

AI & DS - I (Assignment - 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 (10pts)
2. Find the Median (10pts)
3. Find the Mode (10pts)
4. Find the Interquartile range (20pts)

Ans:

$$\text{Mean} = \frac{\sum(\text{data elements})}{\text{Number of data elements}} = \frac{(82+66+70+59+90+78+76+95+99+84+88+76+82+81+91+64+79+76+85+90)}{20}$$
$$= 1611 / 20 = 80.55$$

Mean = 80.55

For finding the median, we have to arrange the give data in ascending order.

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

Number of elements = 20.

Thus, Median = $\{(n/2)^{\text{th}} \text{ element} + [(n/2)+1]^{\text{th}} \text{ element}\} / 2$

$$n/2 = 20 / 2 = 10$$

$$\begin{aligned} \text{Median} &= (10^{\text{th}} \text{ element} + 11^{\text{th}} \text{ element})/2 \\ &= (81+82) / 2 = 163 / 2 \end{aligned}$$

Median = 81.5

The mode of a dataset with discrete elements is the element with the most amount of occurrences.

In this data, 76 is occurring the most number of times (3 times).

Thus, **Mode = 76**

Interquartile range = Q3 - Q1

Where Q3 and Q1 are the Upper and Lower quartiles respectively.

To find Q3 and Q1, we split the dataset amongst the median.

Data (Lower) = 59, 64, 66, 70, 76, 76, 76, 78, 79, 81

Data (Higher) = 82, 82, 84, 85, 88, 90, 90, 91, 95, 99

Now, Q1 = median of Lower data = $(5^{\text{th}} \text{ element} + 6^{\text{th}} \text{ element})/2 = (76+76)/2 = 76$

Q3 = median of Higher data = $(5^{\text{th}} \text{ element} + 6^{\text{th}} \text{ element})/2 = (88+90)/2 = 89$

Thus IQR = Q3 - Q1 = 89 - 76

IQR = 13

Answer:

Mean = 80.55

Median = 81.5

Mode = 76

IQR = 13

Q2] 1) Machine Learning for Kids 2) Teachable Machine

1. 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

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

- Predictive analytic
- Descriptive analytic

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

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Ans:

For **Machine Learning for Kids:**

- 1) **Target Audience:** Primarily designed for school students (ages 8–16) and educators. It is also suitable for beginners with no prior coding experience who want to explore machine learning

2) Usage:

- Provides a child-friendly interface for creating ML models using text, images, numbers, or sounds.
- Often integrated with tools like Scratch or Python to develop interactive projects (e.g., games or chatbots).
- Teachers use it to introduce fundamental ML concepts and encourage creative problem-solving.

3) Benefits:

- Free, web-based, and easily accessible.
- Promotes experiential learning through hands-on projects.
- Combines coding and ML concepts effectively.
- Supports multiple data types (text, images, numbers).

4) Drawbacks:

- Limited scope for advanced ML projects.
- Data collection and labeling can be time-consuming.
- Not suitable for real-world ML deployment.

5) This tool would be described as a **predictive analytic** tool as the models learn from labeled data to predict or classify new inputs (e.g., recognizing if text is positive or negative).

6) This tool would use **supervised learning** as it would have a set of predefined outputs set up for a set of questions and would use the inputs from the kids to compare with the predefined outputs to provide a score.

For Teachable Machine:

1) **Target Audience:** Suitable for general users, including students, educators, artists, developers, and non-coders. Ideal for quick prototyping and demonstrations.

2) Usage:

- Allows users to train ML models using images, sounds, or poses with minimal effort.
- No coding required—models can be exported for web, apps, or TensorFlow.
- Useful for quick demos, accessibility tools, and creative tech projects.

3) Benefits:

- Extremely user-friendly with drag-and-drop functionality.
- Supports multiple input types (image, audio, pose).
- Provides instant feedback and model export options.
- Great for visual learners and rapid experimentation.

4) Drawbacks:

- Models trained on small datasets may not generalize well.
 - Limited customization and no coding flexibility.
 - Not designed for production-level accuracy or fine-tuning.
- 5) This tool would be described as a **predictive analytic** tool as the models learn from labeled data/answers to predict or classify new inputs (e.g., recognizing if text is positive or negative).
- 6) This tool would use **supervised learning** as it would have a set of predefined outputs set up for a set of questions and would use the inputs from the users to compare with the predefined outputs to provide a score.

