LOVELY PROFESSIONAL UNIVERSITY

Project Report Submission

Project title – **Apple Quality Finder Using Machine Learning Algorithms**

Course Code: INT 254

Course Title: **Fundamental of Machine Learning**

Submitted To

Rajan Kakkar

Id- 27659

Submitted by

Name- T K SANTHOSH RAO

Roll No- 17

Id- 12219295

Sec-KM067

DECLERATION

I hereby declare that the project work entitled “Apple Quality Check” is an authentic record of my own work carried out as requirements of Project for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Mr Rajan Kakkar, during February to March 2024. All the information furnished in this project report is based on my own intensive work and is genuine.

ACKNOWLEDGEMENT

It is with my immense gratitude that I acknowledge the support and help of my Professor, Mr Rajan Kakkar, who has always encouraged me into this research. Without his continuous guidance and persistent help, this project would not have been a success for me. I am grateful to the Lovely Professional University, Punjab and the department of Computer Science without which this project would have not been an achievement.

ABSTRACT

In this project we aim to perform a predictive task on a real-world problem using machine learning models. We proposed a baseline model base on classification task. We choose dataset from the Kaggle and train with various classification models. To evaluate the model, we test our model multiple algorithms Logistic Regression, K-Nearest Neighbors(KNN), Single Vector Mechanism (SVM), Random Forests. We performed hyperparameter tuning for Random Forests. The model gives the best accuracy 91% in Random Forests. The proposed model has potential in Apple's quality detection and classification.

TABLE OF CONTENTS

INTRODUCTION

This project is dedicated to tackling a classification task using machine learning models to address a specific real-world challenge. Classification tasks involve predicting the categorical outcome of data points, making them valuable in scenarios such as disease diagnosis, spam detection, or sentiment analysis. The chosen task provides a practical context for evaluating the predictive capabilities of different machine learning models.

For this project, we have opted to explore the performance of three distinct models:

1. Nearest Neighbour Baseline: As a starting point, we employ a nearest neighbour algorithm to establish a baseline for comparison. Two variations are evaluated, where the parameter k (number of neighbours) is set to 1 and 3. In order to pick 3 more classifiers, we will test some popular classifiers on the validation test and see which classifiers are the best for our dataset and pick best 2 classifiers.

2. Support Vector Classifier (SVC): SVC is a powerful model known for its effectiveness in handling complex decision boundaries. Its ability to work well in high-dimensional spaces and capture intricate patterns makes it an interesting candidate for our classification task.

3. Logistic Regression: Logistic regression predicts categorical outcomes by calculating probabilities based on input features, like age or income. It's used to determine the likelihood of events such as "yes" or "no" decisions, like whether someone will buy a product. By fitting a logistic curve to the data, it assigns probabilities ranging from 0 to 1, aiding in decision-making processes.

As our project predicts about good or bad quality of apple so we even also implemented logistic regression to know the accuracy of the model in this classifier .

4. Random forests: Random forests are a group of decision trees that work together to make predictions. They're robust, accurate, and versatile, handling both classification and regression tasks. Random forests provide insights into feature

importance and are less chances for overfitting. In essence, they combine the strength of multiple trees for improved predictive power in machine learning.

Evaluation Metrics:

Evaluation metrics are quantitative measures used to assess the performance and effectiveness of a statistical or machine learning model. These metrics provide insights into how well the model is performing and help in comparing different models or algorithms.

When evaluating a machine learning model, it is crucial to assess its predictive ability, generalization capability, and overall quality. Evaluation metrics provide objective criteria to measure these aspects. The choice of evaluation metrics depends on the specific problem domain, the type of data, and the desired outcome.

To gauge the performance of these models, we rely on a set of well-established evaluation metrics tailored for classification tasks:

A. **Accuracy:** This metric measures the overall correctness of predictions, providing a general overview of model performance.

B. **Confusion Matrix:** Is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.

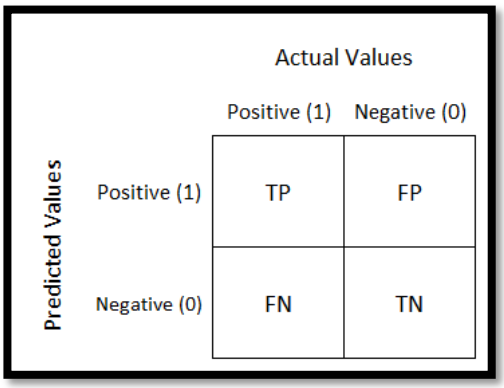


Figure 1: Confusion Matrix.

C. **Precision:** Precision quantifies the accuracy of positive predictions, indicatinghow well the model identifies relevant instances.

D. **Recall:** Recall, or sensitivity, evaluates the ability of the model to capture allrelevant instances within the dataset.

