifier
clf = GradientBoostingClassifier(random_state=42, n_estimators
    ↪ =100)
clf.fit(X_train_selected, y_train)
```

Listing 5: Selekcia a trénovanie použitím Gradient Boosting

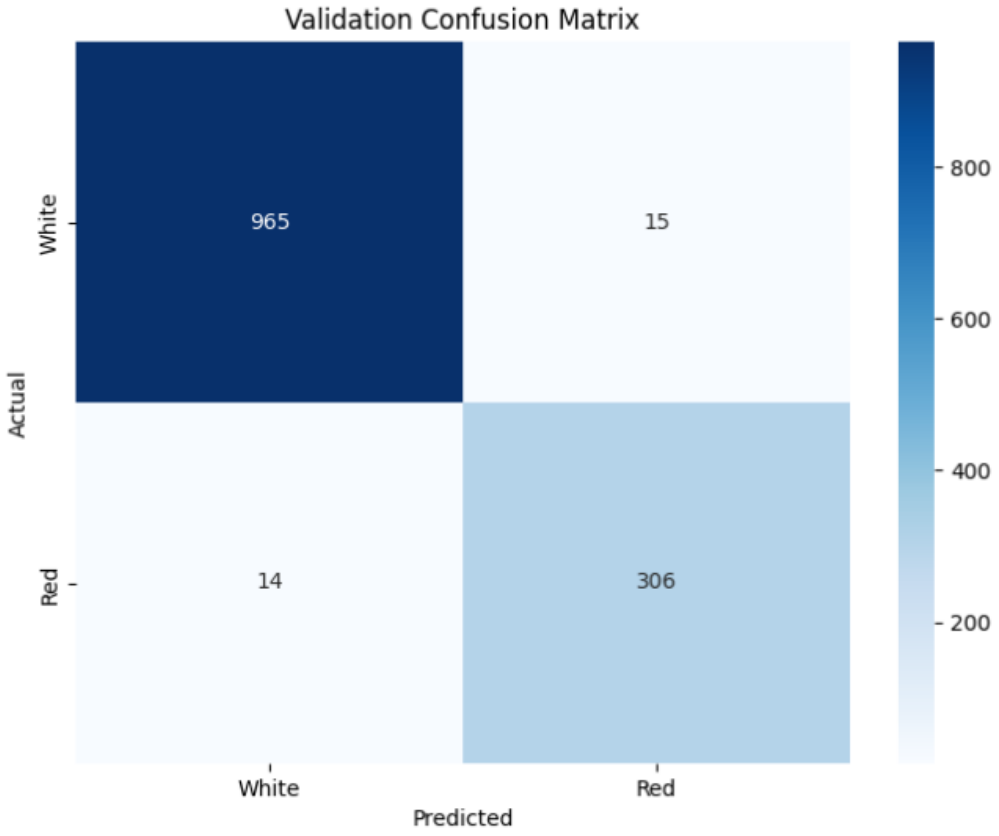
3.2.3 Výsledky a vyhodnotenie

Validačná množina

Model bol vyhodnotený na validačnej množine po aplikovaní výberu vlastností:

- **Presnosť (Precision)**: Pre biele víno (trieda 0) bola presnosť 0.99 a pre červené víno (trieda 1) bola presnosť 0.95.
- **Návratnosť (Recall)**: Návratnosť pre biele víno bola 0.98 a pre červené víno 0.96.
- **F1-skóre**: F1-skóre pre biele víno bolo 0.99 a pre červené víno 0.95.
- **Celková presnosť**: Model dosiahol presnosť 0.98 na validačnej množine.

Matica zámien pre validačnú množinu je zobrazená na obrázku 6.



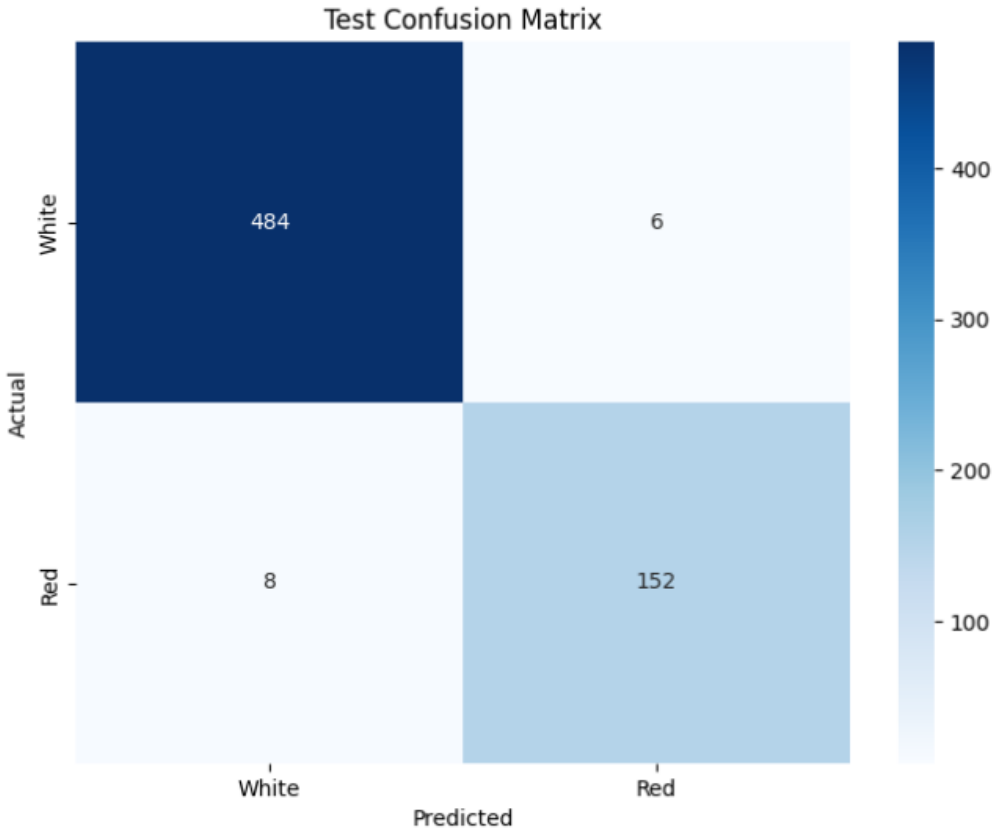
Obr. 6: Matica zámien pre validačnú množinu s Gradient Boosting Classifier.

Testovacia množina

Na testovacej množine model dosiahol nasledovné metriky:

- **Presnosť (Precision)**: Pre biele víno (trieda 0) bola presnosť 0.98 a pre červené víno (trieda 1) bola presnosť 0.96.
- **Návratnosť (Recall)**: Návratnosť pre biele víno bola 0.99 a pre červené víno 0.95.
- **F1-skóre**: F1-skóre pre biele víno bolo 0.99 a pre červené víno 0.96.
- **Celková presnosť**: Model dosiahol presnosť 0.98 na testovacej množine.

Matica zámien pre testovaciu množinu je zobrazená na obrázku 7.

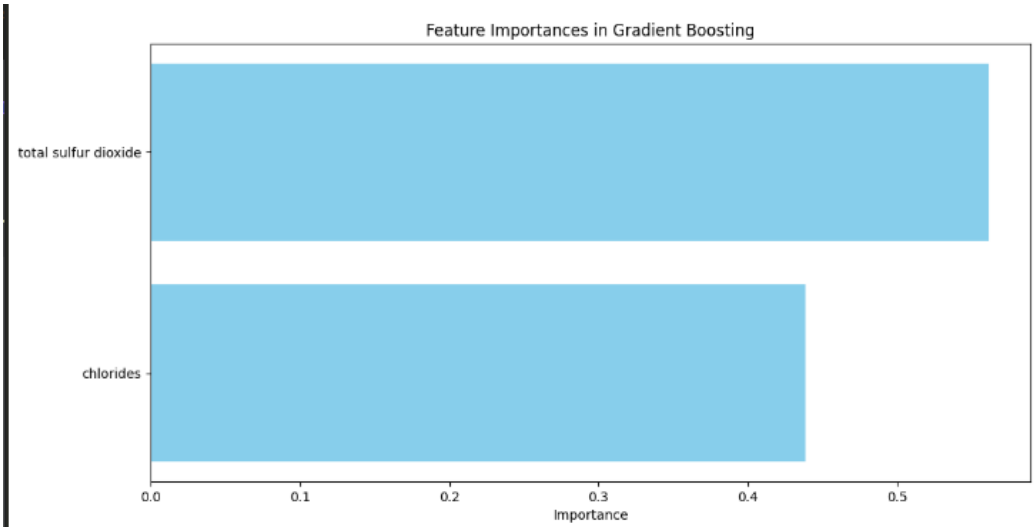


Obr. 7: Matica zámien pre testovaciu množinu s Gradient Boosting Classifier.

Dôležitosť vlastností

Po vybraní vlastností (celkový obsah síry, chloridy) pomocou modelu Gradient Boosting bola analyzovaná ich relatívna dôležitosť. Výsledky ukazujú,

že vlastnosť **celkový obsah síry** má najväčší vplyv na klasifikáciu, nasledovaná vlastnosťou **chloridy**. Vizualizácia dôležitosti vlastností je zobrazená na obrázku 8.



