Wine Quality prediction using IBM watson studio

1 INTRODUCTION

OVERVIEW

Wine is the most commonly used beverage globally, and its values are considered important in society. Wine is an alcoholic drink that is made up of fermented grapes.Quality of wine is important for its consumers, mainly for producers in the present competitive market to raise the revenue. Wine quality refers to the factors that go into producing a wine, as well as the indicators or characteristics that tell you if the wine is of high quality. Historically, wine quality used to be determined by testing at the end of the production.

PROPOSED SYSTEM

Using this project we can predict the quality of wine if it is good or bad. In this project, we present a wine quality prediction technique that utilises historical data to train simple machine learning models which are more accurate and can help us know the quality of wine. The models can be run on much less resource intensive environments. From this the best model is selected and saved in pkl format. We will be doing flask integration and IBM deployment.

2 LITERATURE SURVEY

2.1 EXISTING PROBLEM

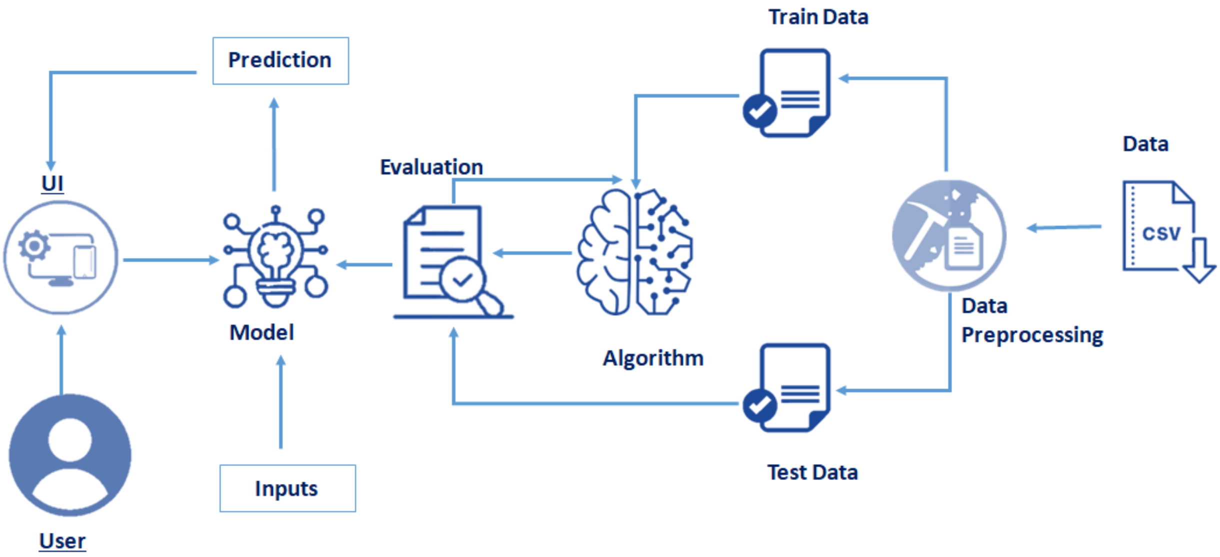
According to experts, wine is differentiated according to its smell, flavour, and colour, but we are not wine experts to say that wine is good or bad. Every person has their own opinion about the tastes, so identifying a quality based on a person’s taste is challenging. Judging the quality of wine manually is a really tough task, even the professional wine tasters have the accuracy of 71%.

2.2 PROPOSED SYSTEM

we present a wine quality prediction technique that utilises historical data to train simple machine learning models which are more accurate and can help us know the quality of wine.In this project, we are building a system that analyses the features of wine like residual sugar,pH,density,alcohol etc which determines the quality of wine. The goal of this project is to predict the quality of wine.

3 THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE/SOFTWARE DESIGNING

- Software requirements:

Anaconda navigator
Python packages
IBM watson studio

4 EXPERIMENTAL INVESTIGATION

Here we are going to build a machine learning model that predicts the quality of wine based on these characteristics.