E. **F1 Score:** The F1 score, a harmonic mean of precision and recall, offers a balanced assessment of a model's performance.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + T_n}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Figure 2: Computing Evaluation Metrics.

These metrics collectively provide a comprehensive understanding of the models' predictive capabilities, allowing us to discern their strengths and weaknesses in the context of the chosen classification task.

Dataset

The dataset used in this project, titled "Apple Quality," is sourced from Kaggle. The dataset comprises various features associated with apples, offering valuable insights into factors influencing apple quality. By the way the data has been scaled and cleaned. Hereis a brief overview of the dataset:

Features

1. **Size:** The size of the apple, representing one of the physical dimensions.
2. **Weight:** The weight of the apple, measured in a suitable unit such as grams or ounces.
3. **Sweetness:** A quantitative measure indicating the degree of sweetness in the apple.

4. **Crunchiness:** A quantitative measure representing the texture and crunchiness of the apple.
5. **Juiciness:** A quantitative measure indicating the level of juiciness in the apple.
6. **Ripeness:** A quantitative measure representing the stage of ripeness of the apple.
7. **Acidity:** A quantitative measure representing the acidity level of the apple.
8. **Quality:** The target variable, indicating the overall quality of the apple. This is the label to be predicted and may be categorized into classes such as "good," and "bad."

Statistics

The dataset comprises a total of 4000 instances, each representing a unique apple.

- ***Quantitative Measures:***

Table 1: Dataset Quantitative Measures.

Name		Mean	Standard Deviation
Size		-0.503015	1.928059
Weight		-0.989547	1.602507
Sweetness		-0.470479	1.943441
Crunchiness		0.985478	1.402757
Juiciness		0.512118	1.930286
Ripeness		0.498277	1.874427
Acidity		0.076877	2.110270

- ***Categorical Measures:***
    - **Quality:** Value Counts: 'Good' ➔ 2004 || 'Bad' ➔ 1996.

Exploratory Data Analysis (EDA) & Preprocessing

Exploratory Data Analysis (EDA)(Part 1)

Exploratory Data Analysis will be conducted to gain deeper insights into the dataset's characteristics. Descriptive statistics will be calculated for numerical features, providing measures such as mean and standard deviation. Categorical features, such as color and texture, will be analyzed using frequency distributions and visualizations to discern patterns within the dataset. The EDA process aims to enhance our understanding of the dataset, identify potential challenges, and inform preprocessing steps as needed before applying machine learning models. This thorough examination of the dataset lays the groundwork for subsequent model evaluations and analyses.

We noticed that every attribute has 4000 values except for the last one which has 4001.

```
Size         4000 non-null
Weight       4000 non-null
Sweetness    4000 non-null
Crunchiness  4000 non-null
Juiciness    4000 non-null
Ripeness     4000 non-null
Acidity      4001 non-null
Quality      4000 non-null
```

NaN	NaN	NaN	NaN	NaN	NaN	Created_by_Nidula_Elgiriyewithana	NaN

We noticed only one row contains Null values, it turned out to be the last row, this row is for author rights, so we will drop it.

Preprocessing Steps(Part 1)

1. **Handling Missing Values:**

- Check for any missing values in the dataset through imputation or removal. In the Dataset only one row contains nan values so we will drop it.

```
A_id         4000 non-null
Size         4000 non-null
Weight       4000 non-null
Sweetness    4000 non-null
Crunchiness  4000 non-null
Juiciness    4000 non-null
Ripeness     4000 non-null
Acidity      4000 non-null
Quality      4000 non-null
```

