Department of Computer Science
Technical University of Cluj-Napoca

Selection of important features and predicting wine quality using machine learning techniques

Machine Learning

Tools: Google Colab, ScikitLearn

Name:Popa Alexandra
Group: 30234
Email:popa_ada_98@yahoo.com

Contents

Chapter 1

Descriere proiect

Pentru acest proiect,am ales articolul "Selection of important features and predicting winequality using machine learning techniques" ce descrie utilizarea tehnicilor de machine learning pentru a prezice calitatea vinurilor prin doua modalitati diferite.[1] In prima faza se determina dependenta calitatii fata de variabilele independente ce descriu caracteristici ale vinurilor, iar mai apoi se prezice valoarea variabilei tinta.

Proiectul incerca sa implementeze ideile regasite in articol, prin aplicarea algoritmilor pe datele privind vinurile albe si analizarea predictiilor obtinute. Aceste rezultate au fost comparate cu cele obtinute in urma expeimentelor mai ample descrise in articol. Totodata, aplicarea algoritmilor incerca sa demonstreze importanta anumitor caracteristici ale vinurilor in prezicerea calitatii, in comparatie cu altele.

Chapter 2

Descriere dataset

Datele folosite pentru realiarea proiectului au fost luate de pe site-ul archive.ics.uci.edu.[2] Toate experimentele au fost efectuate pe dataset-ul winequality-white.csv ce contine 4898 se esantioane. Acesta include toate caracteristicile fizico-chimice necesare pentru a descrie un vin, informatii independente necesare pentru a face predicții și a trage concluzii. Acest set de date publice a fost utilizat de asemenea si pentru realizara experimentelor descrise in articolul ales.

Dataset-ul utilizat prezinta urmaroarele atribute:

→ fixed acidity

→ volatile acidity

→ citric acid

→ residual sugar

→ chlorides

→ free sulfur dioxide

→ total sulfur dioxid

→ density

→ pH

→ sulphates

→ alcohol

→ quality

Atributul "quality" se bazează pe un test senzorial realizat de cel putin trei somelieri si scalat in 11 clase de calitate de la 0 - foarte rau la 10 - excelent.

Chapter 3

Detalii de implementare

3.1 Reading dataset

Pentru inceput am extras tot setul de date citind fisierul de tip .csv, urmand ca apoi sa prelucrez si utilizez datele de intrare initiale. Datele initiale cuprind: 4898 rows x 12 columns.

```
data_df = pd.read_csv("winequality-white.csv")
data_df
```

3.2 Cleaning dataset

Dataset-ul utiliat vine cu specificarea faptului ca nu exista valori ale atributelor lipsa, astfel incat acesta nu necesita a fi curatat.Acest lucru poate fi observat din tabelul ce descrie dataset-ul, deoarece toate coloanele contin numarul maxim de valori, fara a exista valori nule.

```
data_df.describe(include="all")
```

3.3 Split dataset

In urma experimentelor efectuate, in articol se prezinta anumite rezultate conform carora anumite atribute din dataset-ul original influenteaza intr-o masura mai mica prezicerea calitatii fata de alte atribute. Astfel, am preluat aceste informatii si am creat un nou dataset cu atributele selectate ca fiind cele mai relevante in procesul de predictie. Astfel am aplicat aceiasi algoritmi pe ambele dataset-uri pentru a observ diferentele si a face o comparatie intre rezultatele descrise de articol si cele obtinute in proiectul de fata.

Atat setul de date complet cat si cel cu date selectate, au fost divizata in: set de date de test (test set) si set de date de antrenament(training set). Astfel, deoarece doresc ca programul sa imi prezica calitatea vinurilor,am extras din ambele seturi de date atribututul "quality" si am impartit astfel fiecare set de date in cate 2 seturi de date pe care le-am divizat in 2 categorii(X_train, X_test,y_train, y_test-pentru datele complete; X_sel_train, X_sel_test,y_sel_train, y_sel_test-pentru datele selectate).

```
data_sel=data_df.copy()
y=data_df["quality"]
```

```
y_sel=data_sel["quality"]

data_df.drop(columns=["quality"],inplace=True)
data_sel.drop(columns=["quality"],inplace=True)

#noul dataset cu atributele mai semnificative selectate
#atributele mai putin semnificative  sunt eliminate din datasetul compet
data_sel.drop(columns=["citric acid"],inplace=True)
data_sel.drop(columns=["chlorides"],inplace=True)
data_sel.drop(columns=["total sulfur dioxide"],inplace=True)

X=data_df
X_sel=data_sel

X_train,X_test,y_train,y_test=train_test_split(X,y, test_size=0.20,
random_state=42)
X_sel_train,X_sel_test,y_sel_train,y_sel_test=train_test_split(X_sel,y_sel,
test_size=0.20, random_state=42)

X_train
X_sel_train

y_train
y_sel_train
```

