Name : Viven Hotwani
Class : D15C
Roll No : 17

Assignment No: 2

Q.1: Use the following data set for question 1

82, 66, 70, 59, 90, 78, 76, 95, 99, 84, 88, 76, 82, 81, 91, 64, 79, 76, 85, 90

1. **Find the Mean**

To find the mean, sum all values and divide by the number of values:
Sum = 82 + 66 + 70 + 59 + 90 + 78 + 76 + 95 + 99 + 84 + 88 + 76 + 82 + 81 + 91 + 64 + 79 + 76 + 85 + 90
> Sum = 1611

2. **Find the Median**

To find the median, first arrange the data in ascending order: 59, 64, 66, 70, 76, 76, 76, 78, 79, 81, 82, 82, 84, 85, 88, 90, 90, 91, 95, 99
Since there are 20 values (an even number), the median is the average of the 10th and 11th values: Median = (81 + 82) / 2
> Median = 81.5

3. **Find the Mode**

The mode is the value that appears most frequently in the dataset. Looking at the ordered data: 59, 64, 66, 70, 76, 76, 76, 78, 79, 81, 82, 82, 84, 85, 88, 90, 90, 91, 95, 99
The value 76 appears three times, while all other values appear once or twice.
> Mode = 76

4. **Find the Interquartile range**

To find the IQR, I need to calculate Q1 (25th percentile) and Q3 (75th percentile).
For Q1: With 20 values, Q1 is the median of the first 10 values: 59, 64, 66, 70, 76, 76, 76, 78, 79, 81 Q1 = (70 + 76) / 2 = 73
For Q3: Q3 is the median of the last 10 values: 82, 82, 84, 85, 88, 90, 90, 91, 95, 99 Q3 = (88 + 90) / 2 = 89
IQR = Q3 - Q1 = 89 - 73 = 16
> Q1=73, Q3=89, IQR=16

Q.2 For each tool listed above:

- identify the target audience
- discuss the use of this tool by the target audience
- identify the tool's benefits and drawbacks

1. Machine Learning for Kids

- **Target Audience:** Primary and secondary school students (K-12), Teachers and educators with limited coding experience, Parents interested in introducing ML concepts to children

- **Use:** Students use visual blocks-based programming (Scratch) to create and train simple ML models, Teachers incorporate it into STEM curricula to introduce AI/ML concepts, Students learn by creating interactive projects like games, stories, and simple applications that use ML

- **Benefits:**
  - Simplifies complex ML concepts through visual programming
  - No coding experience required to get started
  - Provides real hands-on experience with ML model training
  - Designed specifically for educational contexts
  - Focuses on ethical AI use and understanding
- **Drawbacks:**
  - Limited in complexity compared to professional ML tools
  - Simplified models may not translate directly to real-world applications
  - Requires teacher supervision for younger students
  - Limited model types and capabilities

2. Teachable Machine

- **Target Audience:** Non-technical users interested in ML, Educators at various levels, Artists and creative technologists, Students (middle school through university), Professionals looking to prototype ML solutions quickly

- **Use:** Quick creation of custom image, sound, or pose recognition models, Integration into creative projects and installations, Classroom demonstrations of ML concepts, Rapid prototyping of ML ideas without coding

- **Benefits:**

    - Extremely accessible with no coding required
    - Browser-based with no software installation
    - Real-time training and feedback
    - Models can be exported for use in other applications
    - Free to use
- **Drawbacks:**
    - Limited to specific model types (image, audio, pose)
    - Less customization than professional ML platforms
    - Models may not be as robust as those built with more advanced tools
    - Limited control over model architecture

2. From the two choices listed below, how would you describe each tool listed above? Why did you choose the answer?

Predictive analytics

- **Description:** Predictive analytics uses statistical algorithms, machine learning, and data mining techniques to analyze historical data and make predictions about future events or behaviors. It helps forecast trends and outcomes by identifying patterns and relationships in data. For example, it can be used to predict customer behavior, stock market trends, or equipment failure.
- **Why Predictive analytics:** Predictive analytics is typically chosen when there is a need to anticipate future events, understand trends, and make data-driven decisions. Businesses use it for risk management, forecasting, marketing strategies, and optimizing operations.