2. **Encoding Categorical Labels:**

- Encode the **Quality labels** ('good' and 'bad') into numerical values, ensuring compatibility with machine learning algorithms. Below attached a reference picture of Label Encoding of quality Label.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Quality'] = le.fit_transform(df['Quality'])
# good =1
# bad = 0
```

Exploratory Data Analysis (EDA)(Part 2)

1. Plotting the Target variable unique classes

- We have plotted a pie chart on target variable for unique classes to check whether the data is balanced or not. As a result, we got to know that class distribution is balanced so we don't need to do any sampling, we will use the data as it is.

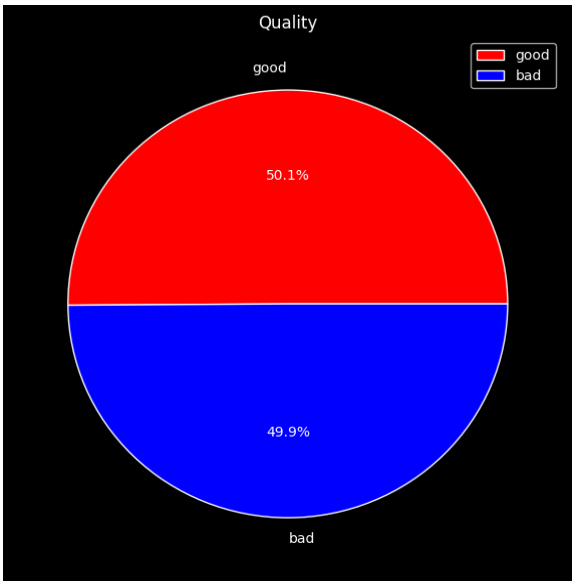


Figure 3: Pi plot of target variable unique classes

2. Correlation Analysis

Correlation analysis is a fundamental statistical technique employed to evaluate the strength and direction of the linear relationship between two quantitative variables within a dataset.

I. Calculate Correlation Coefficients:

- **Pearson Correlation Coefficient:** Measures the linear relationship between two variables. It ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no linear correlation. This coefficient is well-suited for linear relationships.

II.Visualize Correlations:

- Create a correlation matrix heatmap to visually represent the correlationcoefficients between pairs of quantitative features.
- Positive correlations are depicted in warmer colors (closer to 1), usually shades ofred.
- Negative correlations are depicted in cooler colors (closer to -1), typically shadesof blue.
- No correlation is represented in neutral colors, close to 0.

III.Interpretation:

- Analyse the correlation matrix to identify significant relationships betweenquantitative features.