Obr. 8: Dôležitosť vybraných vlastností pre Gradient Boosting Classifier.

3.2.4 Diskusia výsledkov

Model `GradientBoostingClassifier` dosiahol presnosť 0.98 na validačnej aj testovacej množine. Napriek tomu, že boli použité iba dve vlastnosti, model dokázal dosiahnuť výsledky porovnateľné s predošlým modelom, čo poukazuje na silu výberu relevantných vlastností. Väčšina chýb pochádza z nesprávnej klasifikácie červeného vína ako bieleho, čo naznačuje, že pre červené víno by mohla byť užitočná ďalšia optimalizácia.

4 Klasifikácia vína na základe kvality

V tejto sekcii sa zaoberáme klasifikáciou vína na základe jeho kvality, kde cieľom je predpovedať kategóriu kvality vína na základe rôznych chemických vlastností. Tento problém klasifikácie bol riešený použitím viacerých modelov strojového učenia, *Gradient Boosting Classifier* a *XGBoost Classifier*. Klasifikácia bola vykonaná na základe datasetov červeného a bieleho vína, ktoré obsahujú rôzne chemické charakteristiky ako pH, kyseliny, cukry, obsah alkoholu a ďalšie. Kategórie kvality vína boli rozdelené do štyroch tried: *Veľmi zlé*, *Priemerné*, *Dobré* a *Vynikajúce*.

4.1 Gradient Boosting Classifier

4.1.1 Príprava dát

Na začiatku bola vykonaná príprava dát, kde sa dáta z oboch datasetov (červené a biele víno) spojili do jedného datasetu. Následne bola kvalita vína kategorizovaná na základe hodnoty atribútu *quality* do štyroch tried, ako je uvedené nižšie:

$$\text{quality\_category} = \begin{cases} \text{Veľmi zlé} & \text{pre kvalitu} \leq 3 \\ \text{Priemerné} & 4 \leq \text{quality} \leq 5 \\ \text{Dobré} & 6 \leq \text{quality} \leq 7 \\ \text{Vynikajúce} & \text{quality} \geq 8 \end{cases}$$

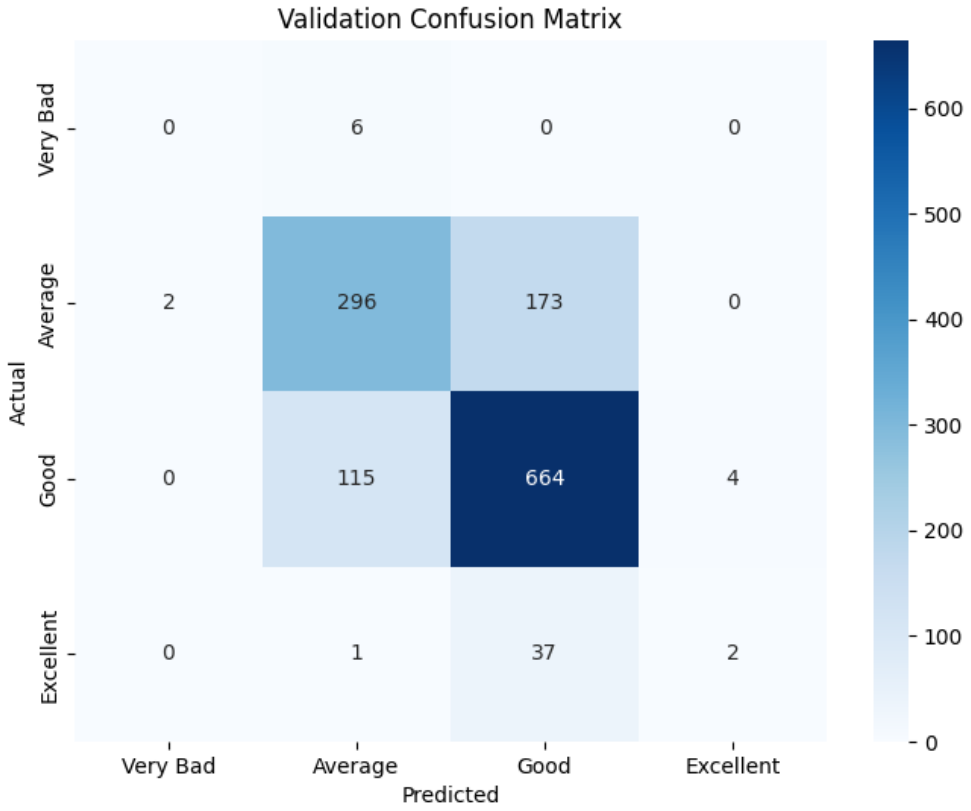
Dataset bol následne rozdelený na trénovaciu, validačnú a testovaciu množinu, pričom sme použili stratifikovaný rozdeľovač, aby sme zabezpečili rovnomerné rozdelenie jednotlivých tried v každej množine.

