Machine learning model deployment using IBM Watson Studio

Wine quality prediction

Objective:

The objective of wine quality prediction is to develop a model that can accurately assess and predict the quality of wine based on various input parameters. This can be valuable for winemakers, wine enthusiasts, and consumers alike, as it helps in making informed decisions about wine production, purchasing, and consumption. The goal is to create a reliable and efficient predictive system that can classify wines into different quality categories.

Definition:

Wine quality prediction involves the use of data analysis and machine learning techniques to evaluate and rate wines based on their characteristics and attributes. This is typically done by assigning a numerical or categorical score to wines, such as a quality rating or a class label. The prediction model takes into account various features of the wine, including its chemical composition, sensory properties, and potentially external factors like weather conditions during grape cultivation.

Ideate:

1. Data Collection: Gather a dataset containing information on a variety of wines, including details on their chemical composition (e.g., pH levels, alcohol content), sensory properties (e.g., aroma, taste), and possibly external factors (e.g., climate data, grape variety). This data serves as the foundation for training and testing the predictive model.

2. Preprocessing: Clean and preprocess the data to handle missing values, outliers, and normalise the features. This step ensures that the dataset is suitable for analysis and modelling.

3. Feature Selection: Identify the most relevant features that are likely to influence wine quality. Feature selection helps reduce dimensionality and improves model accuracy.

4. Model Selection: Choose appropriate machine learning algorithms for wine quality prediction. Common choices include decision trees, random forests, support vector machines, and neural networks.

Empathise:

To better understand the significance of wine quality prediction, it's important to empathise with different stakeholders:

1. Winemakers: They can benefit from this technology by optimising the winemaking process and identifying which factors contribute to higher quality wines.

2. Wine Enthusiasts: Wine lovers can make more informed decisions when purchasing wine, leading to a more satisfying and personalised wine experience.

3. Distributors and Retailers: They can use wine quality predictions to better stock and promote wines that are likely to receive positive reviews.

4. Consumers: Individuals looking for wines that match their taste preferences can use wine quality predictions to select wines that suit their palate.

Overall, wine quality prediction is a data-driven approach to enhance the wine industry, from production to consumption, and provide a more enjoyable and informed wine experience for all involved.

Design thinking process:

- Understand the Problem: Start by gaining a deep understanding of the problem. In this case, it's predicting wine quality based on various factors like acidity, alcohol content, and more.
- User Research: Talk to winemakers, sommeliers, and wine enthusiasts to gather insights into what makes a wine "high quality." This helps you define your criteria for quality.
- Problem Statement: Clearly define the problem you're trying to solve. For example, "How might we predict wine quality using machine learning based on various wine attributes?"
- Constraints and Scope: Define the data sources, technology, and resources available for your project.
- Create a Minimal Viable Model (MVM): Build a simple model using a small dataset and basic features to test your initial ideas.

- Data Collection: Collect wine quality data, including attributes like acidity, alcohol, pH, and quality ratings from winemakers or databases.

Project objective:

1. Data Collection:

Gather a dataset containing information about various attributes of wines, such as acidity levels, residual sugar, alcohol content, pH, and so on. This dataset should also include the corresponding quality ratings given by experts or consumers.

2. Data Preprocessing:

Clean the data by handling missing values, removing outliers, and normalising or standardising features. Data preprocessing is crucial for ensuring the quality and reliability of the predictive model.

3. Exploratory Data Analysis (EDA):

Explore the dataset to gain insights into the relationships between different features and wine quality. EDA helps in understanding the data distribution, correlations, and identifying patterns that can be useful for feature selection and engineering.

4. Feature Selection and Engineering:

Identify relevant features that have a significant impact on wine quality. Feature engineering involves creating new features from existing ones to improve the model's performance.

5. Model Selection:

Choose an appropriate machine learning algorithm for the prediction task. Common algorithms for regression tasks (predicting a numerical value like wine quality) include linear regression, decision trees, random forests, support vector machines, and neural networks.

6. Model Training:

Split the dataset into training and testing sets. Train the selected model on the training data, tuning hyperparameters as necessary to optimize performance. Use cross-validation techniques to ensure the model generalizes well to unseen data.

7. Model Evaluation:

Evaluate the trained model's performance using appropriate metrics, such as mean squared error (MSE) or root mean squared error (RMSE) for regression tasks. Compare the predicted wine quality values with the actual ratings to assess the model's accuracy.