- residual sugar

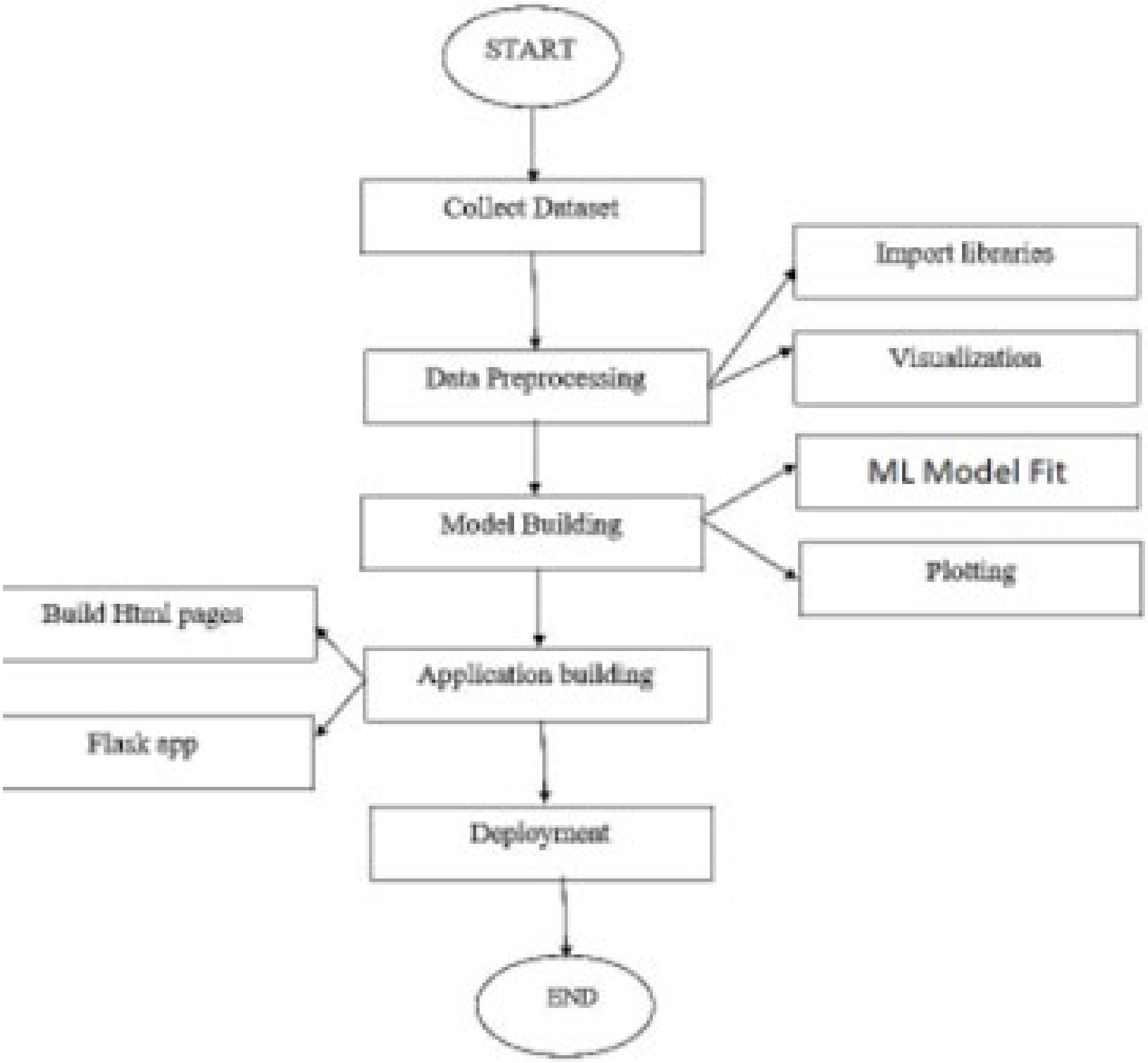
- pH
- Density
- alcohol
- Volatile acidity
- Citric acid
- Chlorides
- sulphates

  The value of these factors of wine are responsible for the quality prediction of wine.