Descriptive analytic

- **Description:** Descriptive analytics involves the process of analyzing historical data to gain insights into what has already happened. It summarizes past events through methods such as reporting, data visualization, and basic statistical analysis. Descriptive analytics answers questions like "What happened?" and "Why did it happen?" It typically includes measures like averages, counts, or percentages.
- **Why choose it:** Descriptive analytics is selected when the goal is to understand past performance or identify trends that have already occurred. It is used for performance tracking, monitoring key metrics, and summarizing large datasets for easy comprehension.

3. **From the three choices listed below, how would you describe each tool listed above? Why did you choose the answer?**

Supervised Learning:

- **Description:** In supervised learning, the model is trained on labeled data. That means the training data includes both the inputs (features) and the correct outputs (labels). The model learns by comparing its predictions to the true labels, and over time, it adjusts to minimize errors.
- **Why choose it**: If we think of how a machine learning model might be trained to predict outcomes based on past data, this corresponds to supervised learning. It's about learning from past examples where the correct answers are already known.

Unsupervised Learning:

- **Description:** In unsupervised learning, the model works with data that has no labels or predefined outcomes. The goal is to find structure or patterns in the data, such as clustering similar items or reducing dimensionality.
- **Why choose it**: This is useful for scenarios where you don’t know the exact relationships beforehand, and you're interested in letting the model explore the data's underlying structure, like grouping similar data points or uncovering hidden patterns.

Reinforcement Learning:

- **Description:** In reinforcement learning, an agent learns by interacting with an environment. It makes decisions and gets feedback in the form of rewards or penalties. The agent's goal is to maximize cumulative reward over time.
- **Why choose it**: This is like training a model to perform tasks based on trial and error, receiving rewards for making good decisions and penalties for bad ones. It's about optimizing actions for long-term success, much like training an AI to play a game or control a robot.

Q.3 Data Visualization: Read the following two short articles:

- Read the article Kakande, Arthur. February 12. “What’s in a chart? A Step-by-Step Guide to Identifying Misinformation in Data Visualization.” *Medium*
- Read the short web page Foley, Katherine Ellen. June 25, 2020. “How bad Covid-19 data visualizations mislead the public.” *Quartz*
- Research a current event which highlights the results of misinformation based on data visualization. Explain how the data visualization method failed in presenting accurate information. Use newspaper articles, magazines, online news websites or any other legitimate and valid source to cite this example. Cite the news source that you found.

> Answer:

Example: Misleading Data Visualization in the Australian Broadcasting Corporation (ABC) Article on Housing Crisis

In a recent article discussing solutions to the housing crisis, the Australian Broadcasting Corporation (ABC) included a chart that received significant criticism for its misleading design. The chart aimed to illustrate the potential of utilizing existing housing stock, including unoccupied homes, to address housing shortages.

How the Data Visualization Failed:

- **Inappropriate Chart Type:** The article featured a line chart to represent data that was not suited for such a visualization. Line charts are typically used to display trends over time, but in this case, the data represented categories that did not follow a temporal sequence.
- **Misleading Representation:** By treating "total" as a regular category and plotting it on the same axis as other data points, the chart created confusion. This design choice gave an impression of a trend where none existed, leading readers to misinterpret the information.