3.4 Machine Learning Algorithm

Pentru aplicarea algoritmilor de Machine Learning este necesar ca ambele seturi de date sa se treaca printr-un pipeline de transformare a coloanelor, cu ajutorul clasei "Pipeline" ce are ca scop standardizarea seturilor ce contin ambele tipuri de date.[3][4] Prin standardizare se intelege maparea valorilor intr-o anumita distributie pentru a putea fi prelucrate ulterior de algoritm.

```
all_features=["fixed acidity","volatile acidity","citric acid","residual
sugar","chlorides","free sulfur dioxide","total sulfur
dioxide","density","pH","sulphates","alcohol"]
sel_features=["fixed acidity","volatile acidity","residual sugar","free
sulfur dioxide","density","pH","sulphates","alcohol"]

preprocessor = ColumnTransformer([
    ("num", Pipeline([("scaler", StandardScaler())]), all_features)],
    remainder="drop")
preprocessor_sel = ColumnTransformer([
    ("num", Pipeline([("scaler", StandardScaler())]), sel_features)],
    remainder="drop")

preprocessor.fit(X_train)
X_train=preprocessor.transform(X_train)
X_test=preprocessor.transform(X_test)

preprocessor_sel.fit(X_sel_train)
X_sel_train=preprocessor_sel.transform(X_sel_train)
X_sel_test=preprocessor_sel.transform(X_sel_test)
```

Urmarind articolul studiat, au fost aplicati algoritmi de machine learning: support vector machine si neural network pe ambele seturi de date.

Algoritmul Linear Regression a fost aplicat doar pe setul de date complet pentru a analiza importanta tuturor atributelor ca intreg, in predictia finala.

3.5 Linear Regression

Acesta algoritm este unul simplu si a fost utilizat fara a se folosi vre-un parametru. Pentru determinarea scorului in urma aplicarii acestui algoritm, am folosit cross_val_score, ce returneaza matricea scorurilor estimatorului pentru fiecare rundă de validare încrucișată.[5] Pentru determinarea acestei matrici s-a folosit mertrica r2_score.

```
linreg=LinearRegression()
linreg.fit(X_train,y_train)
scores=cross_val_score(linreg,X_train,y_train,scoring="r2", cv=5)
np.mean(scores)
y_pred=linreg.predict(X_test)
r2_score(y_test,y_pred)
```

3.6 Multi-Layer Perceptron

Pentru aplicarea acestui algoritm, am utilizat clasa "MLPRegressor" cu parametrul "hidden_layer_sizes" ce reprezinta numarul de layere ascunse si numarul de neuroni continuti de fiecare layer ascuns in parte.In cazul de fata, exista un singur layer ascuns cu 5 neuroni. [6]

```
Rezultate obtinute pentru setul de date compet
mlp = MLPRegressor(random_state=1, max_iter=1000,hidden_layer_sizes=(5))
mlp.fit(X_train, y_train)
scores=cross_val_score(mlp,X_train,y_train,scoring="r2", cv=5)
np.mean(scores)
y_pred=mlp.predict(X_test)
r2_score(y_test, y_pred)
Rezultate obtinute pentru setul cu date selectate
mlp_sel = MLPRegressor(random_state=1, max_iter=1000,hidden_layer_sizes=(5))
mlp_sel.fit(X_sel_train, y_sel_train)
scores=cross_val_score(mlp_sel,X_sel_train,y_sel_train,scoring="r2", cv=5)
np.mean(scores)
y_pred=mlp_sel.predict(X_sel_test)
r2_score(y_sel_test, y_pred)
```

