MAchine Learning Project_DOCUMENTATION

1.1>Project Overview:

This project involves analyzing and predicting house prices in Tehran based on various features such as area, number of rooms, and amenities like parking, warehouse, and elevator availability. The primary objective is to build predictive models to estimate house prices and evaluate their performance to identify the best model.

1.2>Dataset Description:

>Key Features:

- Area: The area of the house (in square meters).
- Room: Number of rooms in the house.
- Parking: Boolean indicating if the house has parking.
- Warehouse: Boolean indicating if the house has a warehouse.
- Elevator: Boolean indicating if the house has an elevator.
- Address: The locality or neighborhood of the house.
- Price: House price in Iranian Rials.
- Price (USD): House price in USD.

1.3>Data Preprocessing

1>Handling Missing Values:

- The Address column had 23 missing values, which were filled with the mode (most frequent value).

2>Removing Duplicates:

- Identified and removed 208 duplicate rows, reducing the dataset size to 3,271 entries.

3>Handling Non-Numeric Values:

- The Area column had non-numeric values, which were identified and handled by converting the column to numeric format. Any invalid entries were set to NaN and subsequently removed.

4>Outlier Treatment:

- Winsorization was applied to the Area and Price (USD) columns to cap extreme outliers (top and bottom 5%).

5>Feature Scaling:

StandardScaler was used to standardize the Area and Price (USD) columns for better model performance.

6>Label Encoding:

- Categorical columns (Parking, Warehouse, Elevator, and Address) were encoded using Label Encoding to convert them into numeric format.

1.4>Exploratory Data Analysis

1. Boxplots:
   - Boxplots were generated to visualize the distribution of numerical columns and identify outliers before and after outlier treatment.
2. Correlation Heatmap:
   - A heatmap was generated to analyze correlations between features and the target variable (Price (USD)). The Area and Room features showed the highest correlation with house prices.

1.5 >Model Building

Three machine learning models were implemented and evaluated:

1. Linear Regression:

- Performance Metrics:
- Mean Squared Error (MSE): 0.3096
- Mean Absolute Error (MAE): 0.3872
- R² Score: 0.6807
- The scatter plot showed a moderate fit between predicted and actual prices.

2. K-Nearest Neighbors (KNN)

- *Optimal k Selection:*
- The elbow method was used to identify the optimal number of neighbors, which was found to be 7.
- *Performance Metrics*:
- Mean Squared Error (MSE): 0.1341
- Mean Absolute Error (MAE): 0.2079
- R² Score: 0.8616
- The scatter plot showed a strong fit between predicted and actual prices.

3. Random Forest Regressor

- Model Configuration:
- Number of estimators: 50
- Performance Metrics:
- Mean Squared Error (MSE): 0.1793
- R² Score: 0.8151
- The scatter plot showed a good fit between predicted and actual prices, though slightly less effective than KNN.

1.6>Visualizations

1

Boxplots: Displayed the distributions and outliers of numerical features before and after treatment.

2

Heatmap: Highlighted correlations among features.

3

Scatter Plots: Illustrated the relationship between actual and predicted prices for each model.

4

Elbow Method Plot: Identified the optimal number of neighbors for the KNN model.

1.7>Results and Conclusion

>Best Performing Model

- The K-Nearest Neighbors (KNN) model outperformed the other models with:
- R^2 Score: 0.8616
- Mean Squared Error (MSE): 0.1341
- Mean Absolute Error (MAE): 0.2079

>Insights:

- The Area and Room features were the most significant predictors of house prices.
- Standardization and outlier handling significantly improved model performance.

2>Plant Disease Classification Using Image Features

2.1> Overview:

This project is focused on classifying plant diseases based on image data using deep learning-based feature extraction and machine learning models. The dataset contains labeled images of plants, with labels corresponding to four categories: healthy, rust, scab, and multiple_diseases.

2.2>Dataset :

>Description :

The dataset consists of a CSV file (train.csv) and an image folder. The CSV file contains the following columns:

- image_id: Unique identifier for each image.
- healthy, multiple_diseases, rust, scab: Binary columns indicating the presence of each condition.