These design flaws led to widespread criticism, with observers labeling the chart as one of the most misleading visualizations encountered. The chart has since been removed from the live article, though it remains accessible via the Internet Archive.

https://www.ft.com/content/58dce921-2cdf-4140-9eb7-77792749dc88

Q. 4 Train Classification Model and visualize the prediction performance of trained model required information

- Data File: Classification data.csv
- Class Label: Last Column
- Use any Machine Learning model (SVM, Naïve Base Classifier)

Requirements to satisfy

- Programming Language: Python
- Class imbalance should be resolved
- Data Pre-processing must be used
- Hyper parameter tuning must be used
- Train, Validation and Test Split should be 70/20/10
- Train and Test split must be randomly done
- Classification Accuracy should be maximized
- Use any Python library to present the accuracy measures of trained model

https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

-> Naïve Base Classifier:

1. **Data Loading & Pre-processing:**
    - Load the data from Classification data.csv.
    - Handle missing data if necessary.
    - Feature scaling using StandardScaler or MinMaxScaler.
    - Split data into features (X) and target (y), with y being the last column.
    - Split into training, validation, and test sets.

2. **Model Selection & Hyperparameter Tuning:**
    - Use either SVM (e.g., SVC from sklearn.svm) or Naive Bayes (e.g., GaussianNB from sklearn.naive_bayes).
    - Use GridSearchCV or RandomizedSearchCV for hyperparameter tuning.

3. **Class Imbalance Handling:**
    - Use techniques such as oversampling (e.g., SMOTE), undersampling, or assigning class weights.

4. **Model Training & Evaluation:**
    - Train the model using the training data.
    - Evaluate on the validation set.
    - Test on the test set and visualize performance using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

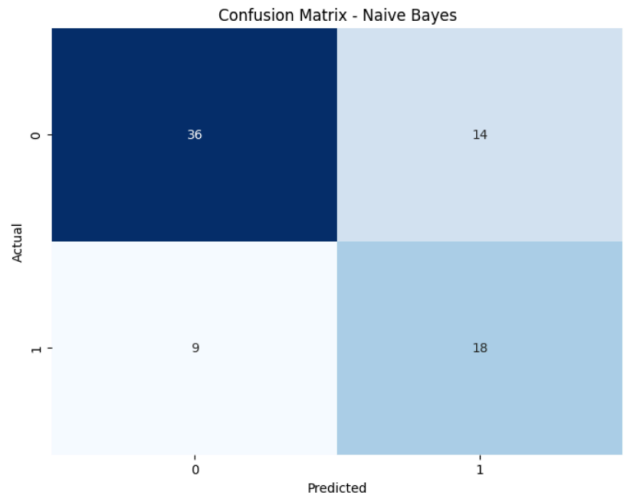
Result:

1. **Best Hyperparameters**: var_smoothing = 0.1 for Naive Bayes.
2. **Validation Accuracy**: 77.78% (performance on validation data).
3. **Test Accuracy**: 70.13% (performance on test data).
4. **Metrics:**
    a. **Class 0 (Negative outcome)**:
        i. **Precision**: 80%
        ii. **Recall**: 72%
        iii. **F1-Score**: 76%

    b. **Class 1 (Positive outcome)**:
        i. **Precision**: 56%
        ii. **Recall**: 67%
        iii. **F1-Score**: 61%

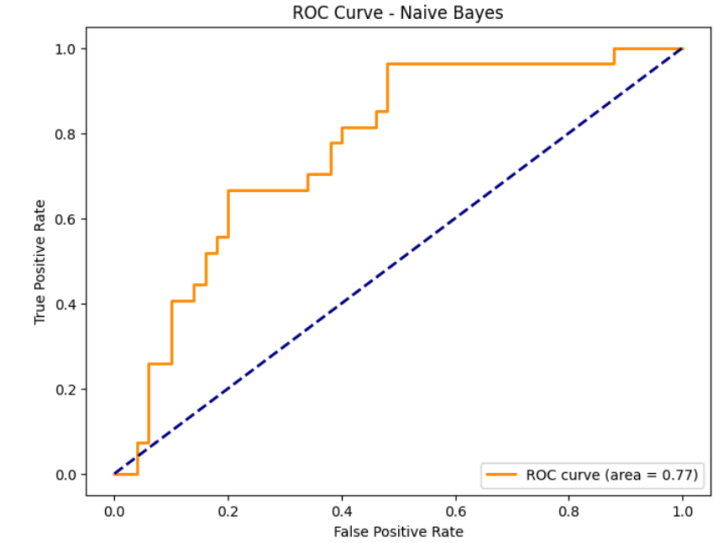
5. **Averages:**
    a. **Macro Average**: Balanced performance across both classes.
    b. **Weighted Average**: More influenced by class 0, as it has more samples.