8. Fine-Tuning and Optimization:

Refine the model further by fine-tuning hyperparameters or trying different algorithms. Optimization is an iterative process aimed at achieving the best possible performance.

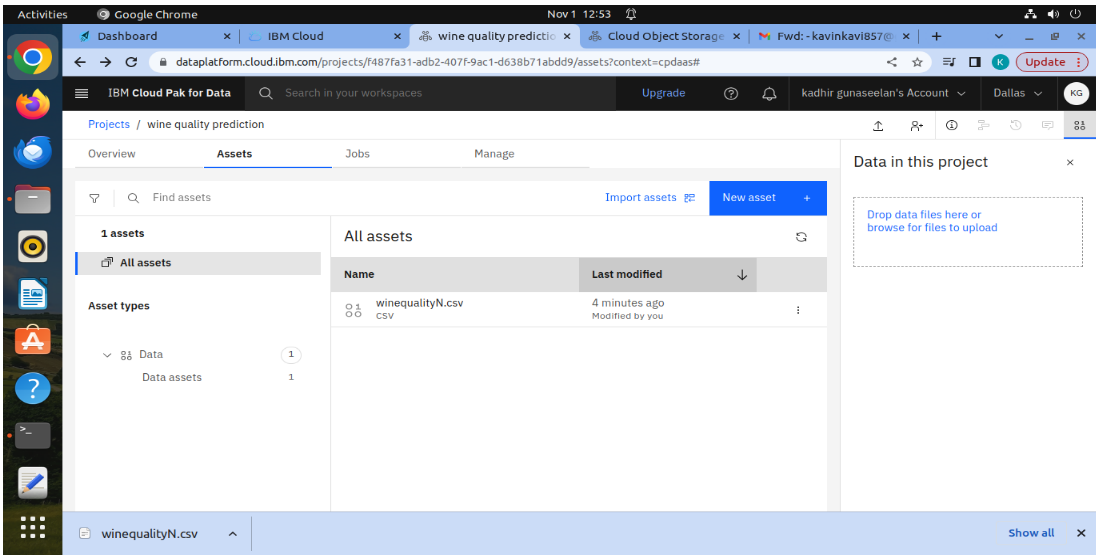
9. Deployment:

Once you have a satisfactory model, deploy it to make predictions on new, unseen wine samples. This deployment can be done through various platforms or APIs, depending on your project requirements.

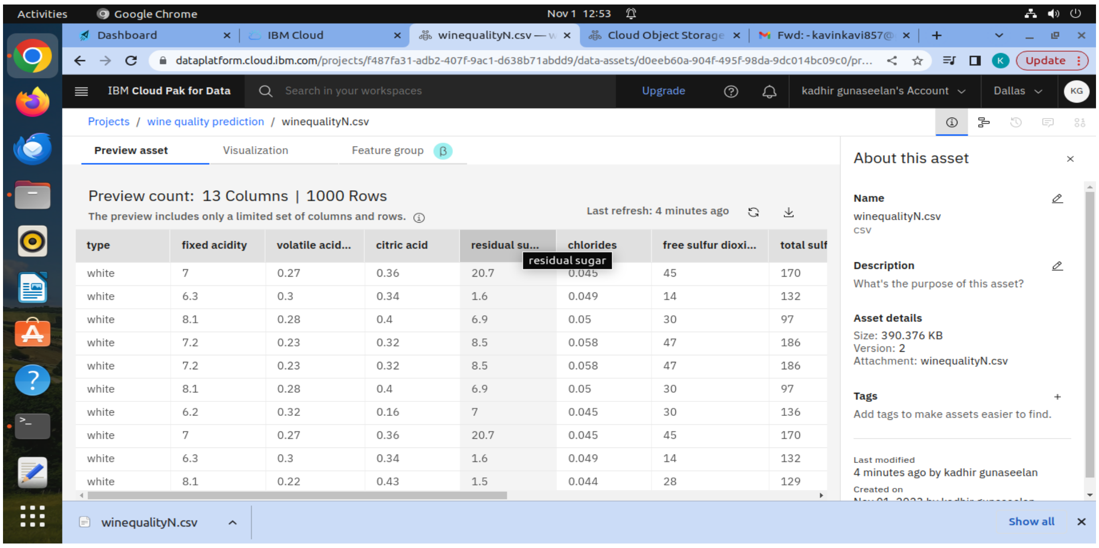
10. Monitoring and Maintenance:

Continuously monitor the model's performance over time and update it as needed. Data patterns can change, and retraining the model with new data ensures its accuracy and relevance.