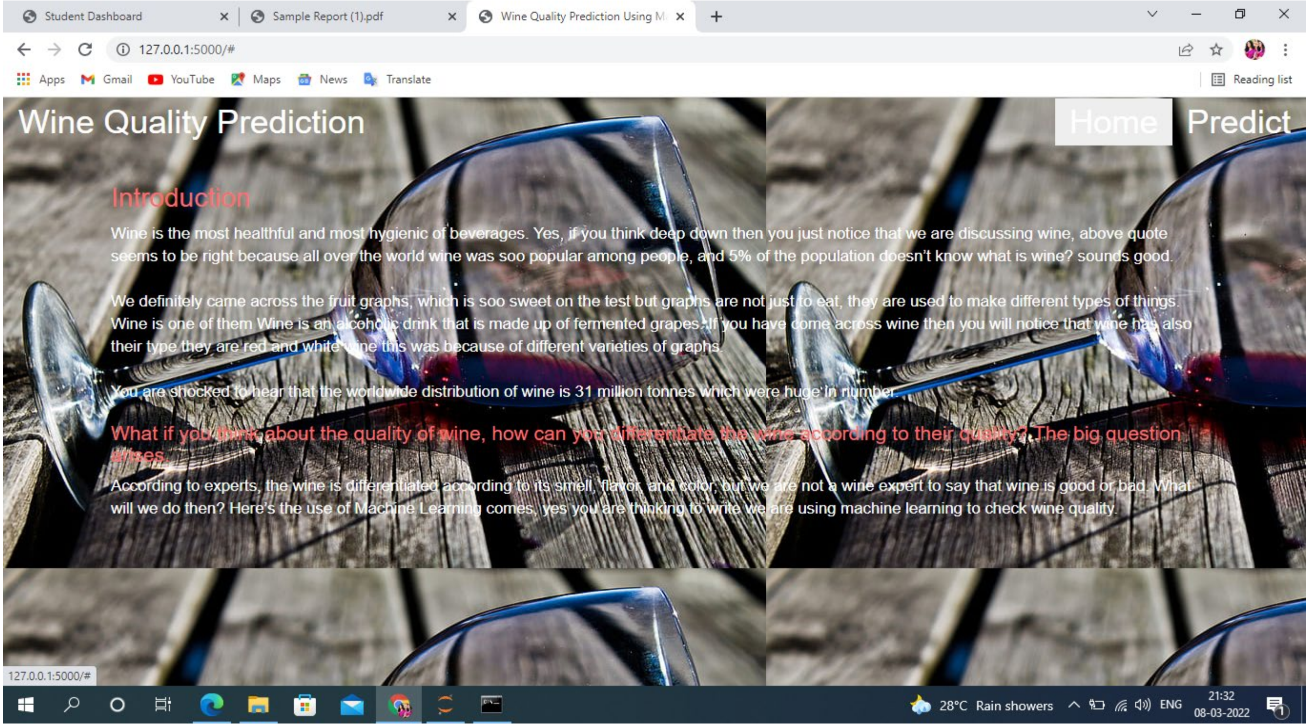
5 FLOWCHART

In UI by entering the details of wine like its colour the values of residual sugar,citric acid,sulphur,volatile acid.etc.Then UI shows whether the quality of wine is good or bad.
To accomplish this, we have to complete all the activities and tasks listed below

1. Data Collection.
2. Data Preprocessing.
3. Import the Libraries.
4. Importing the dataset.
5. Analyse the data
6. Taking care of Missing Data
7. Feature Scaling
8. Data Visualisation
9. Splitting Data into Train and Test.
10. Creating a dataset with a sliding window.
11. Model Building
12. Import the model building Libraries
13. Initialising the model
14. Training the model
15. Model Evaluation
16. Save the Model
17. Test the Model
18. Application Building
19. Create an HTML file
20. Build Python Code


START
Collect Dataset
Import libraries
Data Preprocessing
Visualization
ML Model Fit
Model Building
Plotting
Build Html pages
Application building
Flask app
Deployment
END

6 RESULT



Wine Quality Prediction

Introduction

Wine is the most healthful and most hygienic of beverages. Yes, if you think deep down then you just notice that we are discussing wine, above quote seems to be right because all over the world wine was soo popular among people, and 5% of the population doesn't know what is wine? sounds good.