```
wine_data = pd.concat([red_wine, white_wine], axis=0).
    ↪ reset_index(drop=True)
wine_data['quality_category'] = pd.cut(wine_data['quality'],
    ↪ bins=[-1, 3, 5, 7, 10], labels=['Very␣Bad', 'Average', '
    ↪ Good', 'Excellent'])

X = wine_data.drop(columns=['quality', 'quality_category']) #
    ↪ Features
y = wine_data['quality_category'] # Target categories

# Split into training (70%), validation (20%), and testing
    ↪ (10%) sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y,
    ↪ test_size=0.3, random_state=42, stratify=y)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp,
    ↪  test_size=1/3, random_state=42, stratify=y_temp)
```

Listing 6: Príprava dát

4.1.2 Výber vlastností a tréning modelu

Po príprave dát sme sa rozhodli použiť všetky dostupné chemické vlastnosti vína ako vstupy do modelu. Kvalita vína bola predikovaná pomocou modelu *Gradient Boosting Classifier*, ktorý je vhodný na riešenie problémov s kla-



Obr. 9: Matica zámien na validačnej množine pre *Gradient Boosting Classifier*.

sifikáciou, najmä v prípadoch s nevyváženými dátami. Model bol trénovaný na trénovacej množine a vyhodnocovaný na validačnej a testovacej množine.