Here we will predict the quality of wine on the basis of given features. We use the wine quality dataset available .This dataset has the fundamental features which are responsible for affecting the quality of the wine. By the use of several Machine learning models, we will predict the quality of the wine.



Name	Last modified
winequalityN.csv CSV	4 minutes ago Modified by you



Preview count: 13 Columns | 1000 Rows

The preview includes only a limited set of columns and rows.

Last refresh: 4 minutes ago

type	fixed acidity	volatile acid...	citric acid	residual su...	chlorides	free sulfur dioxi...	total sulf
white	7	0.27	0.36	20.7	0.045	45	170
white	6.3	0.3	0.34	1.6	0.049	14	132
white	8.1	0.28	0.4	6.9	0.05	30	97
white	7.2	0.23	0.32	8.5	0.058	47	186
white	7.2	0.23	0.32	8.5	0.058	47	186
white	8.1	0.28	0.4	6.9	0.05	30	97
white	6.2	0.32	0.16	7	0.045	30	136
white	7	0.27	0.36	20.7	0.045	45	170
white	6.3	0.3	0.34	1.6	0.049	14	132
white	8.1	0.22	0.43	1.5	0.044	28	129

About this asset

Name
winequalityN.csv
csv

Description
What's the purpose of this asset?

Asset details
Size: 390.376 KB
Version: 2
Attachment: winequalityN.csv

Tags
Add tags to make assets easier to find.

Last modified
4 minutes ago by kadhir gunaseelan

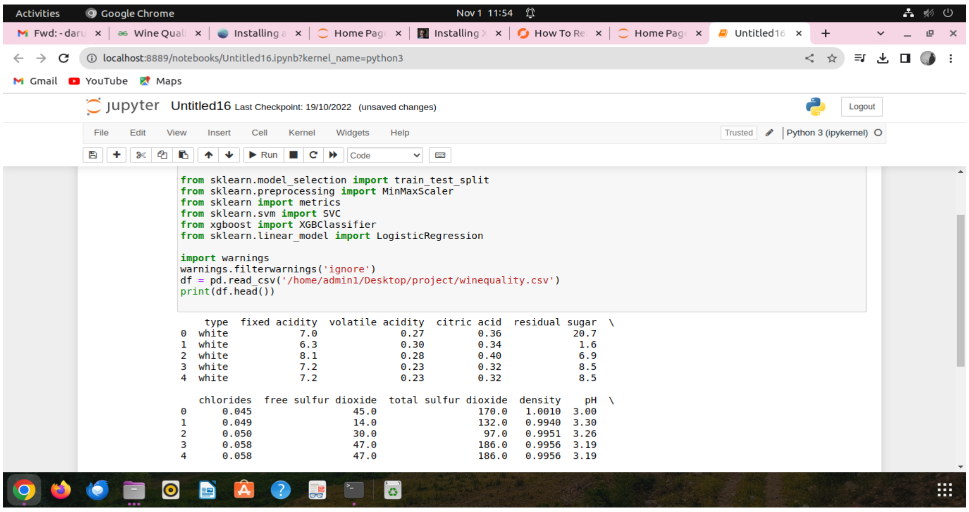
Created on

Importing libraries and Dataset:

- Pandas is a useful library in data handling.
- Numpy library used for working with arrays.
- Seaborn/Matplotlib are used for data visualisation purposes.
- Sklearn – This module contains multiple libraries having pre-implemented functions to perform tasks from data preprocessing to model development and evaluation.
- XGBoost – This contains the eXtreme Gradient Boosting machine learning algorithm which is one of the algorithms which helps us to achieve high accuracy on prediction.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn import metrics
from sklearn.svm import SVC
from xgboost import XGBClassifier
from sklearn.linear_model import LogisticRegression