>Preprocessing:

1. The healthy, multiple_diseases, rust, and scab columns were combined into a single label column.
2. Images were preprocessed and their features were extracted using the VGG16 model pretrained on ImageNet.
3. Extracted features were flattened and stored in a DataFrame.
4. Class imbalance was addressed using SMOTE, which oversampled the minority classes.

>Class Distribution

Before SMOTE:

- healthy: 516
- rust: 622
- scab: 592
- multiple_diseases: 91

After SMOTE:

- All classes balanced to 622 samples each.

>Feature Extraction

The VGG16 model was used to extract features from each image:

- Input size: 256x256x3
- Output: Flattened feature vector of size 32,768.

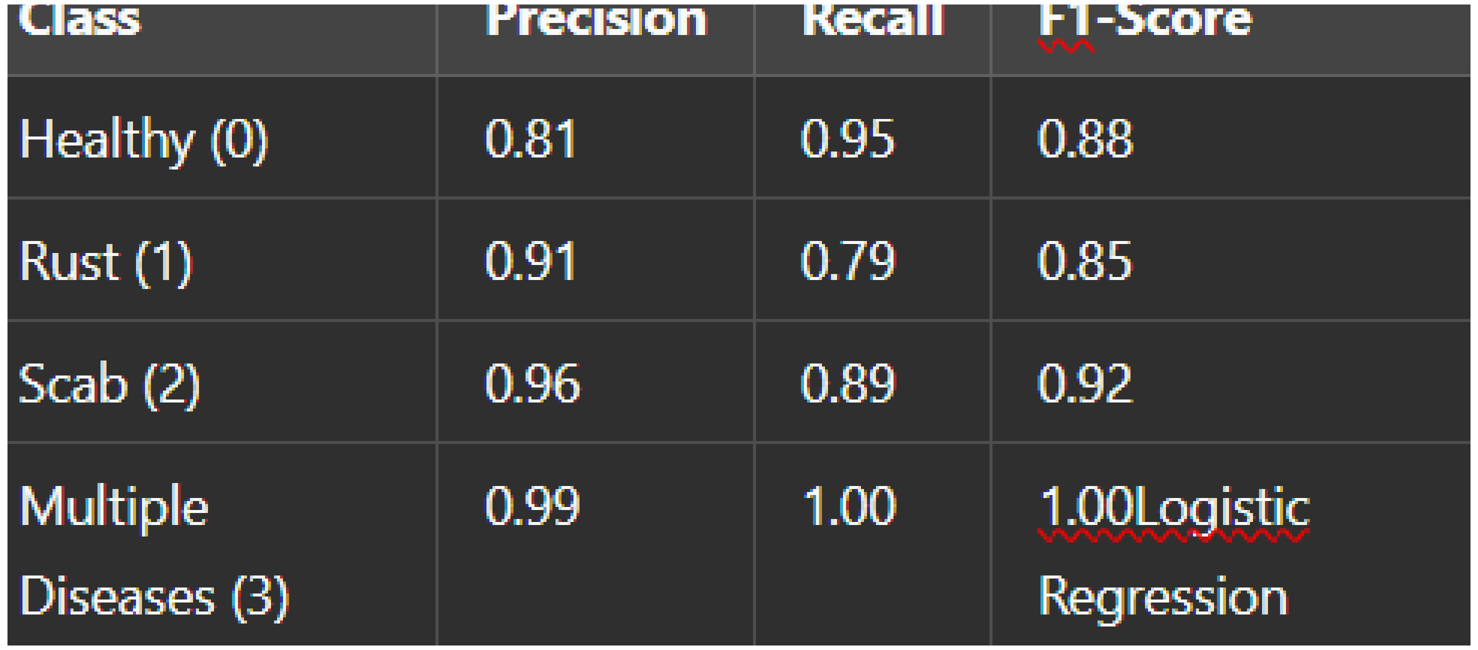
>Data Standardization

- The extracted features were standardized using StandardScaler to ensure uniform scaling before training.