3.7 Support Vector Machine

Acest algoritm a fost aplicat cu urmatorii parametrii SVR(kernel='linear',C=100), specificand Kernel.[7] De asemenea, si pentru acest algoritm se utilizaeza metrica r2_score.

```
Rezultate obtinute pentru setul de date compet
regr = svm.SVR(kernel='linear',C=100)
regr.fit(X_train,y_train)
scores=cross_val_score(regr,X_train,y_train,scoring="r2", cv=5)
np.mean(scores)
y_pred=regr.predict(X_test)
r2_score(y_test, y_pred)
```

Rezultate obtinute pentru setul cu date selectate

```
regr_sel = svm.SVR(kernel='linear',C=100)
regr_sel.fit(X_sel_train,y_sel_train)
scores=cross_val_score(regr_sel,X_sel_train,y_sel_train,scoring="r2", cv=5)
np.mean(scores)
```

```
y_pred=regr_sel.predict(X_sel_test)
r2_score(y_sel_test, y_pred)
```

Chapter 4

Experimente

4.1 Metrici utilizate

Pentru calcularea scorurilor obtinute in urma aplicarii algoritmilor, am ales metrica r2_score. R2_score este o metrica pentru care : cel mai bun scor posibil este 1.0 și poate fi negativ (deoarece modelul poate fi arbitrar mai slab). Un model constant care prezice întotdeauna valoarea așteptată de y, fără a ține cont de caracteristicile de intrare, ar obține un scor R^2 de 0,0. [8] Totodata, predictia s-a facut utilizand aceiasi metrica: r2_score.

De asemenea pentru calcularea scorului, am utilizat clasa "cross_val_score", care imparte datele de training in flod-uri si apoi masoara performanta raportata prin k-fold cross-validation , performanta care este defapt media valorilor obtinute.[9] [10]

4.2 Rezultate

Linear Regression

In urma aplicarii algoritmului s-a obtinut un scor de 0.27712563278727187, ceea ce inseamna ca modelul antrenat este unul constant, mai mare ca 0 ce tinde spre 1(cel mai bun scor posibil). $R^2 < 50\%$, deci unul sau mai multe atribute nu sunt utile pentru a prezice targetul final. In urma predictiei facute s-a obtinut un scor de 0.2652750042179145.

Multi-Layer Perceptron

In urma aplicarii algoritmului cu parametrii random_state=1,max_iter=1000 si hidden_layer_sizes=(5) s-a obtinut un scor de 0.33506904518324804 pentru setul de date complet si 0.30928512702907845 pentru setul cu date selectate.Astfel se observa faptul ca diferenta dintre scoruri este una mica, de unde rezulta faptul ca datele care au fost eliminatedin al doilea set de date nu au o importanta mare in predictiile facute. Astfel, se poate observa ca selectarea datelor prezentata in articol a fost corecta stabilita. In urma obtinerii acestui rezultat, am facut o predictie pentru ambele modele si am obtinut un scor de 0.3548502391066687 pentru date complete si 0.334476931839381 pentru datele selectate.

Support Vector Machine

In urma aplicarii algoritmului folosind parametrul kernel='linear' s-a obtinut un scor de 0.27381149850385245 pentru setul de date complet si 0.27463978509861914,ceea ce inseamna

ca scorurile obtinute sunt constante, mai mari ca 0 ce tind spre 1. Astfel, datele eliminate din primul set nu influenteaza scorul final, ceea ce inseamna ca nu sunt relevante in predictia finala. In urma obtinerii acestui rezultat, am facut o predictii pentru ambele modele si s-a obtinut un scor de 0.25955747891827285, si un scor de 0.26087538583034087 si mai apropiate de valoarea 1.

Chapter 5

Concluzii

In concluzie, proiectul propus a avut ca scop analizarea rezultatelor obtinute in urma aplicarii mai multor algoritmi asupra setului de date. Asemenea ideilor descrise in articolul studiat, doar anumite atribute din setul de date initial contribuie la obtinerea unei predictii cu acuratete mai buna. Astfel, prin aplicarea celor doi algoritmi se demonstreaza faptul ca anumite caracteristici, care s-au dovedit irelevante, nu afecteaza predictia finala.