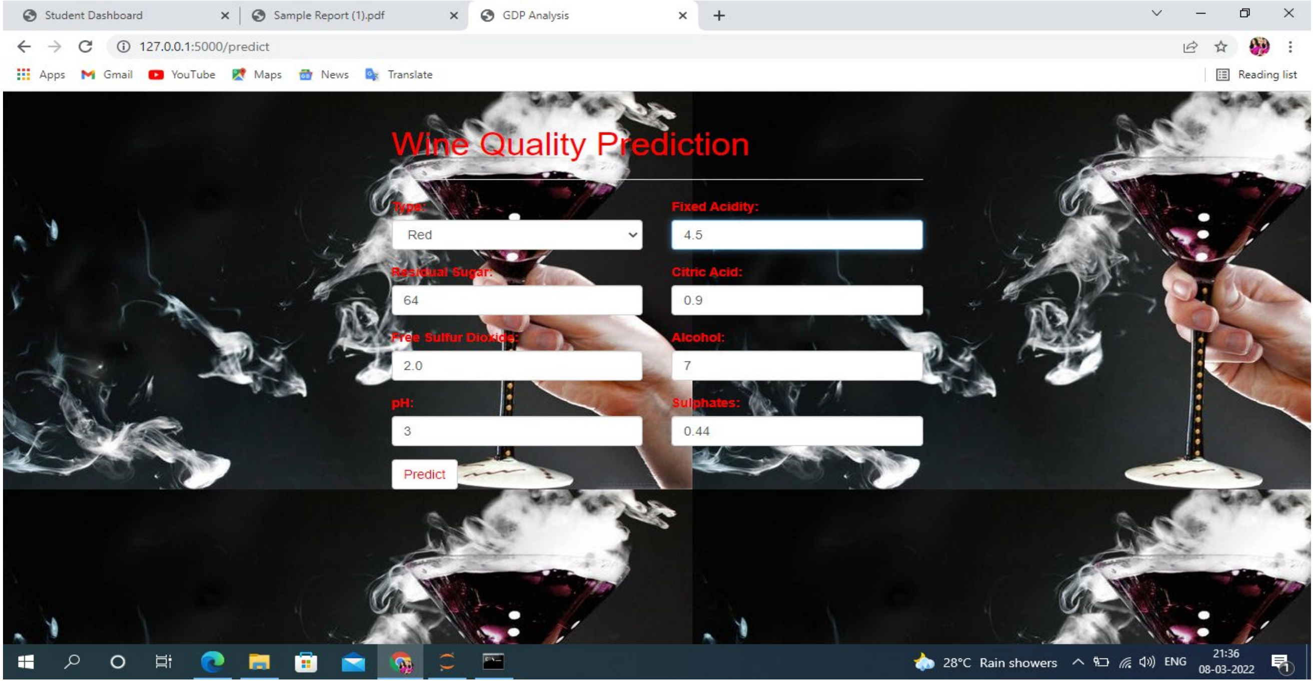
We definitely came across the fruit graphs, which is soo sweet on the test but graphs are not just to eat, they are used to make different types of things. Wine is one of them Wine is an alcoholic drink that is made up of fermented grapes. If you have come across wine then you will notice that wine has also their type they are red and white wine this was because of different varieties of graphs.

You are shocked to hear that the worldwide distribution of wine is 31 million tonnes which were huge in number.

What if you think about the quality of wine, how can you differentiate the wine according to their quality? The big question arises.

According to experts, the wine is differentiated according to its smell, flavor, and color, but we are not a wine expert to say that wine is good or bad. What will we do then? Here's the use of Machine Learning comes, yes you are thinking to write we are using machine learning to check wine quality.