- High positive correlations suggest that as one variable increases, the other tends toincrease.
- High negative correlations suggest that as one variable increases, the other tendsto decrease.
- Near-zero correlations indicate a lack of a linear relationship.

IV. Consider Outliers and Influential Points:

- When conducting correlation analysis, it's essential to be mindful of outliers or influential points within the dataset.
- Outliers are data points that significantly deviate from the rest of the data, potentially skewing the correlation coefficient.
- These outliers can inflate or deflate correlation values, leading to

misinterpretation of the relationship between variables.

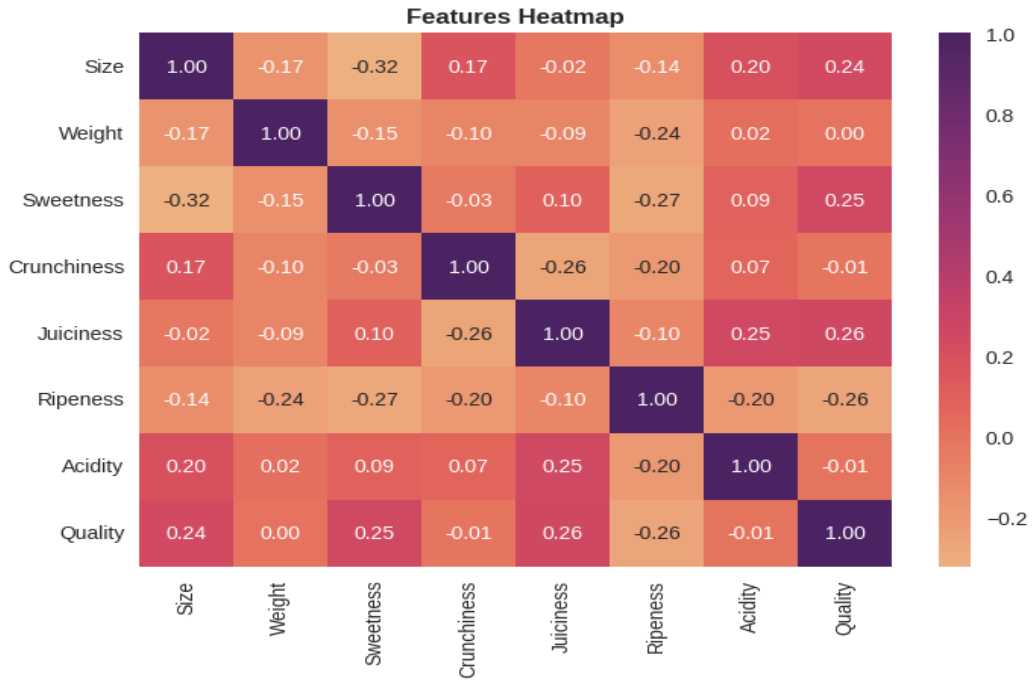


Figure 4: Correlation Matrix.

We can see that there are no strong correlations between the attributes.

3. Plotting Pair Plot

Pair plot is used for visualizing pairwise relationships between variables in a dataset. It creates a grid of scatterplots where each pair of variables in the dataset is represented by a scatterplot, allowing us to quickly visualize the relationships between them.

In Exploratory Data Analysis plotting a pairplot is also a very important step in the to explore the relationship between datapoints which make more sense to plot them with each and every feature in it .After plotting the pairplot we come to know that we got a pairplot which we have no relationship between the features. Relationships in the pairplot indicate 3 relations. Those are: -

Linear Relationships: If the points in the scatterplot form a roughly straight line, it indicates a linear relationship between the variables. This suggests that changes in one variable are associated with proportional changes in the other variable.