```
clf = GradientBoostingClassifier(random_state=42, n_estimators
    ↪ =100)
clf.fit(X_train, y_train)
```

Listing 7: Tréning modelu

4.1.3 Výsledky a vyhodnotenie

V tejto podsekcii uvádzame výsledky hodnotenia modelu *Gradient Boosting Classifier* na validačných a testovacích dátach. Po tréningu modelu na trénovacích dátach sme vykonali vyhodnotenie na validačnej a testovacej množine.

Validačná množina

Na validačnej množine sme dosiahli nasledujúce výsledky:

Kategória	**Presnosť**	**Recall**	**F1-skóre**
Priemerné	0.71	0.63	0.67
Vynikajúce	0.33	0.05	0.09
Dobré	0.76	0.85	0.80
Veľmi zlé	0.00	0.00	0.00

Tabuľka 1: Výsledky klasifikácie na validačnej množine pre model *Gradient Boosting Classifier*.

Na validačnej množine dosiahol model celkovú presnosť 74%. Naopak, kategórie *Veľmi zlé* a *Vynikajúce* boli predikované veľmi slabým spôsobom, čo vedie k nízkym hodnotám presnosti a recall pre tieto triedy. Maticu zámien je možné vidieť na obrázku 9.

Testovacej množina

Na testovacej množine model dosiahol nasledovné výsledky:

Kategória	**Presnosť**	**Recall**	**F1-skóre**
Priemerné	0.75	0.62	0.68
Vynikajúce	1.00	0.05	0.10
Dobré	0.76	0.88	0.82
Veľmi zlé	0.00	0.00	0.00

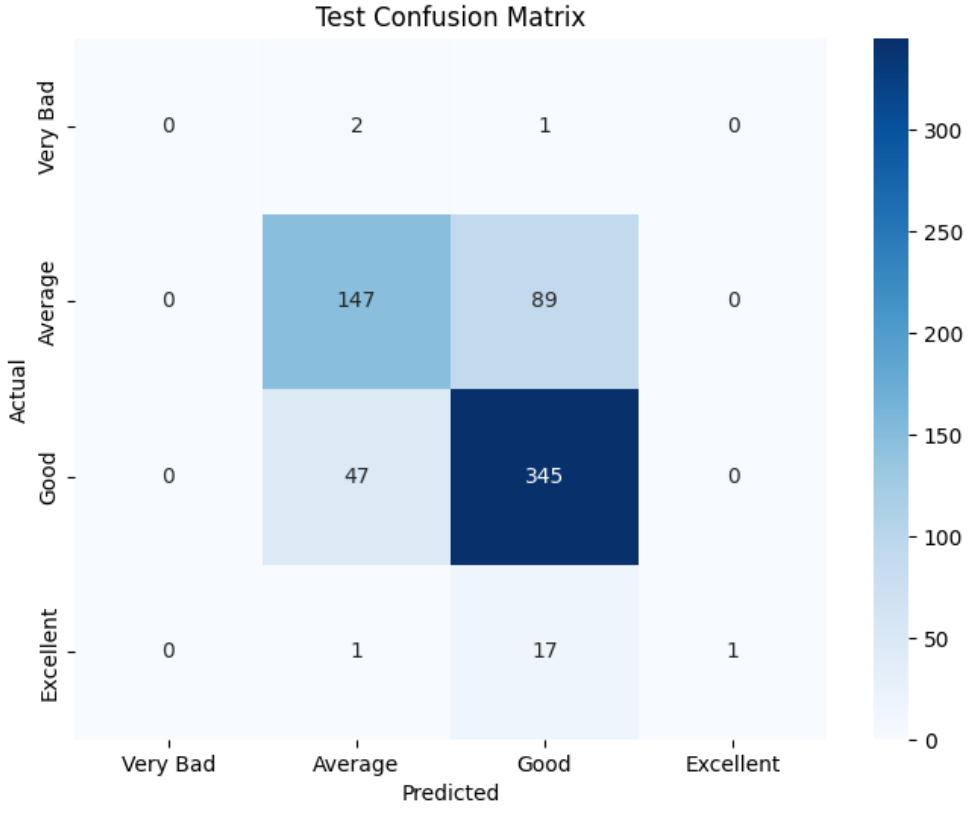
Tabuľka 2: Výsledky klasifikácie na testovacej množine pre model *Gradient Boosting Classifier*.

Testovacia presnosť dosiahla hodnotu 76%, pričom najlepšie výsledky boli opäť dosiahnuté pre kategóriu *Dobré*, kde recall bol 0.88. Naopak, kategória *Veľmi zlé* mala veľmi slabé výsledky s nulovým recall, čo naznačuje, že model nepredpovedal túto kategóriu. Maticu zámien je možné vidieť na obrázku 10.

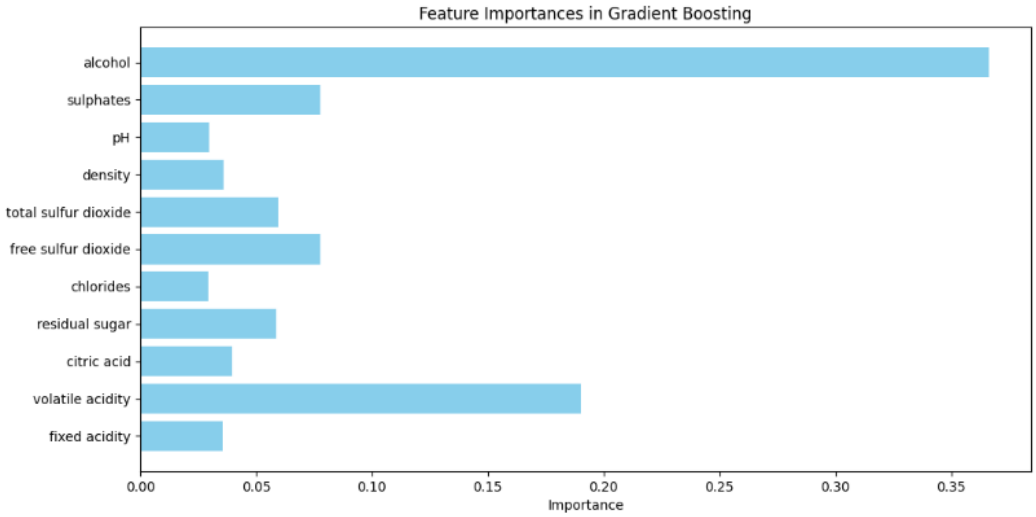
Chybovosť modelu

Chybovosť modelu na validačnej a testovacej množine bola nasledovná:

- Chybovosť na validačnej množine: 26.00%
- Chybovosť na testovacej množine: 24.15%



Obr. 10: Maticová zámien na testovacej množine pre *Gradient Boosting Classifier*.



Obr. 11: Dôležitosť vybraných vlastností pre Gradient Boosting Classifier.

Tento výsledok naznačuje, že model má pomerne vysokú chybovosť, najmä v prípadoch, keď ide o kategórie *Veľmi zlé* a *Vynikajúce*, ktoré sú podpredstavované v datasete.

Dôležitosť vlastností

Z obrázku 11 je vidieť, že hlavnou vlastnosťou pri trénovaní bol alkohol.

Okrem štandardného modelu, ktorý zahŕňal alkohol ako jednu z hlavných vstupných premenných, sme sa rozhodli vykonať experiment s vynechaním tejto premennej a pozorovali sme, ako to ovplyvní výkon modelu.

Výsledky na validačnej a testovacej množine sa ukázali byť veľmi podobné tým, ktoré sme dosiahli so všetkými premennými, vrátane alkoholu. Tento experiment ukázal, že vynechanie alkoholu z modelu nemalo významný vplyv na celkovú výkonnosť modelu. Model stále dosahoval najlepšie výsledky pri predikcii kategórie *Dobré*, ale výsledky pre kategórie *Vynikajúce* a *Veľmi zlé* ostali slabé. To naznačuje, že alkohol, aj keď je relevantnou premennou pre kvalitu vína, nebol rozhodujúcim faktorom pre zlepšenie výkonnosti modelu.

4.1.4 Diskusia výsledkov

Výsledky ukazujú, že model *Gradient Boosting Classifier* je silný nástroj na klasifikáciu vína na základe jeho kvality, no stále existujú oblasti na zlepšenie. Model mal najlepšie výsledky v predikcii kategórie *Dobré*, kde dosiahol vysoký recall a f1-skóre. Naopak, kategórie *Veľmi zlé* a *Vynikajúce* sa ukázali

byť veľmi problematické. Tento problém je spôsobený nevyvážením dát, kde kategórie s nižšou frekvenciou, ako *Veľmi zlé* a *Vynikajúce*, mali veľmi nízku reprezentáciu v trénovacích dátach.

Na zlepšenie modelu by sa mohli využiť techniky vyváženia dát, ako je SMOTE (Synthetic Minority Over-sampling Technique).Okrem vyváženia dát by sme mohli preskúmať aj iné modely strojového učenia, ako napríklad *XGBoost*, ktorý by mohol poskytnúť lepšie výsledky v prípade veľmi nevyvážených tried.

4.2 Klasifikácia vína na základe kvality pomocou XGBoost

V tejto sekcii sme implementovali klasifikáciu vína na základe jeho kvality pomocou modelu XGBoost. Na dosiahnutie čo najlepších výsledkov sme vykonali niekoľko kľúčových krokov, ako je spracovanie dát, vyváženie tried pomocou techniky SMOTE a optimalizácia hyperparametrov.

4.2.1 Príprava dát

Príprava datasetu je rovnaké ako pre Gradient Boosting Classifier.

4.2.2 Vyváženie dát pomocou SMOTE

Vzhľadom na nevyváženosť tried v pôvodnom datasete sme použili techniku *SMOTE* (Synthetic Minority Over-sampling Technique), ktorá generuje syntetické vzorky pre menšinové triedy, čím sa vyváži počet vzoriek medzi jednotlivými triedami. Tento krok bol vykonaný iba na trénovacej množine.