import warnings
warnings.filterwarnings('ignore')
```

Now let's look at the first five rows of the dataset.

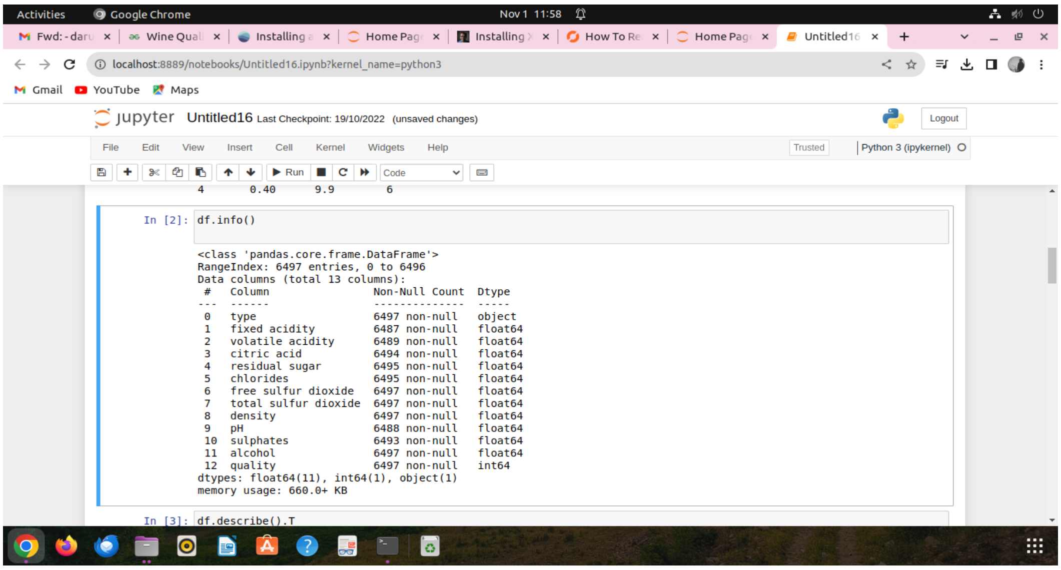


```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn import metrics
from sklearn.svm import SVC
from xgboost import XGBClassifier
from sklearn.linear_model import LogisticRegression

import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv('/home/admin1/Desktop/project/winequality.csv')
print(df.head())
```

```
    type  fixed acidity  volatile acidity  citric acid  residual sugar  \
0  white            7.0              0.27         0.36            20.7
1  white            6.3              0.30         0.34             1.6
2  white            8.1              0.28         0.40             6.9
3  white            7.2              0.23         0.32             8.5
4  white            7.2              0.23         0.32             8.5

   chlorides  free sulfur dioxide  total sulfur dioxide  density    pH  \
0      0.045                 45.0                 170.0   1.0010  3.00
1      0.049                 14.0                 132.0   0.9940  3.30
2      0.050                 30.0                  97.0   0.9951  3.26
3      0.058                 47.0                 186.0   0.9956  3.19
4      0.058                 47.0                 186.0   0.9956  3.19
```

Let's explore the type of data present in each of the columns present in the dataset.

```
df = pd.read_csv('winequality.csv')
print(df.head())
```



```
 4       0.40      9.9        6
```

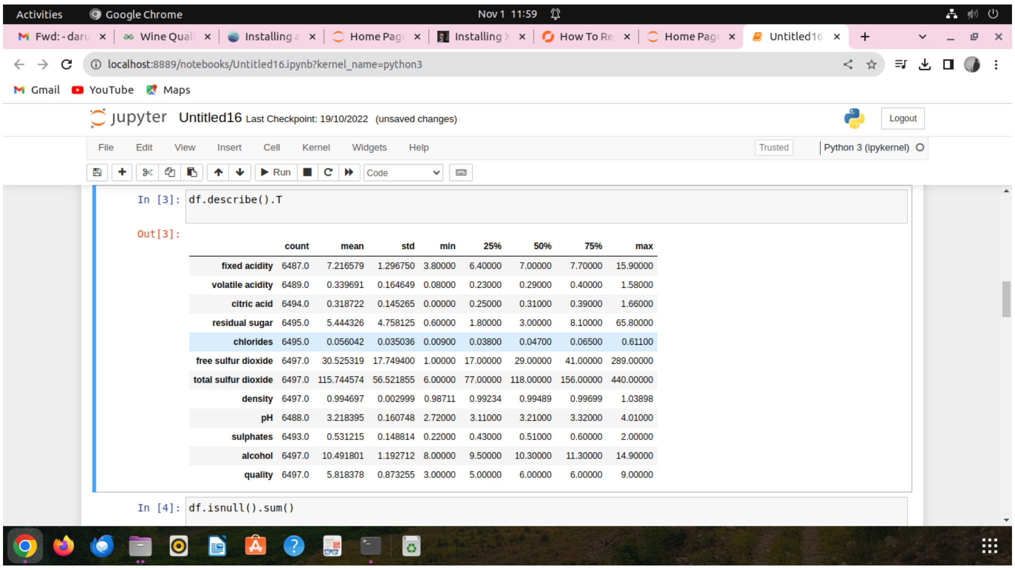
In [2]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6497 entries, 0 to 6496
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   type                  6497 non-null   object 
 1   fixed acidity         6487 non-null   float64
 2   volatile acidity      6489 non-null   float64
 3   citric acid           6494 non-null   float64
 4   residual sugar        6495 non-null   float64
 5   chlorides             6495 non-null   float64
 6   free sulfur dioxide   6497 non-null   float64
 7   total sulfur dioxide  6497 non-null   float64
 8   density               6497 non-null   float64
 9   pH                    6488 non-null   float64
 10  sulphates             6493 non-null   float64
 11  alcohol               6497 non-null   float64
 12  quality               6497 non-null   int64  
dtypes: float64(11), int64(1), object(1)
memory usage: 660.0+ KB
```