Non-linear Relationships: If the points in the scatterplot form a curved or non-linear pattern, it suggests a non-linear relationship between the variables.

No Relationship: If the points in the scatterplot are scattered randomly with no

discernible pattern, it suggests that there is no relationship between the variables.

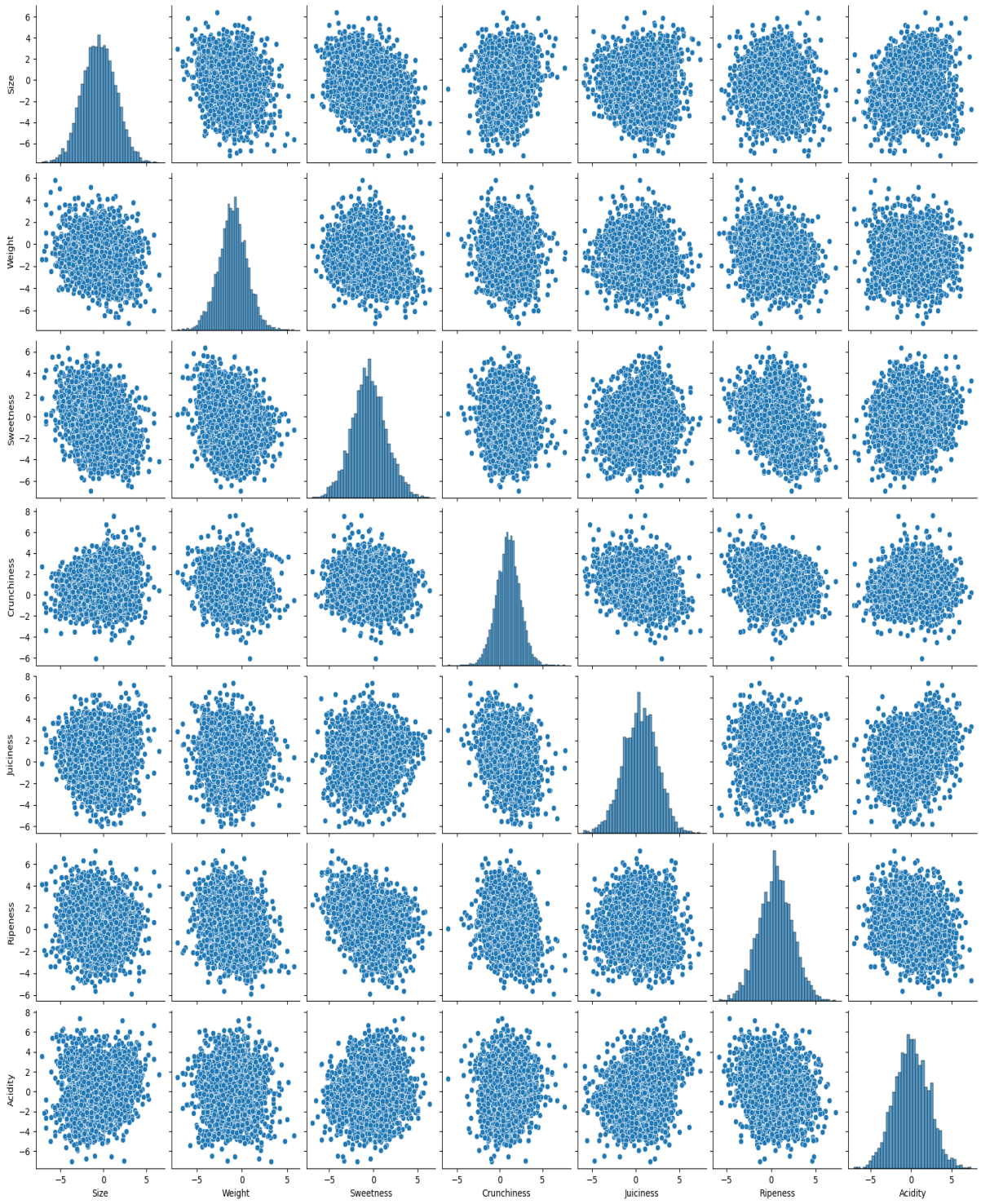


Figure 5: Pair plot

We can see that there are no strong correlations between the attributes.

Preprocessing Steps(Part 2)

1. Outlier Handling

As mentioned above the outlier checking is a crucial and expensive step in the Exploratory Data analysis so we also worked with our data to check with the outliers as a result we come to know that we have outliers present in our Data set which is also representing in the Below Figure

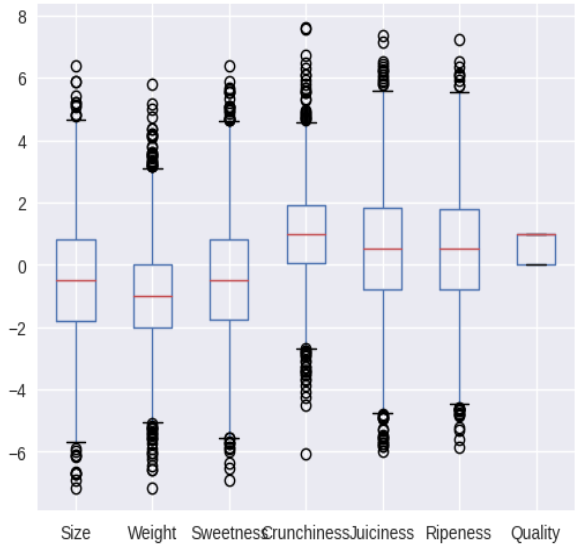


Figure 6: Outlier Detection Before Removal

We handled these outliers IQR capping concepts which makes the outliers free and not at all a complex model. It helps us in the increment in the accuracy score of the model and perfect fitting to the algorithms. IQR follow the formula which is given below

Interquartile Range(IQR) = Q3-QI
Lower bound= QI - 1.5(1QR)
Upper bound = Q3 + 1.5(IQR)

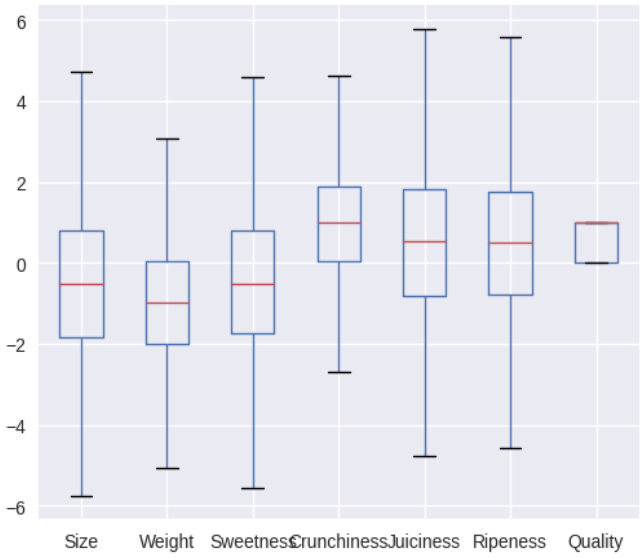


Figure 7: Outlier Detection After Removal

Splitting the data into train, validation, and test sets:

- Splitting the dataset into training, validation, and test sets is a crucial step to assess the performance of machine learning models effectively. This process ensures that models are trained on one subset of the data, validated on another subset to fine-tune parameters, and ultimately tested on a third independent subset to evaluate generalization performance.