Q3] 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.

Ans:**1) What's in a chart? A Step-by-Step guide to Identifying Misinformation in Data Visualization****1. The Growing Need for Responsible Data Visualization**

With over 2.5 quintillion bytes of data generated daily, effective visualization has become crucial for making sense of complex information. However, the same tools meant to clarify data can also mislead through intentional manipulation or unintentional design flaws. As organizations like Mozilla have found, platforms and creators share responsibility for combating misinformation - making it essential for both designers and consumers to develop critical data literacy skills.

2. Common Visualization Pitfalls That Spread Misinformation

Several techniques frequently distort data representation: truncated y-axes exaggerate minor differences; misapplied color schemes reverse conventional meanings (like using red for growth); cherry-picked timeframes present misleading trends; and pie charts displaying impossible percentages (like 110%). Perhaps most dangerously, visualizations often falsely imply causation from correlation - a problem highlighted during COVID-19 when people misinterpreted coinciding data points as proof of relationships.

3. Building Defenses Against Deceptive Visuals

Combating visualization misinformation requires vigilance from both creators and consumers. Designers must maintain ethical standards by using proper scales, consistent color coding, and full-context data. Consumers should scrutinize axes, check sources, and verify whether visual claims match underlying numbers. As noted by data integrity advocates like Pollicy, developing these analytical skills represents our best defense against misleading data representations in an increasingly visual information ecosystem.

2) How bad Covid-19 data visualizations mislead the public.

1. Lack of Context and Misleading Scales

Many state dashboards presented COVID-19 data without proper context, leading to misinterpretation. For example, Arkansas visualized preexisting health conditions using percentage arcs scaled to 100%, making serious comorbidities like hypertension (9.6%) appear insignificant. In reality, this represented over 1,700 high-risk patients—a critical detail lost in the visualization. Similarly, Arizona's case charts omitted y-axis labels, making statewide cases (thousands) appear comparable to county-level data (under 100). Without clear reference points, these visualizations distorted public perception of risk (Quartz, 2020).

2. Overuse of Problematic Chart Types

Several states relied on ineffective formats like pie charts and cluttered snapshots. Alabama's daily reports used pie charts to display case demographics, despite research showing pie charts hinder proportional comparisons (Few, 2007). The state also published dense numerical snapshots without trendlines, obscuring the pandemic's progression. As Quartz's data editor noted, bar charts or tables would have better served public understanding by simplifying cognitive load (Quartz, 2020).

3. Inconsistent or Missing Reference Data

Visualizations often failed to include benchmarks like testing rates or demographic baselines. Washington succeeded by pairing positive test percentages with total tests administered, clarifying testing capacity growth. In contrast, New York's early dashboards showed case counts without adjusting for population density, exaggerating rural outbreaks. Such omissions fueled misinformation, as seen when unlabeled scales in Arizona's heatmaps obscured disproportionate Native American infection rates (NYT, 2020). These cases underscore how missing reference data can alter policy decisions and public behavior.

3) Climate change sceptics use misleading Arctic ice data to make case

In April 2024, a misleading headline from the *Daily Sceptic* claiming that "Arctic Sea Ice Soars to Highest Level for 21 Years" went viral on social media. The article compared sea ice extent on January 8, 2024, with the same date in 2004 and suggested that fears about global warming were exaggerated. This claim, despite being technically accurate for that specific date, was shared alongside comments denying climate change and accusing scientists of promoting a

false narrative. Posts like these spread misinformation by presenting isolated data points out of context and ignoring broader scientific trends.

Experts quickly pointed out that this type of comparison is a classic example of “cherry-picking.” Scientists from the National Snow and Ice Data Center (NSIDC) and the British Antarctic Survey clarified that using just one day’s data to assess long-term changes is scientifically invalid. Arctic sea ice levels fluctuate seasonally, typically peaking in March and reaching their lowest levels in September. Comparing single days across years doesn’t reflect these natural patterns. In fact, although January 8, 2024, showed slightly more sea ice than the same day in 2004, large parts of February and March 2024 had *lower* sea ice levels compared to 2004. This shows that relying on just one favorable data point gives a misleading impression.

The long-term trend, as documented by satellite records since 1979, shows a clear and consistent decline in Arctic sea ice. According to NSIDC, the average minimum ice extent dropped from 6.95 million sq km in 1979–1990 to 4.42 million sq km in 2011–2020. Climate experts emphasize that this continuous decline is far more meaningful than any short-term variation. Misleading visualizations like the one used by the *Daily Sceptic* distort public understanding and can undermine support for urgent climate action. It highlights the importance of presenting data within proper context and educating the public on how to interpret visual information critically.