Confusion Matrix



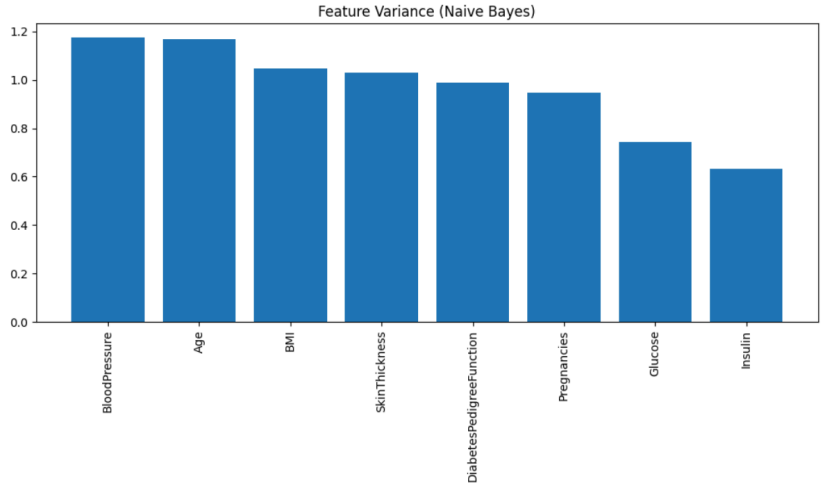
True Negatives (0 predicted as 0): 36, **False Positives: 14**, **False Negatives: 9**, **True Positives: 18**.The model performs better at predicting class 0, but struggles with class 1, showing a tendency to misclassify positives.

ROC Curve



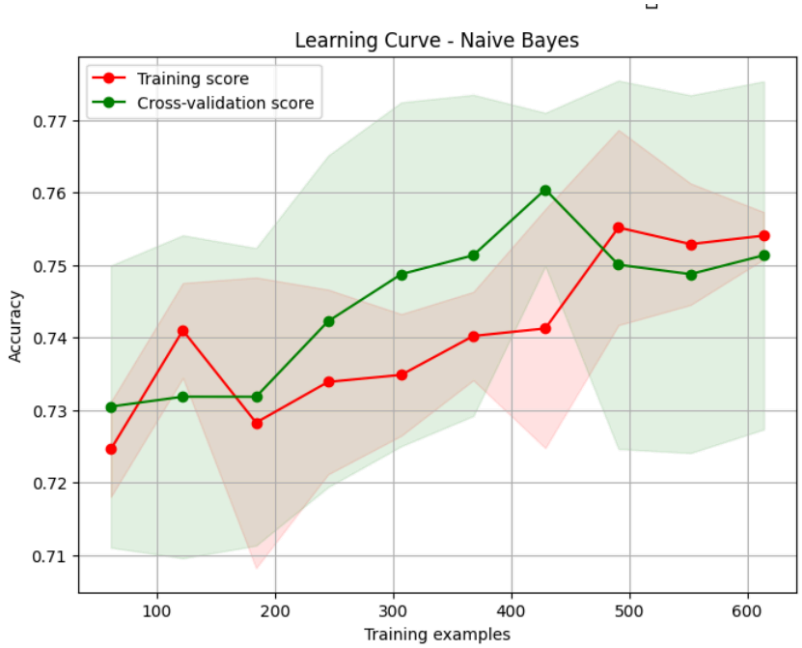
The ROC curve for the Naive Bayes model shows an **AUC of 0.77**, indicating decent performance. The model has a **77% chance of distinguishing between the two classes**, performing better than random but leaving room for improvement.

Feature Variance Visualization



As you can see, blood pressure has the highest variance at around 1.2 and insulin has the lowest at 0.6.