2.3>Model Training and Evaluation

>Logistic Regression:

- *Results*:Accuracy: 91.16%Precision, Recall, and F1-Score:



Class	Precision	Recall	F1-Score
Healthy (0)	0.81	0.95	0.88
Rust (1)	0.91	0.79	0.85
Scab (2)	0.96	0.89	0.92
Multiple Diseases (3)	0.99	1.00	1.00Logistic Regression

Results:Accuracy: 91.16%
Precision, Recall, and F1-Score:

- *Overfitting Check*: The training accuracy was nearly 100%, indicating potential overfitting.
- *Confusion Matrix*:A detailed confusion matrix showed that most predictions were accurate across all classes, with minor misclassifications

>ROC-AUC Analysis:

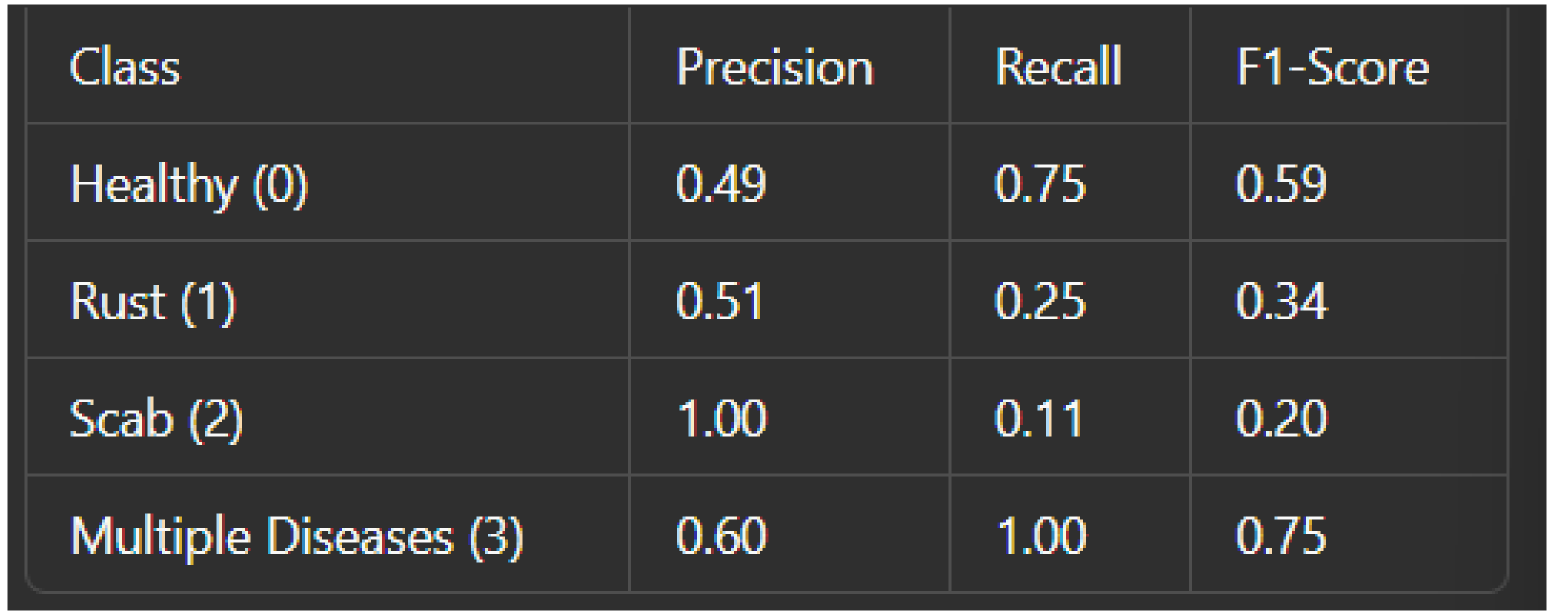
ROC curves were generated for all classes, showing excellent performance with high AUC scores for each class.

>K-Nearest Neighbors (KNN)

*Optimal K:*The elbow method was used to determine the best value for k. The Mean Squared Error (MSE) was minimized when k=2.