Entering the values:



Wine Quality Prediction

Type: Red

Fixed Acidity: 4.5

Residual Sugar: 64

Citric Acid: 0.9

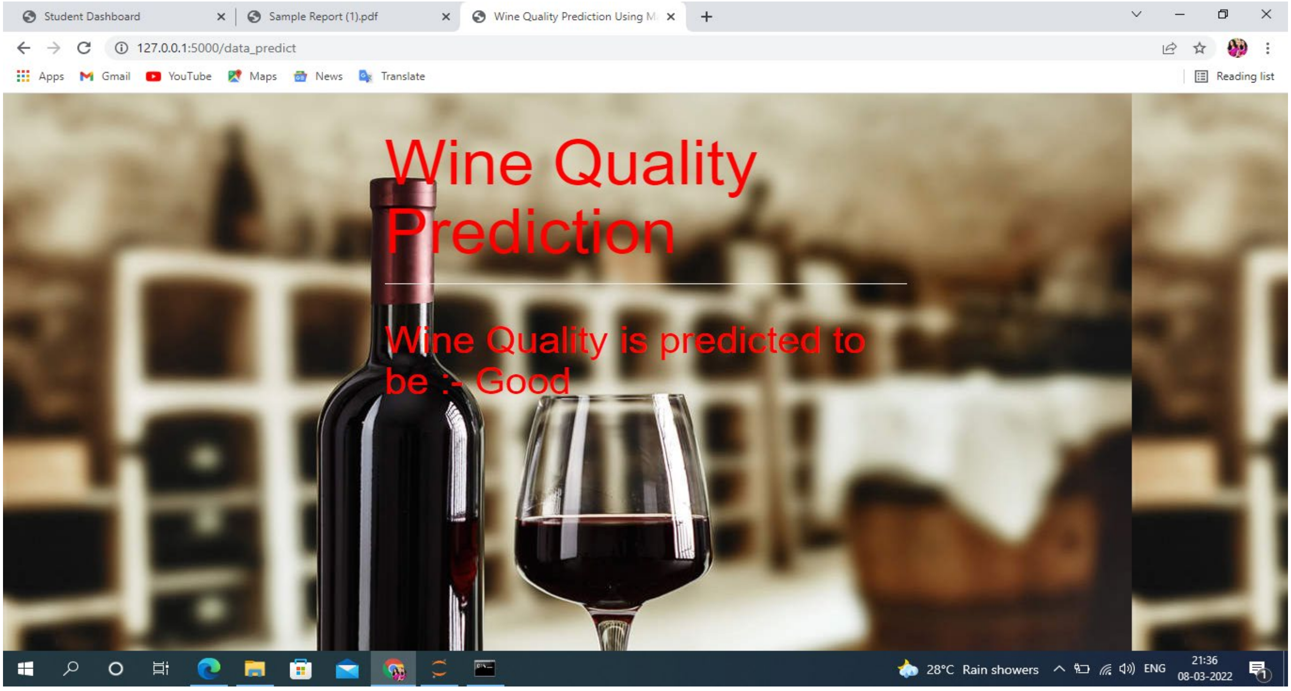
Free Sulfur Dioxide: 2.0

Alcohol: 7

pH: 3

Sulphates: 0.44

Predicted the Quality:



Wine Quality Prediction

Wine Quality is predicted to be :- Good

7 ADVANTAGES AND DISADVANTAGES

- Advantages

- Efficient Platform for quality prediction
- Accurate output is produced
- Will predict quality of wine correctly
- Relatively inexpensive and fast
- Alignment of Strategy and Results

- Disadvantages

- Longer time for getting consensus
- Uncertain environment
- All values should known

8 APPLICATION

- Wine shop
- Malls

9 CONCLUSION

Using this project we can predict the quality of wine most accurately than doing it manually.It takes the dataset values like residual sugar,pH,volatile acid,citric acid,sulphates,chlorides etc to classify the wine .According to these values we get to know the quality of wine is good or bad.This project helps to know the quality of wine according to its smell,flavour,colour and taste.

10 FUTURE SCOPE

Wine quality prediction project allows people and many wine shops to efficiently predict the quality of wine for future growth and manage its reliability. And this makes customers know well about the quality of wine before buying.This can ensure only good quality of wine can be marketed.This makes customers more trustable and reliable towards buying a wine.