Learning Curve



This learning curve shows the accuracy of a Naive Bayes classifier as the number of training examples increases. The model exhibits consistent improvement in both training and cross-validation scores, indicating good generalization with more data.

Q.5 Train Regression Model and visualize the prediction performance of trained model

- Data File: Regression data.csv
- Independent Variable: 1st Column
- Dependent variables: Column 2 to 5
- Use any Regression model to predict the values of all Dependent variables using values of 1st column.

Requirements to satisfy:

- Programming Language: Python
- OOP approach must be followed
- Hyper parameter tuning must be used
- Train and Test Split should be 70/30
- Train and Test split must be randomly done
- Adjusted R2 score should more than 0.99
- Use any Python library to present the accuracy measures of trained model

https://github.com/Sutanoy/Public-Regression-Datasets

-> Step 1: Import Necessary Libraries: Import required libraries such as pandas, numpy, seaborn, matplotlib, and scikit-learn for data processing, modeling, and visualization.

Step 2: Load and Inspect the Dataset

1. Read the dataset using pd.read_csv().
2. Inspect the first few rows with data.head() to understand the structure of the dataset.
3. Check for missing values using data.isnull().sum() to ensure the data is clean.

Step 3: Split Data into Features and Target Variable: Separate the features (X) and the target variable (y). Here, X includes all the columns except ID and Classification, and y is the Classification column.

Step 4: Split Data into Training and Testing Sets: Use train_test_split() from scikit-learn to divide the data into training and testing sets. Typically, 80% of the data is used for training, and 20% is used for testing.

Step 5: Standardize the Features

1. Standardize the feature set using StandardScaler() to bring all features to the same scale (important for certain models, like Logistic Regression).
2. Apply scaling on both the training and testing sets (fit_transform() for training, and transform() for testing).

Step 6: Initialize and Train Models

- **Logistic Regression:**
    a. Initialize the Logistic Regression model using LogisticRegression().
    b. Train the model with the training data using model.fit(X_train_scaled, y_train).
- **Random Forest Classifier:**
    a. Initialize the Random Forest model using RandomForestClassifier().
    b. Train the model similarly with model.fit(X_train, y_train) (no scaling needed for Random Forest).

Step 7: Make Predictions

Use both trained models to predict the target variable (y_pred_log_reg for Logistic Regression and y_pred_rf for Random Forest) on the testing set (X_test or X_test_scaled).

Step 8: Evaluate the Models

- **Logistic Regression:**
    a. Calculate the accuracy using accuracy_score(y_test, y_pred_log_reg).
    b. Generate a classification report with classification_report(y_test, y_pred_log_reg).
    c. Create and visualize the confusion matrix using confusion_matrix() and seaborn.heatmap().
- **Random Forest:**Similarly, calculate the accuracy, classification report, and confusion matrix for the Random Forest model.

Step 9: Visualize the Results

- **Confusion Matrix:**
    - Plot confusion matrices for both models using seaborn.heatmap() to visually compare the performance of both models.
- **Feature Importance (for Random Forest):**
    - Extract the feature importances using rf_model.feature_importances_.
    - Create a DataFrame to sort and plot the feature importances, visualizing which features are most impactful for the Random Forest model's predictions.
- **Model Accuracy Comparison:**
    - Create a bar plot using seaborn.barplot() to visually compare the accuracies of Logistic Regression and Random Forest models.

Result:

	ID	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	\
0	1	48	23.500000	70	2.707	0.467409	8.8071	9.702400	
1	2	83	20.690495	92	3.115	0.706897	8.8438	5.429285	
2	3	82	23.124670	91	4.498	1.009651	17.9393	22.432040	
3	4	68	21.367521	77	3.226	0.612725	9.8827	7.169560	
4	5	86	21.111111	92	3.549	0.805386	6.6994	4.819240	