Source: [Climate change sceptics use misleading Arctic ice data to make case](#)

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

[Pima Indians Diabetes Database](#)

Ans:

Dataset Description: The Pima Indians Diabetes Dataset contains 768 records of female patients aged 21+, with 8 medical features and a target column Outcome indicating diabetes (1) or not (0). It's widely used for binary classification and poses challenges like missing values, class imbalance, and non-linear relationships.

Features provided:

- **Pregnancies:**
Number of times the patient has been pregnant. It is a numeric count and can influence diabetes risk due to hormonal changes.
- **Glucose:**
Plasma glucose concentration (mg/dL) measured 2 hours after a glucose tolerance test. Higher values often indicate diabetes.
- **BloodPressure:**
Diastolic blood pressure (mm Hg). Elevated blood pressure is often correlated with diabetes and other metabolic issues.
- **SkinThickness:**
Triceps skinfold thickness (mm). It provides an estimate of body fat percentage.
- **Insulin:**
2-hour serum insulin level (mu U/mL). Helps assess insulin resistance; zero values may indicate missing data.
- **BMI:**
Body Mass Index (weight in kg/(height in m)²). A high BMI can be a risk factor for diabetes.
- **DiabetesPedigreeFunction:**
A function that scores likelihood of diabetes based on family history (genetic predisposition).
- **Age:**
Age of the patient in years. Risk of diabetes generally increases with age.
- **Outcome:**
Target variable — 1 if the patient has diabetes, 0 otherwise. Used for classification.

Models:

1) Class Imbalance:

The dataset's class imbalance issue is addressed using SMOTE (Synthetic Minority Oversampling Technique) from the *imblearn* library. After splitting and scaling the training data, SMOTE is applied to generate synthetic samples of the minority class, ensuring a balanced dataset for model training. This helps the model avoid bias toward the majority class and improves learning of decision boundaries for underrepresented classes. The balanced dataset is then used to train both SVM and Naïve Bayes classifiers, leading to more robust and fair performance during validation and testing phases.

```
Class distribution before SMOTE:  
Outcome  
0    350  
1    187  
Name: count, dtype: int64  
  
Class distribution after SMOTE:  
Outcome  
0    350  
1    350
```

Applying SMOTE to balance the classes significantly improved the models' ability to recognize minority class instances. Without this step, the model would likely have favored the majority class, leading to skewed predictions and poor recall for the minority class.

2) Data Preprocessing:

Data preprocessing is essential to make raw features suitable for machine learning models. In this case, the features were scaled using `StandardScaler`, which standardizes each feature to have zero mean and unit variance. This normalization step ensures that all attributes like Glucose, BMI, and Insulin contribute equally to model learning, rather than allowing features with larger magnitudes to dominate.

3) Train Test Validation splitting/Random splitting

The train, validation, and test split (70/20/10) is achieved by first splitting the dataset into training + validation (70%) and test (10%) using `train_test_split`. Then, the training data is further divided into training (about 62.22%) and validation (about 17.77%) sets, resulting in approximately 70% for training, 20% for validation, and 10% for testing. The splitting is done randomly using `train_test_split`, ensuring a random distribution of data for each subset. The parameter `random_state=42` ensures reproducibility of the splits, making sure the data is consistently divided across different runs.

4) Models used:

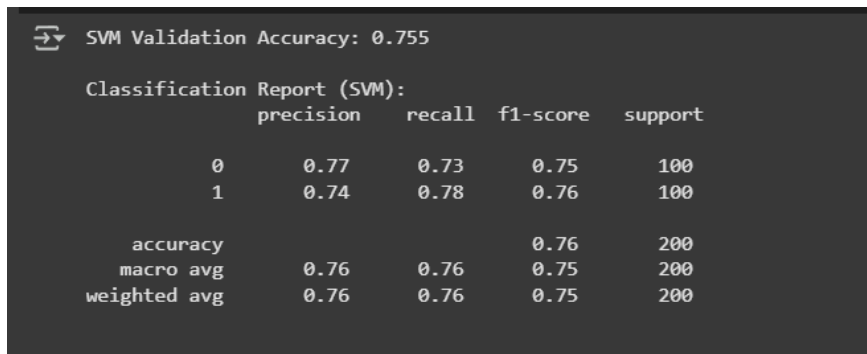
- **SVM** is a powerful classification model that finds the optimal hyperplane to separate classes. It was tuned using `GridSearchCV` for optimal parameters like `C`, kernel, and gamma. SVM works well for high-dimensional data and can handle both linear and non-linear boundaries.
- **Naïve Bayes** assumes feature independence and works well with imbalanced datasets. The Gaussian Naïve Bayes model assumes features follow a normal distribution. Despite its simplicity, it performed better than SVM, making it the final model for testing. Both models were evaluated on their classification performance.