```
Y train value counts:
Quality
0    1280
1    1280
Name: count, dtype: int64
----------------
Y validation value counts:
Quality
1    325
0    315
Name: count, dtype: int64
----------------
Y test value counts:
Quality
0    401
1    399
Name: count, dtype: int64
----------------
```

Figure 8: Splitting the data into train, validation, and test sets

Modelling and Evaluating

1. Logistic Regression

Logistic Regression, a classic classification algorithm, models the probability of binary outcomes with interpretable coefficients. It optimizes parameters like regularization strength (e.g., L1 or L2 penalty), convergence tolerance, and solver method (e.g., 'liblinear' for small datasets, 'sag' for large datasets). Renowned for its simplicity and interpretability, it's favoured for understanding feature importance. Cross-validation assesses its generalization performance reliably. Suited for linear decision boundaries, it handles numerical and categorical features effectively. It serves as a baseline model for complex algorithms, offering robust performance across diverse datasets.

Results

Accuracy: **0.876**

2. Baseline Model - Nearest Neighbor

As a starting point, a Nearest Neighbor algorithm was employed to establish a baseline for comparison. Two variations were evaluated: k=1 and k=3. The performance metrics, such as accuracy, precision, recall, and F1 score, were computed for each variation.

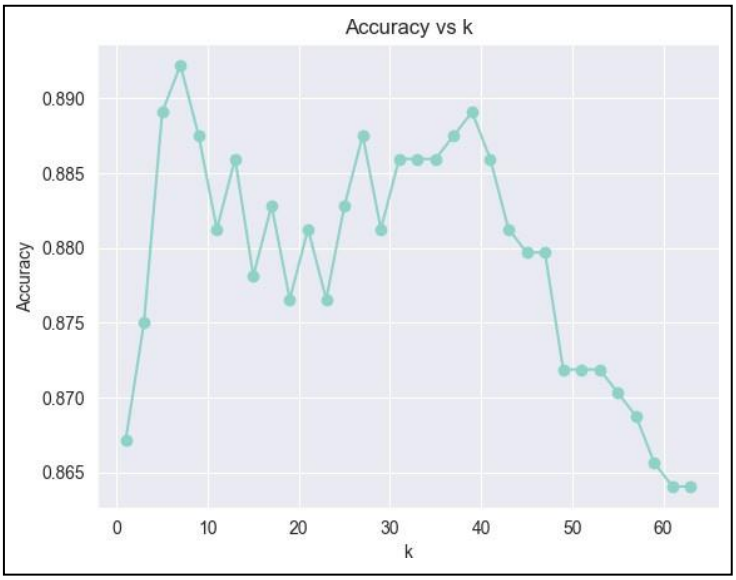


Figure 6: Baseline Model.

Results

- **K = 1:**

Confusion Matrix: [352 49]

[40 359]

Accuracy: *0.88875*

Classification Report:

Table 2: K=1 Classification Report.

Class	Precision	Recall	F1-score
0	0.90	0.88	0.89
1	0.88	0.90	0.89

- K = 3:

Confusion Matrix: [355 46]

[43 356]

Accuracy: *0.88875*

Classification Report:

Table 3: K=3 Classification Report.

Class	Precision	Recall	F1-score
0	0.89	0.89	0.89
1	0.89	0.89	0.89

We notice from Baseline Model that the best K Nearest Neighbor was 7.

- K = 7:

Confusion Matrix: [360 41]

[39 360]

Accuracy: *0.90*

Classification Report:

Table 4: K=7 Classification Report.

Class	Precision	Recall	F1-score
0	0.90	0.90	0.90
1	0.90	0.90	0.90

3. Support Vector Classifier (SVC)

The Support Vector Classifier (SVC) was chosen for its effectiveness in handling complex decision boundaries. Various hyper-parameters, such as C and kernel type, wereexplored through grid search. The model was evaluated using cross-validation.

- **Hyper-parameter Selection**
  - Grid search explored C values and different kernel types.
  - After testing different values for C, we found that **C=100** is the best value.

- **Results**

Confusion Matrix: [371 30]
[37 362]

Accuracy: **0.91625**

Classification Report:

Table 5: SVC Classification Report.

Class	Precision	Recall	F1-score
0	0.91	0.93	0.92
1	0.92	0.91	0.92

4. Random Forests

Random Forest, an ensemble learning method, constructs multiple decision trees using bootstrap sampling and random feature selection. It aggregates predictions through voting (classification) or averaging (regression). Hyperparameters like the number of trees, maximum depth, and minimum samples for splitting are crucial for tuning. Cross-validation is commonly employed to assess model performance. Renowned for handling complex data with robustness, Random Forest's effectiveness lies in its ability to balance variance and bias.

- **Hyper-parameter Selection**
    - Grid search explored n_estimators, max_depth, min_samples_split, min_samples_leaf values and different validations.
    - After testing different values for n_estimators, max_depth, min_samples_split, min_samples_leaf values, we found that **'max_depth': 20, 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 150** is the best value.

- **Results**

Confusion Matrix: [365 37]
[39 343]

Accuracy: **0.885**

Classification Report:

Table 6: Random Forest Classification Report.

Class	Precision	Recall	F1-score
0	0.87	0.85	0.89
1	0.88	0.90	0.82

Hyperparameter Tuning

Hyperparameters are parameters that are set before the learning process begins and cannot be learned from the data.

Grid Search:

GridSearchCV is a technique used for hyperparameter tuning in machine learning. It exhaustively searches through a specified grid of hyperparameters for the best combination that maximizes model performance.

Cross-Validation:

GridSearchCV uses cross-validation to evaluate the performance of each hyperparameter combination. By default, it is k-fold cross-validation, where the dataset is split into k smaller parts (folds), and the model is trained k times on different pairs of training and validation sets.

In the project we used the normal C value as the 10 in some cases and C value as 5 in other some cases.

Best Estimator and Best Parameters: After the hyperparameter search is complete, GridSearchCV identifies the best estimator (model) and the corresponding best hyperparameters based on the chosen scoring function. You can access the best estimator using the .best_estimator_ attribute and the best parameters using the .best_params_ attribute. Below is picture attached that indicates the parameters tuning on the random forest Algorithm in the model...