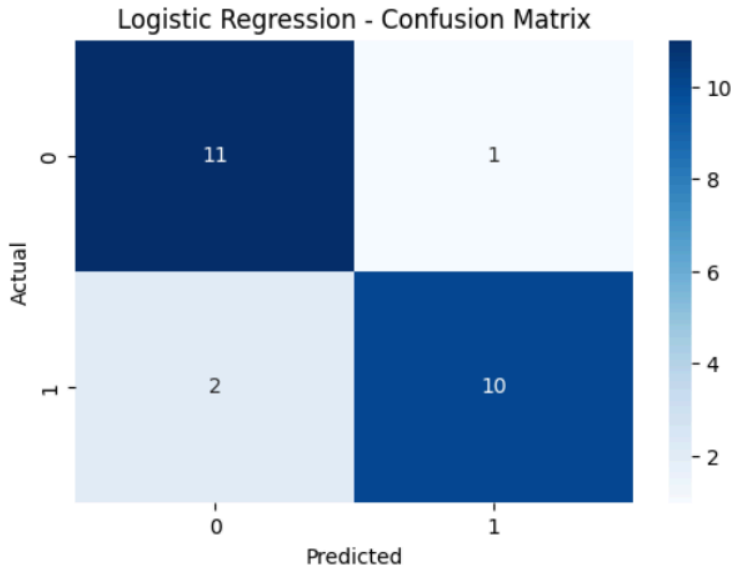
	Resistin	MCP.1	Classification
0	7.99585	417.114	0
1	4.06405	468.786	0
2	9.27715	554.697	0
3	12.76600	928.220	0
4	10.57635	773.920	0

Logistic Regression:

Logistic Regression Accuracy: 0.875

Classification Report (Logistic Regression):

	precision	recall	f1-score	support
0	0.85	0.92	0.88	12
1	0.91	0.83	0.87	12
accuracy			0.88	24
macro avg	0.88	0.88	0.87	24
weighted avg	0.88	0.88	0.87	24



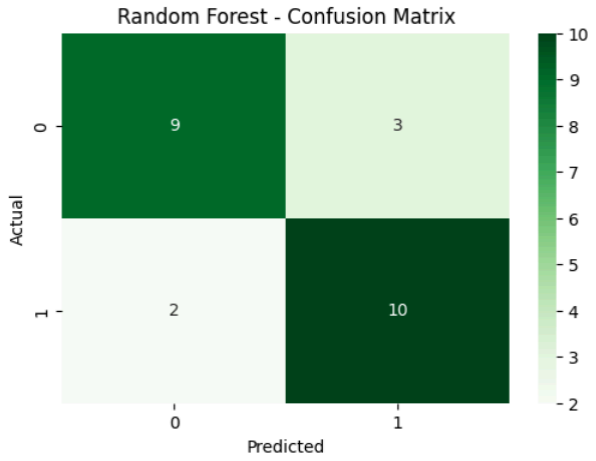
Random Forest:

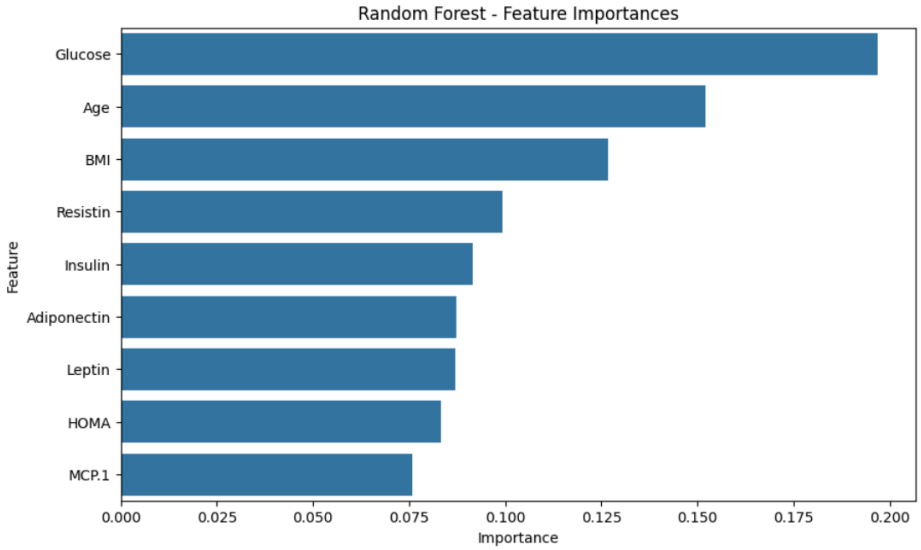
```
Random Forest Accuracy: 0.7916666666666666

Classification Report (Random Forest):
              precision    recall  f1-score   support

           0       0.82      0.75      0.78        12
           1       0.77      0.83      0.80        12

    accuracy                           0.79        24
   macro avg       0.79      0.79      0.79        24
weighted avg       0.79      0.79      0.79        24
```