Validation Performance:

The validation performance of both models varied significantly. The Support Vector Machine (SVM) achieved a validation accuracy of 70.77%, which, while decent, was lower than expected given its ability to handle high-dimensional data. SVM's

performance was limited by class imbalance and the need for careful hyperparameter tuning. On the other hand, Naïve Bayes outperformed SVM with a validation accuracy of 77.27%, demonstrating its robustness in handling imbalanced datasets despite the simplifying assumption of feature independence. This highlights Naïve Bayes' strength in such contexts, where its simplicity and efficiency provide a better fit for the data.

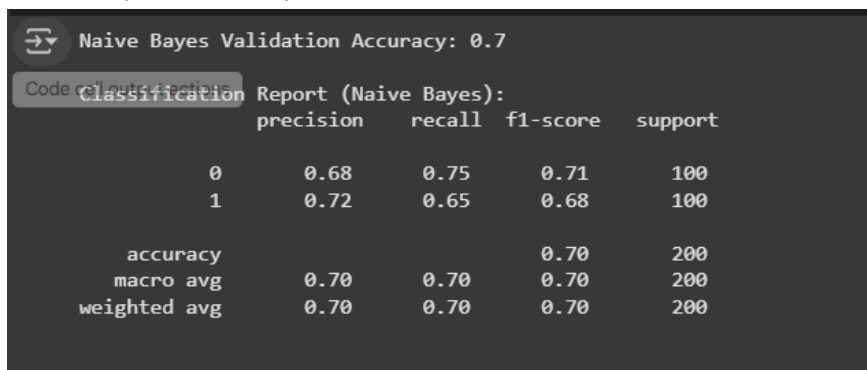
SVM Validation Accuracy:



The screenshot shows a terminal window with the title 'SVM Validation Accuracy: 0.755'. Below the title is a 'Classification Report (SVM):' table. The table has columns for 'precision', 'recall', 'f1-score', and 'support'. The rows are for classes '0' and '1', and summary statistics: 'accuracy', 'macro avg', and 'weighted avg'.

	precision	recall	f1-score	support
0	0.77	0.73	0.75	100
1	0.74	0.78	0.76	100
accuracy			0.76	200
macro avg	0.76	0.76	0.75	200
weighted avg	0.76	0.76	0.75	200

Naive-Bayes Accuracy:



The screenshot shows a terminal window with the title 'Naive Bayes Validation Accuracy: 0.7'. Below the title is a 'Classification Report (Naive Bayes):' table. The table has columns for 'precision', 'recall', 'f1-score', and 'support'. The rows are for classes '0' and '1', and summary statistics: 'accuracy', 'macro avg', and 'weighted avg'.

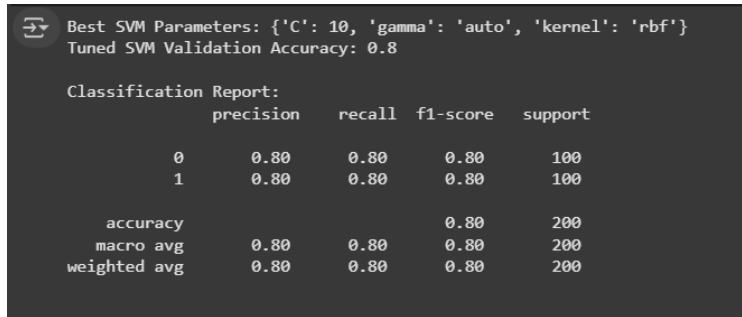
	precision	recall	f1-score	support
0	0.68	0.75	0.71	100
1	0.72	0.65	0.68	100
accuracy			0.70	200
macro avg	0.70	0.70	0.70	200
weighted avg	0.70	0.70	0.70	200

As it is required to select the model giving a better accuracy of predictions, we would be choosing the SVM Model for further performance.