Results:

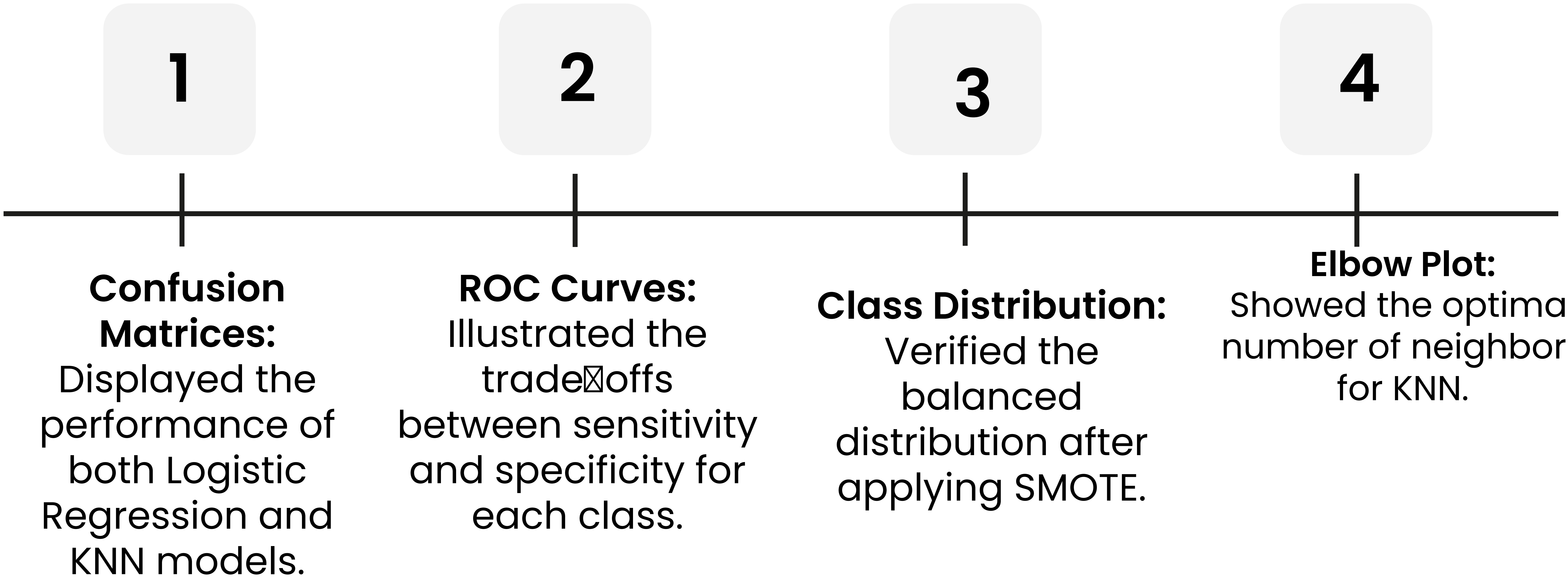
- Accuracy: 55.22%
- Precision, Recall, and F1-Score:



Class	Precision	Recall	F1-Score
Healthy (0)	0.49	0.75	0.59
Rust (1)	0.51	0.25	0.34
Scab (2)	1.00	0.11	0.20
Multiple Diseases (3)	0.60	1.00	0.75

Confusion Matrix:The confusion matrix showed significant misclassifications, particularly for the scab and rust classes.

2.4>Visualizations:



1

Confusion Matrices: Displayed the performance of both Logistic Regression and KNN models.

2

ROC Curves: Illustrated the trade‑offs between sensitivity and specificity for each class.

3

Class Distribution: Verified the balanced distribution after applying SMOTE.

4

Elbow Plot: Showed the optimal number of neighbors for KNN.

2.5 >Conclusion:

1. ConclusionBest Model: Logistic Regression achieved the best accuracy (91.16%) and balanced performance across all metrics.KNN Performance:
2. KNN underperformed due to the high dimensionality of features and the nature of the dataset.
3. Recommendations:

- For further improvement, explore other machine learning models (e.g., SVM, Random Forest) or fine-tune the VGG16 model.
- Implement dimensionality reduction techniques like PCA to address high feature dimensionality.