```
smote = SMOTE(random_state=42, sampling_strategy='auto',
    ↪ k_neighbors=5)
X_train_resampled, y_train_resampled = smote.fit_resample(
    ↪ X_train, y_train)

label_encoder = LabelEncoder()

# Fit and transform the training, validation, and test labels
y_train_encoded = label_encoder.fit_transform(y_train_resampled
    ↪ )
y_val_encoded = label_encoder.transform(y_val)
y_test_encoded = label_encoder.transform(y_test)
```

Listing 8: Vyváženie dát pomocou SMOTE

4.2.3 Optimalizácia hyperparametrov

Pre optimalizáciu modelu sme použili starý spósob cez for-cykli, keďže nám grid search nechcel fungovať, s rôznymi hodnotami hyperparametrov. Parametre, ktoré sme optimalizovali, zahŕňali:

- `n_estimators` (počet stromov v modeli)
- `learning_rate` (rýchlosť učenia)
- `max_depth` (maximálna hĺbka stromu)

Po optimalizácii sme získali najlepšie parametre: `n_estimators=200`, `learning_rate=0.1`, a `max_depth=9`.

```
for n_estimators in param_grid['n_estimators']:
    for learning_rate in param_grid['learning_rate']:
        for max_depth in param_grid['max_depth']:
            # Create model with current hyperparameters
            model = XGBClassifier(
                n_estimators=n_estimators,
                learning_rate=learning_rate,
                max_depth=max_depth,
                random_state=42,
            )
            # Fit model
            model.fit(X_train_resampled, y_train_encoded)

            # Evaluate model on validation set
            y_val_pred_encoded = model.predict(X_val)
            y_val_pred = label_encoder.inverse_transform(
                ↪ y_val_pred_encoded)

            score = accuracy_score(y_val, y_val_pred)

            # Update best score and parameters
            if score > best_score:
                best_score = score
                best_params = {
                    'n_estimators': n_estimators,
                    'learning_rate': learning_rate,
                    'max_depth': max_depth,
```

```
}
```

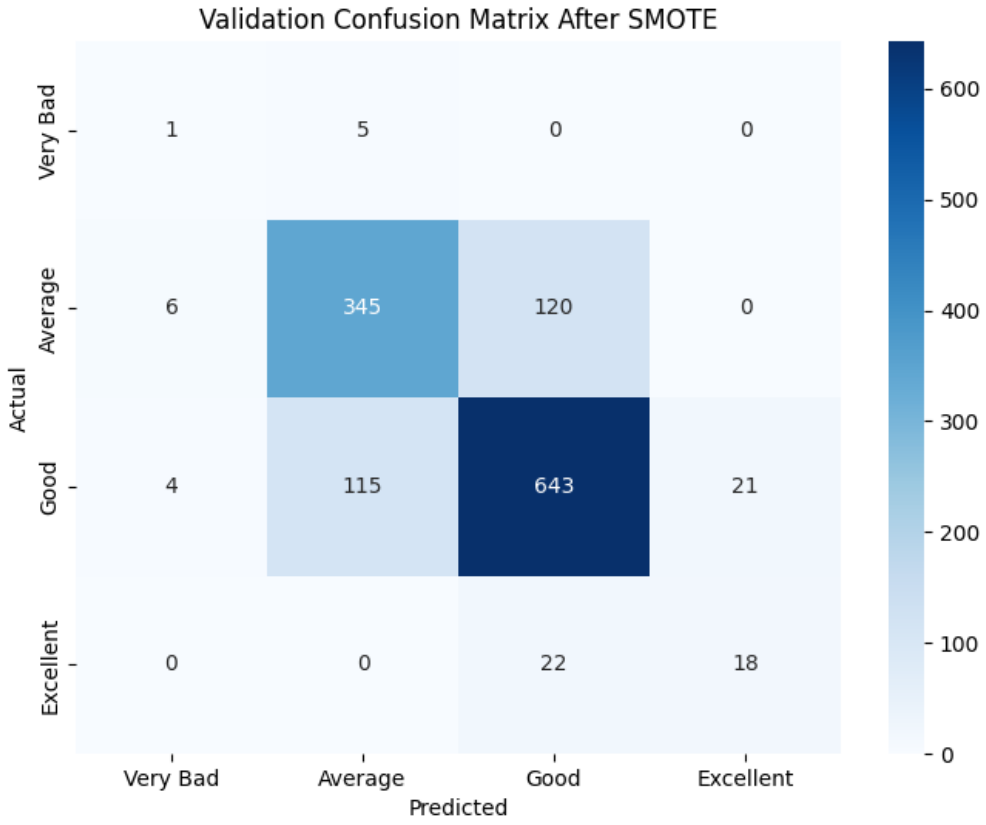
Listing 9: Optimalizácia hyperparametrov

4.2.4 Tréning modelu

Model sme natrénovali na najlepšom modeli, ktorý nám vyšiel z optimalizácie hyperparametrov.

```
model = XGBClassifier(random_state=42, n_estimators=200,
    ↪ learning_rate=0.1, max_depth=9)