In [3]:
```
df.describe().T
```

Now we’ll explore the descriptive statistical measures of the dataset.

df.describe().T



In [3]:
```
df.describe().T
```

Out[3]:

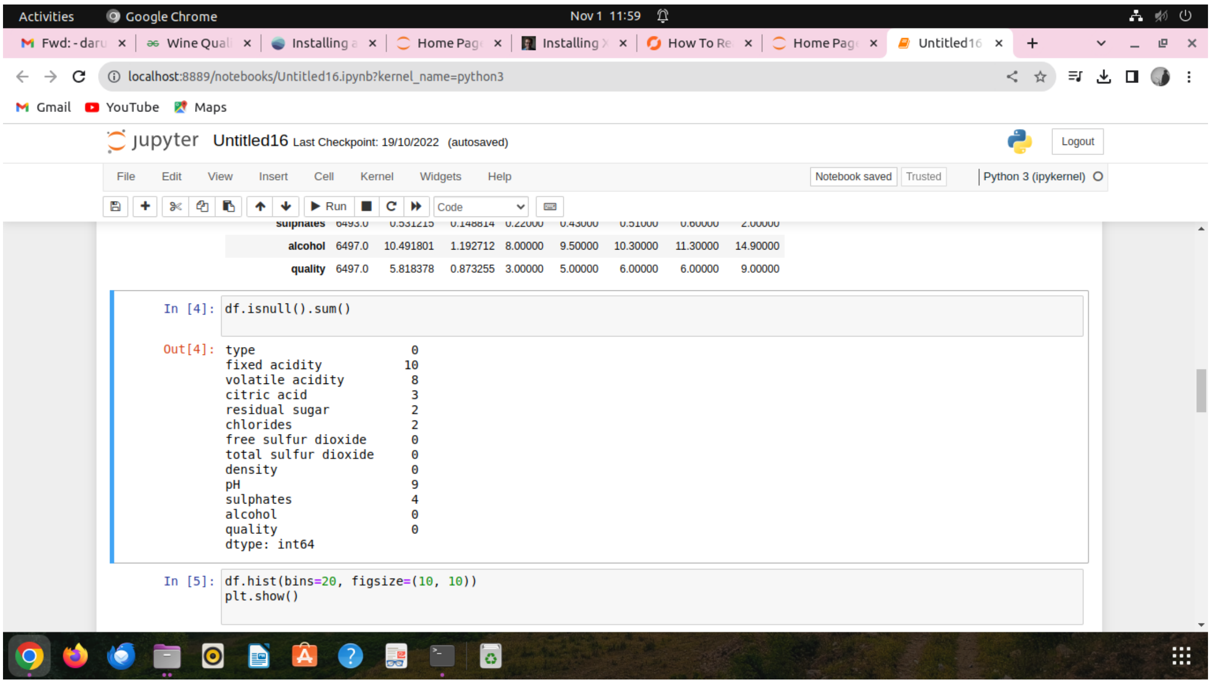
	count	mean	std	min	25%	50%	75%	max
fixed acidity	6487.0	7.216579	1.296750	3.80000	6.40000	7.00000	7.70000	15.90000
volatile acidity	6489.0	0.339691	0.164649	0.08000	0.23000	0.29000	0.40000	1.58000
citric acid	6494.0	0.318722	0.145265	0.00000	0.25000	0.31000	0.39000	1.66000
residual sugar	6495.0	5.444326	4.758125	0.60000	1.80000	3.00000	8.10000	65.80000
chlorides	6495.0	0.056042	0.035036	0.00900	0.03800	0.04700	0.06500	0.61100
free sulfur dioxide	6497.0	30.525319	17.749400	1.00000	17.00000	29.00000	41.00000	289.00000
total sulfur dioxide	6497.0	115.744574	56.521855	6.00000	77.00000	118.00000	156.00000	440.00000
density	6497.0	0.994697	0.002999	0.98711	0.99234	0.99489	0.99699	1.03898
pH	6488.0	3.218395	0.160748	2.72000	3.11000	3.21000	3.32000	4.01000
sulphates	6493.0	0.531215	0.148814	0.22000	0.43000	0.51000	0.60000	2.00000
alcohol	6497.0	10.491801	1.192712	8.00000	9.50000	10.30000	11.30000	14.90000
quality	6497.0	5.818378	0.873255	3.00000	5.00000	6.00000	6.00000	9.00000