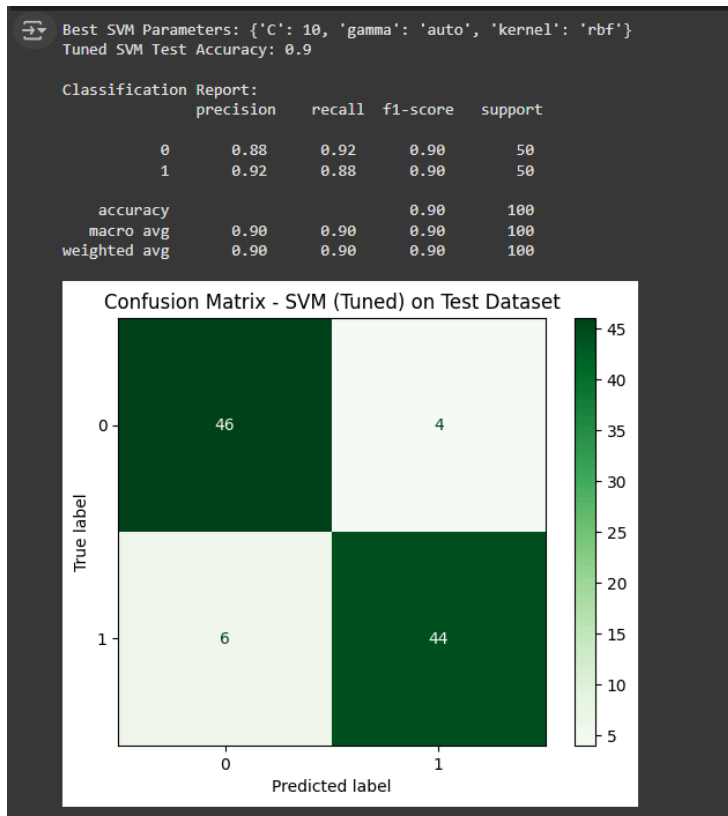
5) Hyper parameter tuning

Hyperparameter tuning involves optimizing the parameters of a machine learning model to achieve the best performance. In this case, GridSearchCV was used to tune the hyperparameters of the Support Vector Machine (SVM) model, specifically the regularization parameter C, kernel type, gamma, and polynomial degree. The goal was to find the optimal configuration that maximized the model's validation accuracy.

SVM Accuracy (After Hyperparameter Tuning):



Also, after running this same code on the test dataset, we have achieved a high accuracy of 90%.



Using this method (SVM with Hyperparameter tuning) we have achieved a high accuracy of Validation accuracy at 80% and Test Accuracy of 90%. These accuracies are also supported by the other performance parameters such as precision, recall and f1-score.

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>

<https://raw.githubusercontent.com/selva86/datasets/master/BostonHousing.csv>

<https://archive.ics.uci.edu/ml/machine-learning-databases/00477/Real%20estate%20valuation%20data%20set.xlsx>

Ans:

Dataset Description:

The Dry Bean Dataset consists of 13,611 samples of seven types of dry beans. Each sample is represented by various numerical features extracted from images through shape analysis techniques. These features provide geometric, statistical, and morphological insights into each bean's physical characteristics.

The dataset is often used for classification and regression tasks related to agricultural or computer vision applications.

Feature Description:

- Area: Total number of pixels inside the bean boundary, representing its size.
- Perimeter: Total length of the bean's outer boundary (contour).
- MajorAxisLength: Length of the major axis of the ellipse that best fits the bean.
- MinorAxisLength: Length of the minor axis of the ellipse that best fits the bean.
- Eccentricity: Measure of how elongated the shape is, ranging from 0 (circle) to 1 (line).
- ConvexArea: Number of pixels in the convex hull that encloses the bean.
- EquivDiameter: Diameter of a circle with the same area as the bean.
- Extent: Ratio of the bean area to the area of its bounding box.
- Solidity: Ratio of the bean's area to its convex area, measuring convexity.
- Compactness: Shape compactness measured using the perimeter and area.
- roundness: Indicates how close the bean shape is to a perfect circle.

- AspectRatio: Ratio of major axis length to minor axis length, indicating elongation.
- ShapeFactor1: Derived shape descriptor using area and perimeter.
- ShapeFactor2: Derived shape descriptor using perimeter and major axis.
- ShapeFactor3: Derived shape descriptor using area and minor axis.
- ShapeFactor4: Derived shape descriptor using area and major axis.
- Class: Categorical label representing the bean type (e.g., SIRA, HOROZ, etc.).