Chapter 6

Ghid de rulare a proiectului

Dupa deschiderea proiectului de pe link-ul de Google Colab este necesar sa se faca upload la dataset-ul aflat in arhiva: winequality-white.csv.Ulterior se va rula fiecare celula de cod in parte utilizand SHIFT+ENTER si se pot observa rezultatele obtinute in sectiunea de sub celula rulata.

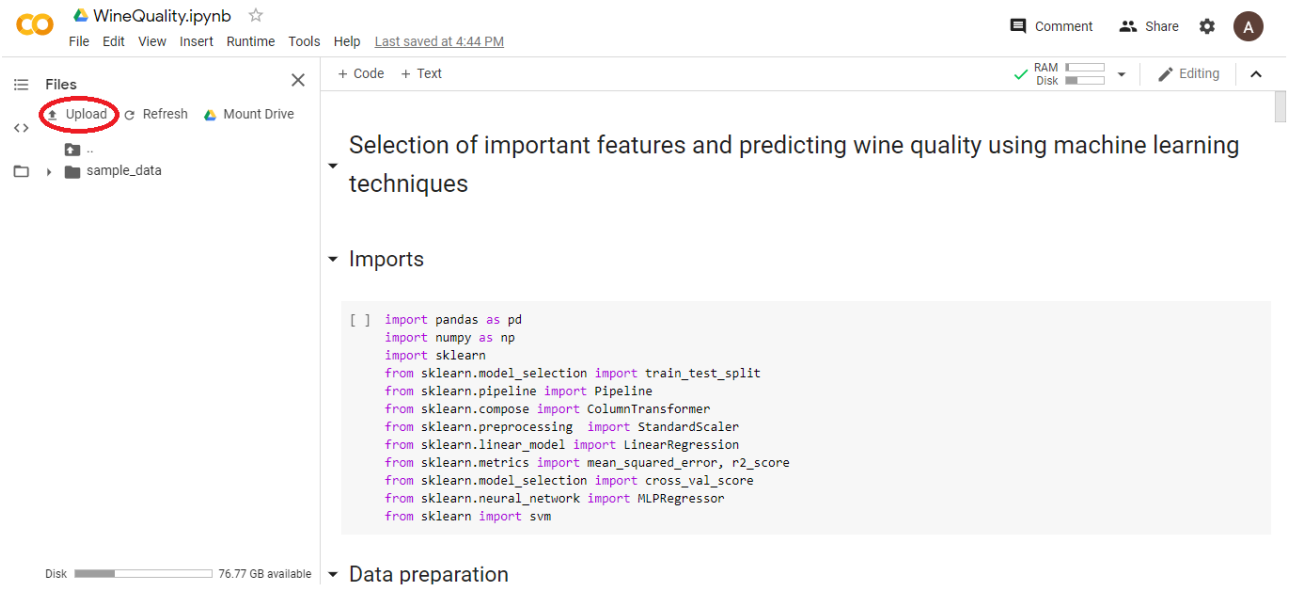

Figure 6.1: Upload dataset

Bibliography

[1] URL: https://www.sciencedirect.com/science/article/pii/S1877050917328053.

[2] URL: https://archive.ics.uci.edu/ml/datasets/wine+quality?fbclid=IwAR30z7qVICHGzVvgR8cu3n5uuqnpIMaBSVTh0ooGMNuJ-_9qrIbhVBoK2Do.

[3] URL: https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html?highlight=pipeline#sklearn.pipeline.Pipeline.

[4] URL: https://scikit-learn.org/stable/modules/generated/sklearn.compose.ColumnTransformer.html?highlight=column%20transformer#sklearn.compose.ColumnTransformer.

[5] URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html?highlight=linear%20regression#sklearn.linear_model.LinearRegression.

[6] URL: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html?highlight=mlpregresso#sklearn.neural_network.MLPRegressor.

[7] URL: https://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html#support-vector-regression-svr-using-linear-and-non-linear-kernels.

[8] URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html?highlight=r2_score#sklearn.metrics.r2_score.

[9] URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html?highlight=cros%20val%20score#sklearn.model_selection.cross_val_score.

[10] URL: https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation.

Intelligent Systems Group