```
from sklearn.model_selection import GridSearchCV
param_grid = {
    'n_estimators': [50, 100, 150],  # Number of trees in the forest
    'max_depth': [6, 8,10,20],  # Maximum depth of individual trees
    'min_samples_split': [4, 6, 8],  # Minimum samples required to
split a node
    'min_samples_leaf': [2, 4],   # Minimum samples required at each
leaf node
}

rfc = RandomForestClassifier()
```

```
# Create a GridSearchCV
object                                                    ##Hyper
Parameter tuning of Random Forest
grid_search = GridSearchCV(estimator=rfc, param_grid=param_grid, cv=10,
scoring='accuracy')

grid_search.fit(x_train, y_train)
print("Best Parameters:", grid_search.best_params_)
print("Best Score:", grid_search.best_score_)
```

Analysis

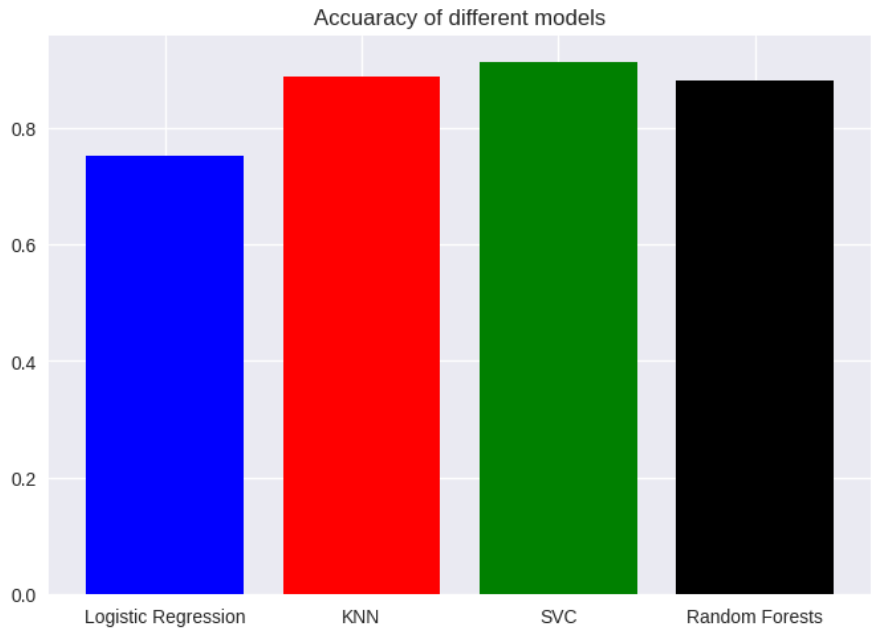
I. Model Selection

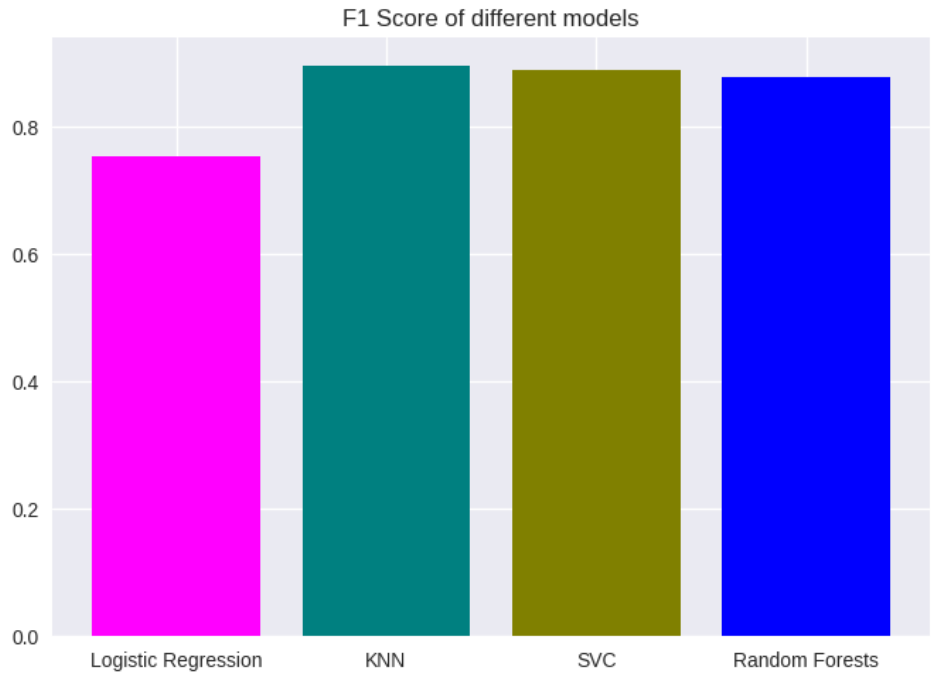
Support Vector Classifier (SVC) was identified as the best model based on a comprehensive evaluation using metrics such as accuracy, precision, recall, and F1 score.

II. Performance Metrics

- The **accuracy** of the SVC on the test set was **0.91625**, highlighting its overall correctness in predicting Apple classifier.
- **Precision**, indicating the model's ability to correctly identify 'Good' quality apple, achieved **0.92**.
- **Recall**, representing the model's ability to capture all 'Good' quality apple, reached **0.91**.
- The **F1 score**, balancing precision, and recall, demonstrated a robust **0.92**.

Plotting the Bar graphs of Accuracy and F1 Scores of the all Models :-





Conclusion and Future Scope

The project aimed to evaluate the performance of Four models—Logistic Regression, Nearest Neighbor BaselineL, Support Vector Classifier (SVC), and Random Forests for apple quality classification. While the initial baseline model exhibited lower accuracy, hyperparameter tuning demonstrably improved its performance.

The models exhibited varying degrees of success, with SVC emerging as the best performer based on comprehensive evaluation metrics.

Logistic Regression : Logistic Regression offered a straightforward approach to classification, its accuracy of 78% . Logistic Regression's strength lies in its simplicity.

Nearest Neighbor Baseline(KNN): The baseline models, particularly k=7 in Nearest Neighbor, provided a solid starting point. However, more sophisticated models surpassed their performance. KNN achieved 90% accuracy with the k value 7.

Support Vector Classifier (SVC): SVC demonstrated superior performance with an accuracy of 91.625%. Its precision, recall, and F1 score reflected a balanced and robust classification ability.

Random Forests: While Random Forests Classifier performed well, achieving an accuracy of 88.5%, it fell slightly short of SVC in terms of overall metrics.

Evaluation Metrics:

Strengths:

The chosen metrics, including accuracy , precision, recall, F1 score, offered a comprehensive evaluation of model performance.

Limitations:

The binary classification approach might not fully capture the nuances of apple quality. Expanding to a multi-class classification could provide more detailed assessments.