model.fit(X_train_resampled, y_train_encoded)
```

Listing 10: Natrénovanie modelu na best model z optimalizácie.



Obr. 12: Matica zámien na validačnej množine po použití SMOTE

4.2.5 Výsledky a vyhodnotenie

Model bol trénovaný na vyváženej trénovacej množine a následne vyhodnotený na validačnej a testovacej množine. Výsledky hodnotenia modelu na validačnej a testovacej množine sú zobrazené v tabuľkách nižšie:

Validačná množina

Kategória	**Presnosť**	**Recall**	**F1-skóre**
Priemerné	0.74	0.73	0.74
Vynikajúce	0.46	0.45	0.46
Dobré	0.82	0.82	0.82
Veľmi zlé	0.09	0.17	0.12

Tabuľka 3: Klasifikačná správa na validačnej množine po použití SMOTE

Testovacia množina

Kategória	**Presnosť**	**Recall**	**F1-skóre**
Priemerné	0.75	0.72	0.74
Vynikajúce	0.46	0.32	0.38
Dobré	0.82	0.83	0.83
Veľmi zlé	0.00	0.00	0.00

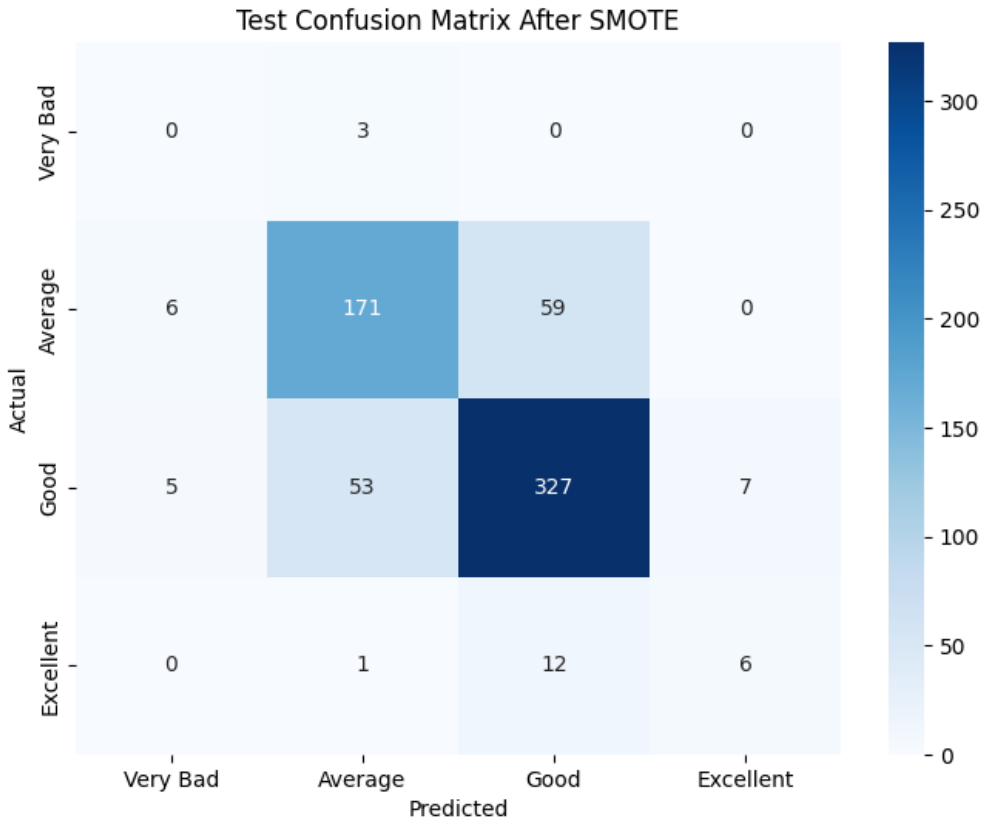
Tabuľka 4: Klasifikačná správa na testovacej množine po použití SMOTE

4.2.6 Vizualizácia výsledkov

Výsledky hodnotenia sme vizualizovali pomocou matice zámien a grafu dôležitosti vlastností. Na obrázkoch 12 a 13 sú zobrazené matice zámien pre validačnú a testovaciu množinu.

4.2.7 Diskusia výsledkov

Na základe dosiahnutých výsledkov môžeme konštatovať, že aplikácia SMOTE na vyváženie dát výrazne zlepšila výkon modelu, najmä pokiaľ ide o triedu *Good*. Avšak model mal stále problémy s predikciou triedy *Very Bad*, kde presnosť bola veľmi nízka. Najlepšie výsledky boli dosiahnuté s parametrami `n_estimators=200`, `learning_rate=0.1` a `max_depth=9`, pričom celková presnosť na testovacej množine bola 78%.



Obr. 13: Matica zámien na testovacej množine po použití SMOTE

5 Clustering a semi-supervised learning

5.1 Predspracovanie dát

Najprv sme načítali dva datasety, jeden pre červené víno a druhý pre biele víno, ktoré sme následne spojili do jedného datasetu. Pre spracovanie dát sme oddelili vstupné atribúty (chemické vlastnosti) a cieľovú premennú (kvalitu vína). Dataset sme rozdelili na trénovaciu a neoznačenú množinu v pomere 20% na trénovanie a 80% na testovanie.

5.2 Semi-supervised learning

Pre aplikáciu semi-supervised learning sme použili K-means klastrovanie na neoznačené dáta. K-means klastrovanie rozdelilo neoznačené dáta do štyroch klastrov. Tieto klastre sme následne použili ako pseudo-štítky pre trénovanie

modelu.

```
kmeans = KMeans(n_clusters=4, random_state=42)
kmeans_labels = kmeans.fit_predict(X_unlabeled_scaled)

# Assign pseudo-labels to the unlabeled data
X_combined = pd.concat([X_train, X_unlabeled])
y_combined = pd.concat([y_train, pd.Series(kmeans_labels)])
```

Listing 11: K-means klastrovanie a pridelenie pseudo-štítkov

5.3 Tréning a hodnotenie modelu

Model bol trénovaný pomocou náhodného lesa (Random Forest), ktorý bol aplikovaný na kombinovanú množinu označených a pseudo-označených dát.

Na hodnotenie výkonu modelu sme použili klasifikačnú správu (classification report), ktorá obsahuje metriky ako presnosť, zmysluplnosť (recall), f1-skóre a podporu (support). Tieto metriky sme vyhodnotili pred a po aplikovaní semi-supervised learning na trénovacích a testovacích dátach.