Q.6 What are the key features of the wine quality data set? Discuss the importance of each feature in predicting the quality of wine? How did you handle missing data in the wine quality data set during the feature engineering process? Discuss the advantages and disadvantages of different imputation techniques. (Refer dataset from Kaggle).

Dataset link: https://www.kaggle.com/datasets/rajyellow46/wine-quality

The **Wine Quality Dataset** on Kaggle is a popular dataset for predicting wine quality based on various physicochemical features. It's used in machine learning and data analysis to classify wine quality, which is usually scored between 0 and 10. The dataset has two versions: one for **red wine** and one for **white wine**. Here are the **key features** of the dataset, their importance, and a discussion on **missing data handling** during feature engineering:

Key Features of the Wine Quality Dataset

The dataset contains several physicochemical features, which are important for determining the quality of the wine. Here's a list of the most important ones:

1. **Fixed Acidity**:
   - Represents the amount of fixed acids (e.g., tartaric acid) in the wine.
   - Importance: Affects taste, especially sourness. It can also influence the pH, which in turn impacts the stability and fermentation process of wine.

2. **Volatile Acidity**:
   - Measures the concentration of acetic acid and other volatile acids in the wine.
   - Importance: High levels of volatile acidity can cause the wine to smell vinegary and lower its quality. It's an important indicator for the perception of off-flavors.

3. **Citric Acid**:
   - A natural preservative that adds a fresh, citrusy flavor.
   - Importance: It plays a role in the overall flavor profile and helps balance out acidity in the wine. It can also contribute to the wine's stability and aging potential.

4. **Residual Sugar**:
   - The amount of sugar remaining after fermentation.
   - Importance: A higher sugar level can result in a sweeter wine, and sugar content has a direct effect on the wine's body and taste.

5. **Chlorides**:
   - Measures the concentration of salts (e.g., sodium chloride) in the wine.

- Importance: Higher chloride concentrations may indicate a salty taste, which could negatively affect wine quality.

6. **Free Sulfur Dioxide (SO2)**:
- Refers to the amount of free sulfur dioxide in the wine.
- Importance: Acts as an antioxidant and antimicrobial agent, preventing oxidation and microbial growth. However, high levels can give a pungent smell and negatively impact flavor.

7. **Total Sulfur Dioxide (SO2)**:
- Total amount of sulfur dioxide, both free and bound.
- Importance: Similar to free sulfur dioxide, but also considers bound forms. It helps to prevent spoilage, but excessive levels can affect the taste and aroma.

8. **Density**:
- Measures the wine's density, which is influenced by alcohol content, sugar concentration, and other factors.
- Importance: It's an indirect indicator of alcohol content and sugar levels, which both influence wine quality.

9. **pH**:
- Indicates the acidity or alkalinity of the wine.
- Importance: It affects the stability, flavor, and aging potential of the wine. A balanced pH contributes to better taste and preservation.

10. **Sulphates**:
- Measures the concentration of sulfur-containing compounds in the wine.
- Importance: Similar to sulfur dioxide, it has preservative properties, but excessive levels could impact flavor.