The evaluation metrics are based on a specific threshold; sensitivity analysis for threshold selection could further enhance understanding.

Future Scope

- **Advanced Machine Learning Techniques**: Further developments of machine learning algorithms, commonly known as deep learning, reinforcement learning, and ensemble methods, might be used to increase the accuracy and efficiency of grid stability prognostic models. They are designed to manage high-dimensional, complex data and, thus, recognize and perfectly catch nonlinear relationships than traditional approaches .

- **Integration of Big Data and IoT**: Big data analytics and the Internet of Things may be discovered and used to gather and analyze huge amounts of real-time data from the various Datasets, smart Scanners, and other sources placed in the grid composition. Machine learning-based models can then cultivate this data to identify regularities, problems, and prognostic insights to boost grid stability.

- **App based software working model:-** After applying all the advanced models we can also implement this project with real time public with the help of an app to scan fruits,or vegetable or any other quality finder products to get quality very instantly to the public and very accurately.

- **Data Monitoring:** Create a system to monitor performance standards over time and check for possible deteriorations. This will involve collecting new data and periodically re-evaluating the accuracy of the model.

GOOLE COLAB LINK: -

Below is Link of our project in google colab

https://colab.research.google.com/drive/158SwiVnMB7s_Pzb07twAGdqVKHZe3fai?usp=sharing

References:

- Xiaobo, Zou, and Zhao Jiewen. "Apple quality assessment by fusion three sensors." In *SENSORS, 2005 IEEE*, pp. 4-pp. IEEE, 2005.

- Rehkugler, G. E., and J. A. Throop. "Apple sorting with machine vision." *Transactions of the ASAE* 29, no. 5 (1986): 1388-1397.

- Singh, Swati, Isha Gupta, Sheifali Gupta, Deepika Koundal, Sultan Aljahdali, Shubham Mahajan, and Amit Kant Pandit. "Deep learning based automated detection of diseases from Apple leaf images." *Computers, Materials & Continua* 71, no. 1 (2022).

- Çetin, Necati, Kevser Karaman, Erhan Kavuncuoğlu, Bekir Yıldırım, and Ahmad Jahanbakhshi. "Using hyperspectral imaging technology and machine learning algorithms for assessing internal quality parameters of apple fruits." *Chemometrics and Intelligent Laboratory Systems* 230 (2022): 104650.

- Zou, Xiuguo, Chenyang Wang, Manman Luo, Qiaomu Ren, Yingying Liu, Shikai Zhang, Yungang Bai, Jiawei Meng, Wentian Zhang, and Steven W. Su. "Design of electronic nose detection system for apple quality grading based on computational fluid dynamics simulation and k-nearest neighbor support vector machine." *Sensors* 22, no. 8 (2022): 2997.

- Shurygin, Boris, Igor Smirnov, Andrey Chilikin, Dmitry Khort, Alexey Kutyrev, Svetlana Zhukovskaya, and Alexei Solovchenko. "Mutual augmentation of spectral sensing and machine learning for non-invasive detection of apple fruit damages." *Horticulturae* 8, no. 12 (2022): 1111.