11 BIBLIOGRAPHY

- https://www.analyticsvidhya.com/blog/2021/04/wine-quality-prediction-using-machine-learning/
- https://youtu.be/CBxJuwrGrc4
- https://youtu.be/W25TEa93T_I
- https://www.geeksforgeeks.org/wine-quality-prediction-machine-learning/

12 APPENDIX

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
From sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.linear_model import SGDClassifier
from sklearn.model_selection import GridSearchCV, cross_val_score
from sklearn.svm import SVC
import pickle
data = pd.read_csv(r'C:\Users\Rajalakshmi Gopinathan\Wine Quality Prediction\Dataset\winequalityN.csv')
data.head()
data.columns
data.describe()
data.info()
data.isnull().sum()
data['quality'].value_counts()
plt.figure(figsize=(12,5))
sns.distplot(data['alcohol'],color='r')
plt.show()
df_cat = data.select_dtypes(include='object')
df_cat.head()
plt.figure(figsize=(20,5))
for i,j in enumerate(df_cat):
    plt.subplot(1,4,i+1)
   sns.countplot(data[j])
```

```
axarr = data.hist(column=['quality'], bins=100, figsize=(6, 6))
ax = axarr.flatten()[0]
ax.set_xlabel(f"{ax.get_title()} value")
ax.set_ylabel("Quantity")
title = ax.get_title()
ax.set_title(f"Histogram of {title}")
plt.show()
plt.figure(figsize=(10,5))
sns.countplot(data['quality'],hue=data['type'])
plt.legend(loc='upper right')
plt.scatter(data['quality'], data['fixed acidity'], color = 'green')
plt.title('relation of fixed acidity with wine')
plt.xlabel('quality')
plt.ylabel('fixed acidity')
plt.legend()
plt.show()
plt.bar(data['quality'], data['alcohol'], color = 'maroon')
plt.title('relation of alcohol with wine')
plt.xlabel('quality')
plt.ylabel('alcohol')
plt.legend()
plt.show()
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'citric acid', data = data)
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'residual sugar', data = data)
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'sulphates', data = data)
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'free sulfur dioxide', data = data)
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'sulphates', data = data)
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'sulphates', data = data)
```

```
f, ax = plt.subplots(figsize=(10, 8))

corr = data.corr()

sns.heatmap(corr, mask=np.zeros_like(corr, dtype=np.bool),
cmap=sns.diverging_palette(220, 10, as_cmap=True),

        square=True, ax=ax)

plt.figure(figsize = (20, 10))

sns.heatmap(data.corr().abs(), annot = True)

plt.show()

plt.figure(figsize=(15,7))

data.describe()

data.head()

data=data.drop(['volatile acidity','total sulfur dioxide','chlorides','density'],axis=1)

print(data.shape)

data['quality']=data['quality'].map({3:'bad',4:'bad',5:'bad',6:'good',7:'good',8:'good'})

data['quality'].value_counts()

data['type'].value_counts()

data.isnull().any()

data.isnull().sum()

data["fixed acidity"].fillna(data["fixed acidity"].mean(),inplace=True)

data["sulphates"].fillna(data["sulphates"].mean(),inplace=True)

data["pH"].fillna(data["pH"].mean(),inplace=True)

data["residual sugar"].fillna(data["residual sugar"].mean(),inplace=True)

data["citric acid"].fillna(data["citric acid"].mean(),inplace=True)

data["quality"].fillna(data["quality"].mode()[0],inplace=True)

ata.isnull().any()

le=LabelEncoder()

data['quality']=le.fit_transform(data['quality'])

data['type']=le.fit_transform(data['type'])

sns.countplot(data['quality'])

x=data.iloc[:,:8]

y=data.iloc[:,8:9]

print(x.shape)

print(y.shape)