11. **Alcohol**:
- The alcohol content in the wine.
- Importance: Higher alcohol content is often associated with higher quality wine, as it affects body and mouthfeel. It also plays a role in fermentation and preservation.

12. **Quality**:
- The target variable, representing the quality of the wine (scored between 0 and 10).
- **Importance**: This is the variable we aim to predict using the other features.

Handling Missing Data During Feature Engineering

Handling missing data is a crucial part of the feature engineering process. If missing data is not addressed properly, it can significantly affect the quality of predictions and the overall model's performance. Here's how to approach missing data and the techniques for imputation:

1. **Identifying Missing Data:** Before deciding how to handle missing data, it's essential to identify which features contain missing values. This can be done using techniques like checking the percentage of missing values per feature and visualizing missing data patterns.

2. **Imputation Techniques:**

- Mean/Median Imputation:
    - Description: Replacing missing values with the mean or median of the feature. The mean is used for numerical features that are normally distributed, and the median is typically used for skewed features.
    - Advantages: Simple, fast, and easy to implement. It works well when data is missing at random and the distribution is relatively stable.
    - Disadvantages: It may introduce bias and reduce variance in the data, especially if the data is not missing at random (e.g., systematic patterns). Also, it doesn't capture the underlying distribution of the missing values.

- Mode Imputation (for categorical data):
    - Description: For categorical features, missing values can be replaced with the most frequent category.
    - Advantages: Quick and easy to apply.
    - Disadvantages: Similar to mean/median imputation, it can distort the distribution of the feature and lead to biased models if the missingness is not random.

- K-Nearest Neighbors (KNN) Imputation:
    - Description: This technique imputes missing values based on the values of similar instances (neighbors).
    - Advantages: It can provide more contextually accurate imputations by leveraging relationships between features.
    - Disadvantages: Computationally expensive, especially with large datasets. Also, the choice of "k" (the number of neighbors) can influence the results.

- Multivariate Imputation by Chained Equations (MICE):
    - Description: MICE imputes missing values by modeling each feature with missing data as a function of the other features in the dataset. It performs multiple imputations to provide a more robust estimate.
    - Advantages: It is effective when dealing with complex datasets with missing values in multiple columns. It helps preserve the relationships between features.

    - Disadvantages: More computationally intensive and might require careful parameter tuning.

- Model-based Imputation:
    - Description: Use machine learning models (e.g., regression, decision trees) to predict missing values based on other features.
    - Advantages: It can capture complex relationships in the data and provide accurate imputations.
    - Disadvantages: Can be computationally expensive and may overfit if not implemented carefully.

3. Deleting Missing Data: In cases where the missing data is minimal (say less than 1-2% of the total data), one might choose to drop the rows or columns containing missing values.

- Advantages: Simple and reduces the risk of introducing bias.
- Disadvantages: Can lead to loss of valuable information if missing data is not random or if large portions of the data are missing.

Advantages and Disadvantages of Different Imputation Techniques

- **Mean/Median Imputation**:
    - Advantages: Simple and efficient for numerical features, especially when the dataset is small and the missingness is random.
    - Disadvantages: Can distort feature distributions, leading to inaccurate predictions.
- **KNN Imputation**:
    - Advantages: Considers the similarity between data points, leading to potentially more accurate imputations.
    - Disadvantages: Computationally expensive, especially with large datasets or high-dimensional data.
- **MICE (Multiple Imputation by Chained Equations)**:
    - Advantages: Accounts for the relationships between features and performs multiple imputations to reduce bias.
    - Disadvantages: Requires more computational resources and is slower than other methods.
- **Model-based Imputation**:
    - Advantages: Captures non-linear relationships between features and produces more accurate imputations.

- Disadvantages: Can overfit if not carefully validated and is computationally demanding.

In summary, handling missing data is an essential step in the feature engineering process. The choice of imputation technique depends on the nature of the data, the amount of missing data, and the computational resources available. Each technique has its strengths and weaknesses, so understanding the characteristics of the dataset and the underlying patterns of missingness is crucial for making the right decision.