Models used:

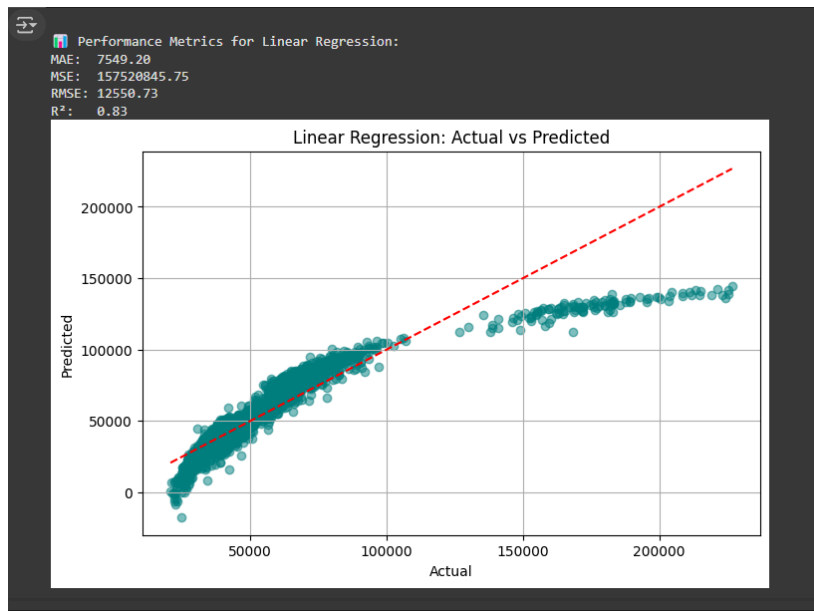
In this analysis, two regression models were employed: Linear Regression and Random Forest Regressor. Linear Regression is a simple, interpretable model that assumes a linear relationship between input features and the target variable. It serves as a good baseline but may underperform with complex, non-linear data. The Random Forest Regressor, on the other hand, is an ensemble-based model that builds multiple decision trees and averages their predictions, improving accuracy and reducing overfitting. It captures non-linear patterns effectively. Hyperparameter tuning further enhances its performance, making it more robust and reliable for real-world prediction tasks involving complex feature interactions.

Inference:

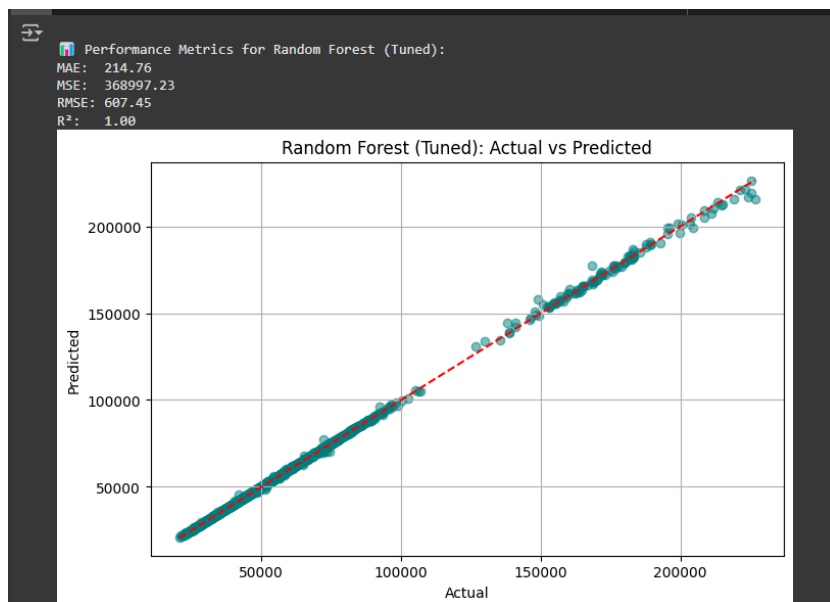
The predictive performance of both the Linear Regression and the tuned Random Forest Regressor was evaluated using key regression metrics — Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2). The Linear Regression model, though relatively simple, achieved an R^2 value of 0.83, indicating that it could explain about 83% of the variability in the bean area using the selected features. However, it also exhibited a high MAE of 7549.20 and an RMSE of 12550.73, suggesting substantial prediction errors and lower reliability for precise estimations.