In [4]:
```
df.isnull().sum()
```

Exploratory Data Analysis

EDA is an approach to analysing the data using visual techniques. It is used to discover trends, and patterns, or to check assumptions with the help of statistical summaries and graphical representations. Now let’s check the number of null values in the dataset columns.

df.isnull().sum()

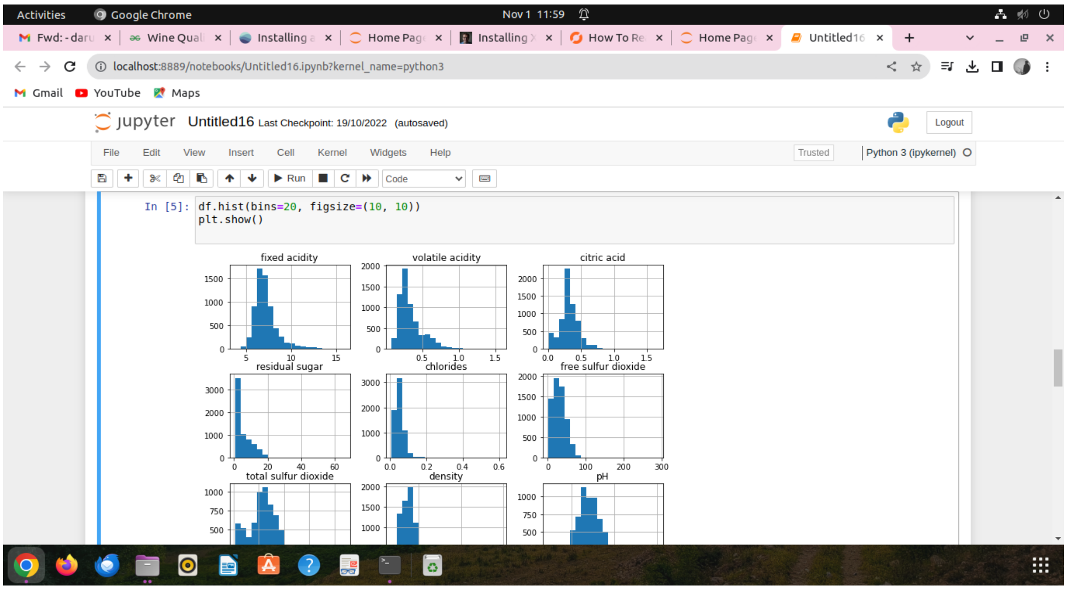


sulphates	6493.0	0.531215	0.148814	0.22000	0.43000	0.51000	0.60000	2.00000
alcohol	6497.0	10.491801	1.192712	8.00000	9.50000	10.30000	11.30000	14.90000
quality	6497.0	5.818378	0.873255	3.00000	5.00000	6.00000	6.00000	9.00000

```
In [4]: df.isnull().sum()

Out[4]: type                     0
        fixed acidity           10
        volatile acidity         8
        citric acid              3
        residual sugar           2
        chlorides                2
        free sulfur dioxide      0
        total sulfur dioxide     0
        density                  0
        pH                       9
        sulphates                4
        alcohol                  0
        quality                  0
        dtype: int64

In [5]: df.hist(bins=20, figsize=(10, 10))
        plt.show()
```

Let’s draw the histogram to visualise the distribution of the data with continuous values in the columns of the dataset.