```
clf = RandomForestClassifier(random_state=42)
clf.fit(X_combined, y_combined)

# Evaluate the model on the labeled data
y_pred = clf.predict(X_train)
print("Classification␣Report␣before␣Semi-supervised␣Learning:")
print(classification_report(y_train, y_pred, zero_division=1))

# Evaluate the model on the test set, unlabeled
y_test_pred = clf.predict(X_unlabeled)
print("Classification␣Report␣after␣Semi-supervised␣Learning␣(
    ↪ using␣pseudo-labels):")
print(classification_report(y_unlabeled, y_test_pred,
    ↪ zero_division=1))
```

Listing 12: Tréning modelu a hodnotenie

5.4 Výsledky

5.4.1 Pred semi-supervised learning

Bez použitia semi-supervised learning sme získali nasledujúci klasifikačný report:

	precision	recall	f1-score	support
0	0.00	1.00	0.00	0
1	0.00	1.00	0.00	0
2	0.00	1.00	0.00	0
3	0.24	1.00	0.39	7
4	1.00	0.91	0.95	44
5	1.00	0.77	0.87	433
6	1.00	0.77	0.87	567
7	1.00	0.71	0.83	209
8	1.00	0.62	0.76	39
accuracy			0.76	1299
macro avg	0.58	0.86	0.52	1299
weighted avg	1.00	0.76	0.86	1299

5.4.2 Po semi-supervised learning

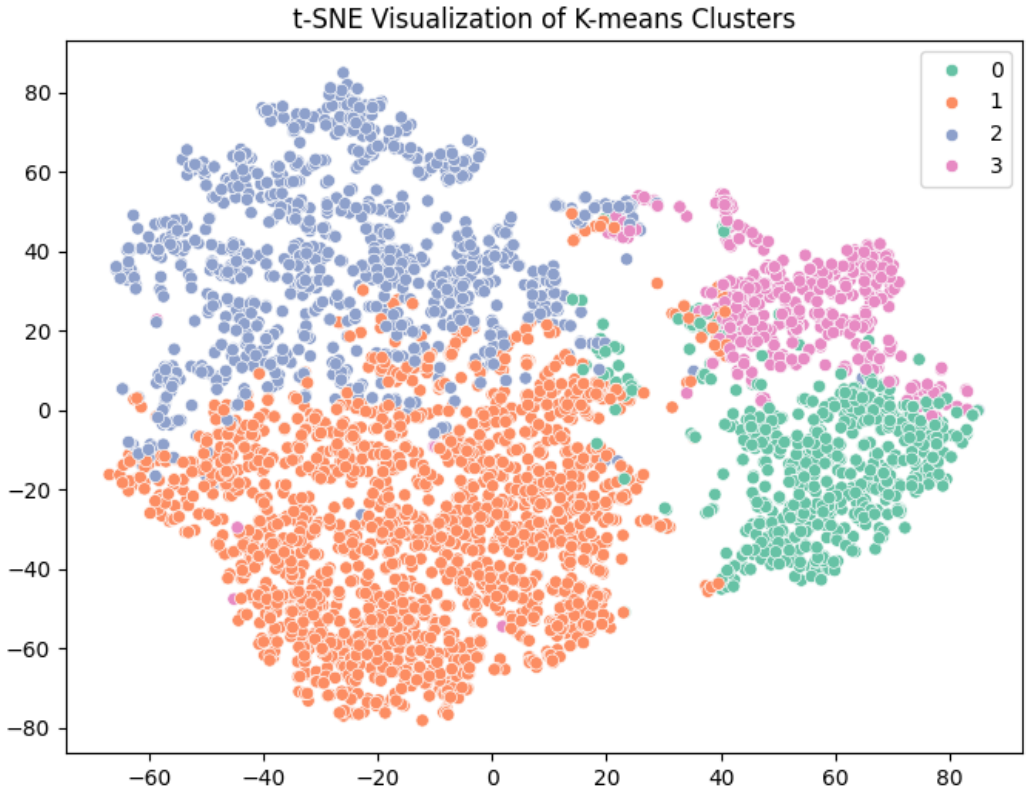
Po aplikovaní semi-supervised learning a použití pseudo-štítkov na neoznačené dáta sme získali nasledujúci klasifikačný report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	0
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	0
3	0.01	0.17	0.01	23
4	0.00	0.00	0.00	172
5	1.00	0.01	0.02	1705
6	1.00	0.01	0.02	2269
7	1.00	0.01	0.02	870
8	1.00	0.01	0.01	154
9	0.00	0.00	0.00	5
accuracy			0.01	5198
macro avg	0.40	0.02	0.01	5198
weighted avg	0.96	0.01	0.02	5198

Ako je vidieť z výsledkov, použitie pseudo-štítkov viedlo k výraznému zníženiu výkonu modelu na neoznačených dátach. Predpokladáme, že K-means klastrovanie nevytvorilo dostatočne kvalitné pseudo-štítky, čo spôsobilo zníženie presnosti modelu.

5.5 Vizualizácia

Pre lepšie pochopenie štruktúry dát a klastrovania sme použili metódu t-SNE na zníženie dimenzionality a vizualizáciu klastrov. Na obrázku 14 je zobrazená vizualizácia neoznačených dát a výsledkov K-means klastrovania.



Obr. 14: t-SNE vizualizácia K-means klastrov na neoznačených dátach.

5.6 Diskusia výsledkov

Použitie semi-supervised learning pomocou K-means klastrovania a pseudoštítkov na neoznačených dátach nepreukázalo očakávané zlepšenie výkonu modelu. Výsledky ukazujú, že K-means nebol schopný vytvoriť kvalitné pseudoštítky pre trénovanie klasifikátora. Tento prístup sa ukázal ako nevhodný pre tento konkrétny dataset.

6 Clustering a PCA

Cieľom tejto časti klasifikácia kvality vína pomocou dimenzionálnej redukcie a klastrovania. Pre túto úlohu sme použili techniky ako **PCA** (Principal Component Analysis) na redukciu dimenzií a **K-means klastrovanie** na rozdelenie dát do rôznych skupín.

6.1 Načítanie a predspracovanie dát

Na začiatku sme načítali dva dataset, jeden pre červené víno a druhý pre biele víno. Vzhľadom na to, že PCA je citlivé na škálu dát, normalizovali sme všetky numerické atribúty pomocou `StandardScaler`, ktorý zabezpečil, že všetky atribúty majú rovnakú váhu.

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Listing 13: Príprava dát

6.2 Redukcia dimenzií pomocou PCA

Na redukciu dimenzií sme použili metódu PCA s dvoma hlavnými komponentmi, čo nám umožnilo vizualizovať dáta v dvojrozmernom priestore.

```
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
```

Listing 14: PCA metóda

Výsledky PCA boli vizualizované pomocou scatter plotu, kde jednotlivé body boli zafarbené podľa hodnoty kvality vína (viď obrázok 15).

6.3 K-means klastrovanie

Po aplikovaní PCA sme na redukované dáta aplikovali K-means klastrovanie s počtom klastrov 4, čo nám umožnilo rozpoznať vzory a rozdeliť vína do rôznych skupín podľa podobnosti ich vlastností.