On the other hand, the Random Forest Regressor, after hyperparameter tuning, significantly outperformed Linear Regression across all metrics. It achieved a near-perfect R^2 score of 1.00, indicating that it almost completely captured the variance in the target variable. The MAE dropped to just 214.76, and RMSE reduced dramatically to 607.45, reflecting far more accurate and stable predictions. This performance gap highlights that Random Forest's ensemble-based, non-linear approach is better suited for this dataset, likely because it can model complex relationships between features that a linear model cannot. Therefore, for tasks involving the prediction of bean area based on physical shape features, the Random Forest Regressor is clearly the more effective and robust choice.

Performance of linear Regression:



Performance of Random Forest Regressor:



Q6] 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).

Ans:

- **Feature Importance**

1. **Alcohol (0.476)** – Higher alcohol content is strongly associated with better wine quality.
2. **Sulphates (0.251)** – Moderate levels of sulphates contribute positively to perceived wine quality.
3. **Citric Acid (0.226)** – Citric acid enhances freshness and has a mild positive impact on quality.
4. **Fixed Acidity (0.124)** – Slightly helps in preserving wine quality, but not a major predictor.
5. **Residual Sugar (0.014)** – Has minimal influence on quality; too little or too much may be undesirable.
6. **Free Sulfur Dioxide (-0.051)** – Shows negligible negative correlation; excessive use may degrade quality.
7. **pH (-0.058)** – Very weak negative relation; lower pH (more acidic wine) could slightly improve quality.
8. **Chlorides (-0.129)** – Higher salt content tends to reduce the quality of wine.
9. **Density (-0.175)** – Denser wines tend to be of lower quality, possibly due to higher sugar content.
10. **Total Sulfur Dioxide (-0.185)** – Excessive sulfur dioxide can negatively affect wine quality and taste.
11. **Volatile Acidity (-0.391)** – Strong negative impact; higher levels often indicate spoilage or poor quality.

- **Handle Missing Data**

One common method for handling missing data is mean or median imputation, where missing values in a feature are replaced with the mean or median of that column. This technique is simple and works well when the data is normally distributed or has low variability, though it may distort variance and relationships between variables. Another method is mode imputation, used particularly for categorical data, although it can be less applicable in a numerical dataset like wine quality. K-Nearest Neighbors (KNN) imputation considers the similarity between observations and imputes values based on the average of the nearest samples. While more accurate, it is computationally expensive. Multivariate imputation techniques, such as Multiple Imputation by Chained Equations (MICE), use the relationships between multiple variables to estimate missing values, making them more robust for datasets with complex interdependencies.

- **Advantages/Disadvantages of these methods**

- 1) **Mean/Median Imputation:**

Advantages:

Mean or median imputation is one of the simplest and quickest methods to handle missing data. It helps retain the entire dataset without losing any rows and is easy to implement. Median imputation, in particular, is useful when the data contains outliers, as it is not influenced by extreme values.

Disadvantages:

This method can distort the natural variability in the data and weaken the relationships between variables, especially when a large number of values are missing. Additionally, it assumes that the data is missing completely at random, which may not always hold true in real-world scenarios.

- 2) **Mode Imputation:**

Advantages:

Mode imputation is best suited for categorical features. It is easy to use and ensures that the dataset size remains the same. This method is helpful when a particular category is dominant in the data.

Disadvantages:

It can lead to a bias toward the most frequent category, especially if the mode is overrepresented due to imputation. This may reduce the predictive power of the model, especially when diversity in categories is important.

- 3) **K-nearest Neighbours:**

Advantages:

KNN imputation considers the similarity between instances and uses neighboring data points to estimate the missing values, which often results in more accurate and context-aware imputations. It preserves the underlying structure and relationships in the data.

Disadvantages:

It can be computationally expensive, especially for large datasets, and may not perform well when many values are missing or when the data has noise. It also requires proper tuning of the number of neighbors and careful feature scaling.

- 4) **Dropping Rows/Columns**

Advantages:

Dropping rows or columns is the easiest method and ensures that the remaining data is complete. It is useful when only a very small proportion of the dataset is affected by missing values.

Disadvantages:

This method can result in significant loss of valuable information, especially when the missing data is not random or when many rows or columns are removed. It can also introduce bias if the removed data had specific patterns or relationships.