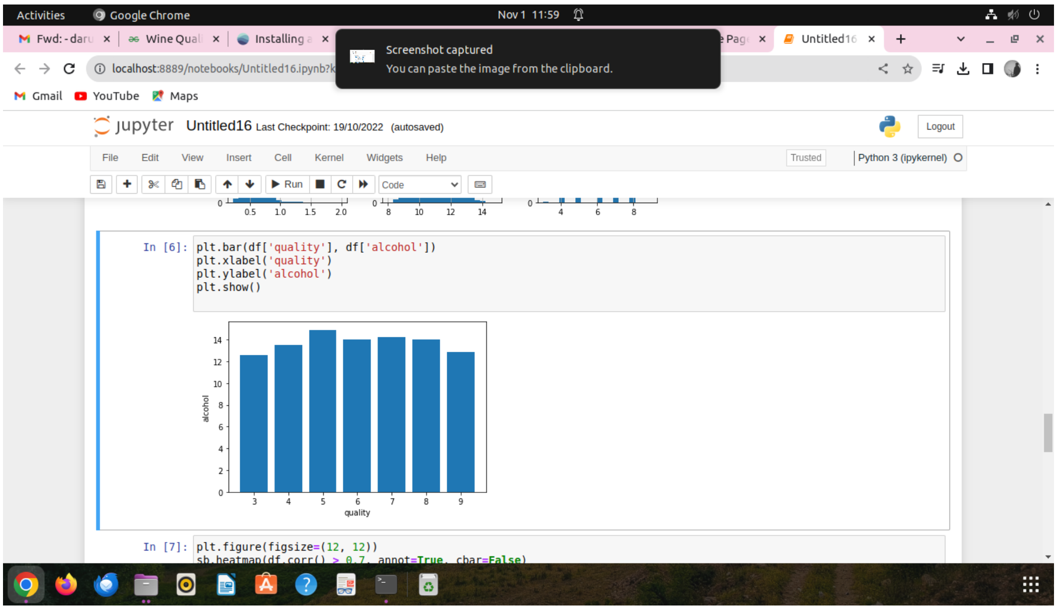
df.hist(bins=20, figsize=(10, 10))
plt.show()



```
In [5]: df.hist(bins=20, figsize=(10, 10))
        plt.show()
```

Now let’s draw the count plot to visualise the number data for each quality of wine.

```
plt.bar(df['quality'], df['alcohol'])
plt.xlabel('quality')
plt.ylabel('alcohol')
plt.show()
```



Screenshot captured
You can paste the image from the clipboard.

```
In [6]: plt.bar(df['quality'], df['alcohol'])
        plt.xlabel('quality')
        plt.ylabel('alcohol')
        plt.show()
```

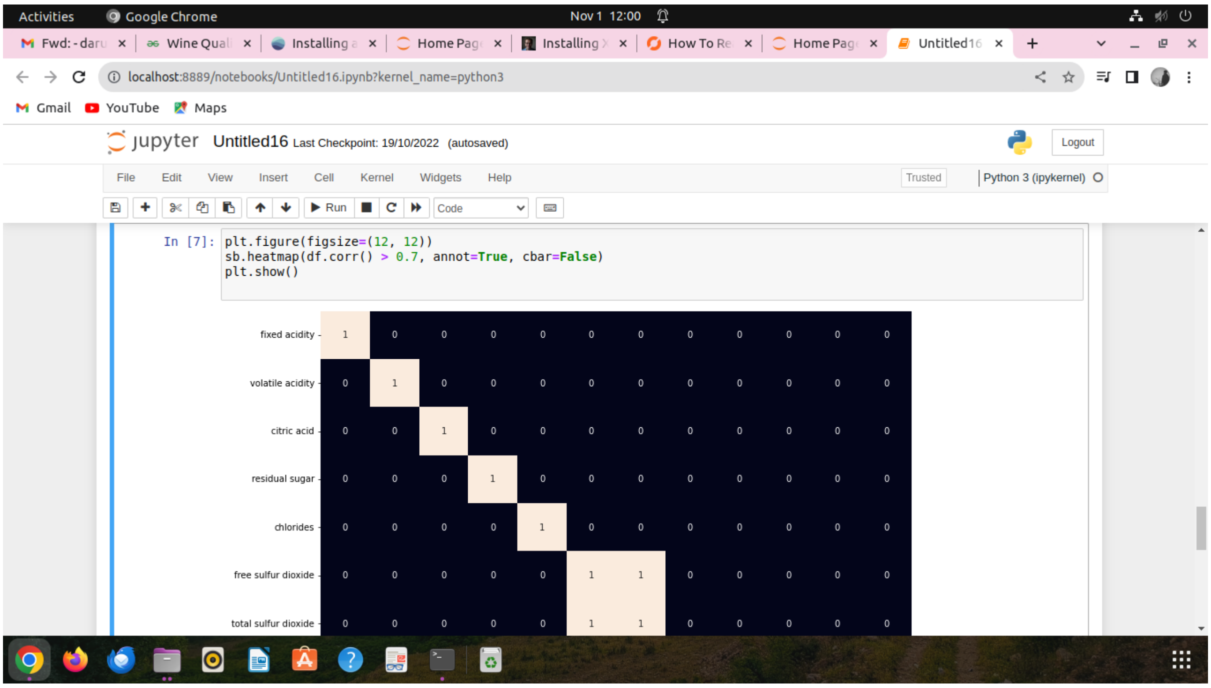
```
In [7]: plt.figure(figsize=(12, 12))
        sb.heatmap(df.corr() > 0.7, annot=True, cbar=False)
```