```
kmeans = KMeans(n_clusters=4, random_state=42)
kmeans_labels = kmeans.fit_predict(X_pca)

# Add K-Means labels as a new feature to the dataset
X_pca_with_clusters = np.hstack((X_pca, kmeans_labels.reshape
    ↪ (-1, 1)))
```

Listing 15: Kmeans a pridanie ako feature do dát

Obr. 15: Vizualizácia dát po PCA (kvalita vína na základe hlavných komponentov).

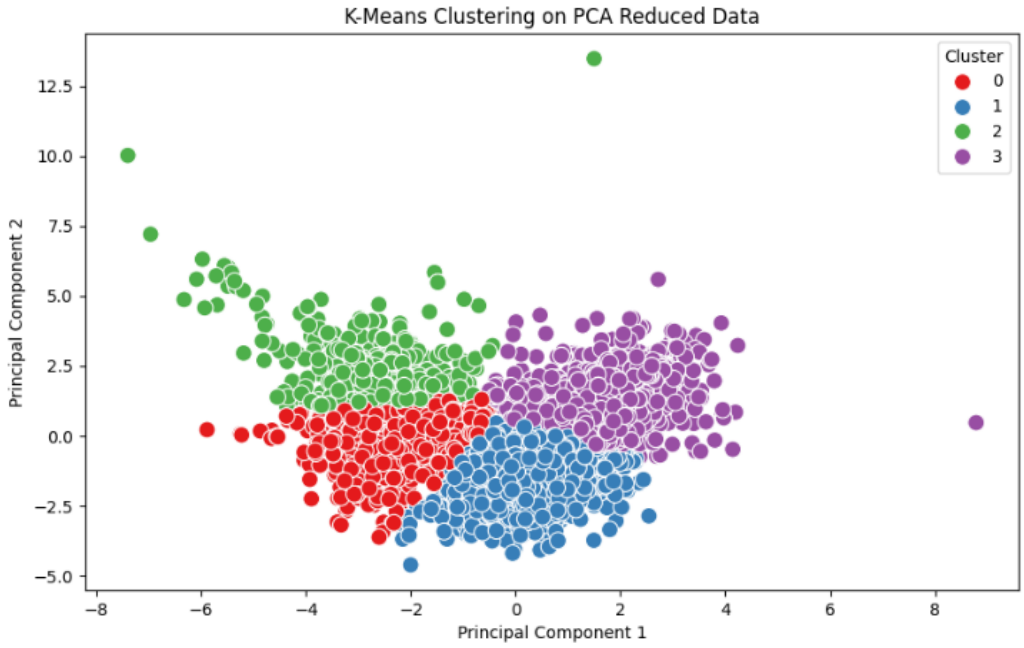
Vizualizácia klastrov pomocou scatter plotu ukazuje, ako K-means algoritmus rozdelil dáta do 4 rôznych klastrov (viď obrázok 16).

6.4 Tréning klasifikátora

Na klasifikáciu kvality vína sme použili Random Forest klasifikátor. Dáta sme rozdelili na trénovaciu a testovaciu množinu v pomere 80/20. Po tréningu modelu sme vyhodnotili jeho výkonnosť pomocou klasifikačnej správy.

```
X_train, X_test, y_train, y_test = train_test_split(
    ↪ X_pca_with_clusters, y, test_size=0.2, random_state=42)

# Train on features (including clusters)
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
```

Listing 16: Tréning klasifikátora



Obr. 16: K-means klastrovanie na dátach po PCA.

6.5 Výsledky

Po aplikovaní PCA a tréningu Random Forest klasifikátora sme získali nasledujúcu klasifikačnú správu:

```
Classification Report after PCA and K-Means:
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         6
           4       0.24      0.14      0.18        43
           5       0.55      0.61      0.58       402
           6       0.63      0.68      0.65       597
           7       0.62      0.51      0.56       215
           8       0.55      0.33      0.41        36
           9       0.00      0.00      0.00         1

    accuracy                           0.60      1300
   macro avg       0.37      0.32      0.34      1300
weighted avg       0.59      0.60      0.59      1300
```

Výsledky ukazujú, že model dosiahol celkovú presnosť 60%, pričom najlepšie

výsledky boli dosiahnuté pre triedy s vyššou frekvenciou, ako sú triedy 5 a 6.

6.6 Diskusia výsledkov

Aplikovaním PCA na redukciu dimenzií sme získali vizualizovateľné dáta, ktoré umožnili lepšie pochopenie vzorcov v dátach. K-means klastrovanie nám pomohlo identifikovať rôzne skupiny vína na základe ich vlastností. Klasifikácia s použitím Random Forest klasifikátora na týchto redukovaných dátach dosiahla uspokojivý výkon, avšak s nižšou presnosťou pri klasifikácii menej častých tried.

7 Záver

V tomto projekte sme sa zamerali na predikciu kvality vína pomocou rôznych metód strojového učenia a analýzy dát. Zamerali sme sa na viacero techník strojového učenia. V oblasti regresie sme použili optimalizáciu hyperparametrov. Klasifikačné metódy ako Random Forest, Gradient Boosting a XGBoost sme aplikovali na predikciu typu vína, pričom sme sa zaoberali optimalizáciou modelov, výberom vlastností a vyvážením dát pomocou techniky SMOTE. Pre každú z týchto metód sme vykonali podrobnú analýzu výsledkov, vrátane vizualizácie a diskusie.

V ďalšej časti sme sa venovali semi-supervised učeniu a clusteringu. Pomocou K-means klastrovania sme generovali pseudo-označenia pre neoznačené dáta a trénovali sme klasifikátor na kombinovaných dátach. Taktiež sme aplikovali PCA na zníženie dimenzií a následne vykonali K-means klastrovanie a klasifikáciu, aby sme zistili vplyv zníženia dimenzií na výkonnosť modelu.

Celkovo tento projekt poskytol hlboký pohľad na rôzne techniky strojového učenia, ako aj na ich aplikácie v oblasti analýzy vína, a naznačil potenciálne cesty na zlepšenie výkonnosti modelov pri práci s neoznačenými dátami.