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=44)

print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
model = LogisticRegression()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
model = SGDClassifier(penalty=None)
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print(classification_report(y_test, y_pred))
model = SVC()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))
param = {
    'C': [0.8,0.9,1,1.1,1.2,1.3,1.4],
    'kernel':['linear', 'rbf'],
    'gamma' :[0.1,0.8,0.9,1,1.1,1.2,1.3,1.4]
}
grid_svc = GridSearchCV(model, param_grid = param, scoring = 'accuracy', cv = 10)
#grid_svc.fit(x_train, y_train)
#grid_svc.best_params_
model2 = SVC(C = 1.4, gamma = 0.1, kernel = 'rbf')
model2.fit(x_train, y_train)
y_pred = model2.predict(x_test)
```

```
print(classification_report(y_test, y_pred))
model = DecisionTreeClassifier()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
model_eval = cross_val_score(estimator = model, X = x_train, y = y_train, cv = 10)
model_eval.mean()
rfmodel = RandomForestClassifier(n_estimators = 200)
rfmodel.fit(x_train, y_train)
y_pred = rfmodel.predict(x_test)
print("Training accuracy :", rfmodel.score(x_train, y_train))
print("Testing accuracy :", rfmodel.score(x_test, y_test))
classification_report(y_test, y_pred)
confusion_matrix(y_test, y_pred)
model_eval = cross_val_score(estimator = rfmodel, X = x_train, y = y_train, cv = 5)
model_eval.mean()
def logisticRegression(x_train, x_test, y_train, y_test):
    model = LogisticRegression()
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    print('***logisticRegression***')
    print("Training accuracy :", model.score(x_train, y_train))
    print("Testing accuracy :", model.score(x_test, y_test))
    print(classification_report(y_test, y_pred))
    print(confusion_matrix(y_test, y_pred))
def SGD(x_train, x_test, y_train, y_test):
    model = SGDClassifier(penalty=None)
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    print('***Stochastic Gradient Descent Classifier***')
    print("Training accuracy :", model.score(x_train, y_train))
    print("Testing accuracy :", model.score(x_test, y_test))
    print(classification_report(y_test, y_pred))
    print(confusion_matrix(y_test, y_pred))
def SVClassifier(x_train, x_test, y_train, y_test):
    model = SVC()
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    print('***Support Vector Classifier***')
    print("Training accuracy :", model.score(x_train, y_train))
    print("Testing accuracy :", model.score(x_test, y_test))
    print(classification_report(y_test, y_pred))
    print(confusion_matrix(y_test, y_pred))
def decisionTree(x_train, x_test, y_train, y_test):
    dt=DecisionTreeClassifier()
    dt.fit(x_train,y_train)
    yPred = dt.predict(x_test)
    print('***DecisionTreeClassifier***')
    print("Training accuracy :", dt.score(x_train, y_train))
    print("Testing accuracy :", dt.score(x_test, y_test))
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
def randomForest(x_train, x_test, y_train, y_test):
    rf = RandomForestClassifier()
    rf.fit(x_train,y_train)
    yPred = rf.predict(x_test)
    print('***RandomForestClassifier***')
    print("Training accuracy :", rf.score(x_train, y_train))
    print("Testing accuracy :", rf.score(x_test, y_test))
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

```
def xgboost(x_train, x_test, y_train, y_test):
    xg = GradientBoostingClassifier()
    xg.fit(x_train,y_train)
    yPred = xg.predict(x_test)
    print('***GradientBoostingClassifier***')
    print("Training accuracy :", xg.score(x_train, y_train))
    print("Testing accuracy :", xg.score(x_test, y_test))
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
    print("Testing accuracy :", xg.score(x_test, y_test))
def compareModel(x_train, x_test, y_train, y_test):
    logisticRegression(x_train, x_test, y_train, y_test)
    print('-'*100)
    SGD(x_train, x_test, y_train, y_test)
    print('-'*100)
    SVClassifier(x_train, x_test, y_train, y_test)
    print('-'*100)
    decisionTree(x_train, x_test, y_train, y_test)
    print('-'*100)
    randomForest(x_train, x_test, y_train, y_test)
    print('-'*100)
    xgboost(x_train, x_test, y_train, y_test)
    print('-'*100)
compareModel(x_train, x_test, y_train, y_test)
pickle.dump(rfmodel,open('wineQuality_new.pkl','wb'))
```