There are times the data provided to us contains redundant features they do not help with increasing the model’s performance. That is why we remove them before using them to train our model.

plt.figure(figsize=(12, 12))

sb.heatmap(df.corr() > 0.7, annot=True, cbar=False)

plt.show()



Activities
Google Chrome
Nov 1 12:00
localhost:8889/notebooks/Untitled16.ipynb?kernel_name=python3
Gmail
YouTube
Maps
Jupyter Untitled16 Last Checkpoint: 19/10/2022 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
fixed acidity 1 0 0 0 0 0 0 0 0 0 0 0
volatile acidity 0 1 0 0 0 0 0 0 0 0 0 0
citric acid 0 0 1 0 0 0 0 0 0 0 0 0
residual sugar 0 0 0 1 0 0 0 0 0 0 0 0
chlorides 0 0 0 0 1 0 0 0 0 0 0 0
free sulfur dioxide 0 0 0 0 0 1 1 0 0 0 0 0
total sulfur dioxide 0 0 0 0 0 1 1 0 0 0 0 0

```
In [7]: plt.figure(figsize=(12, 12))
        sb.heatmap(df.corr() > 0.7, annot=True, cbar=False)
        plt.show()
```

Conclusion:

Predicting wine quality is a complex task that involves various factors such as grape variety, climate, soil conditions, and winemaking techniques. In the context of data science and machine learning, researchers and practitioners have developed models to predict wine quality based on these factors. Here is a possible conclusion for a wine quality prediction study:

In conclusion, the study aimed to predict wine quality using machine learning techniques and relevant features such as acidity, residual sugar, alcohol content, and pH levels. Through the analysis of the dataset and the implementation of various algorithms such as regression, random forest, or neural networks, the study demonstrated the potential of these models in predicting wine quality with a reasonable level of accuracy.

The results showed that certain features, such as alcohol content and volatile acidity, had a significant impact on wine quality, confirming their importance in winemaking. The models developed in this study, after rigorous testing and evaluation, performed well in predicting wine quality within the given dataset. However, it's crucial to note that the models' accuracy might vary when applied to different datasets or real-world scenarios due to variations in grape varieties, winemaking processes, and regional differences.

Additionally, this study highlights the importance of data preprocessing, feature selection, and hyperparameter tuning in improving the predictive performance of machine learning models. Regular updates and retraining of models with new data can further enhance their accuracy and reliability.

In practical terms, the developed models can be valuable tools for winemakers and vineyard owners to assess and improve their wine quality. By understanding the impact of different factors on wine quality, producers can make informed decisions during the winemaking process, ultimately leading to the production of higher-quality wines that cater to consumers' preferences.

However, it's important to acknowledge the limitations of this study. The quality of wine is a subjective measure and can vary based on individual preferences. While the models provide valuable insights, they should be used as supportive tools rather than definitive judgments of a wine's quality. Moreover, ongoing research and data collection are essential to refine and improve these models, ensuring their applicability and accuracy in the dynamic and diverse world of winemaking.

In summary, the study successfully demonstrated the potential of machine learning techniques in predicting wine quality. While there are challenges and limitations, the findings contribute to the growing body of knowledge in the field of wine science and data analytics. As technology advances and more data becomes available, further research in this area can lead to more sophisticated models, providing even deeper insights into the intricate relationship between factors and wine quality.