DATA, INFERENCE

&

APPLIED MACHINE LEARNING

(COURSE 18-785)

ASSIGNMENT 6

Mark Iraguha

(miraguha)

Table of Contents

Libraries Used:

Matplotlib – a python plotting library used to create animated, interactive and static visualizations.[1]
Pandas – another Python library used that provides data structures and functions used to carry out data analysis.[2]
Numpy – a simple yet powerful data structure provided in python.[3]
Tabulate – a python library that tabulates data to an output[4].
Statsmodel – a python library that provides a wide range of statistical models and tools for analyzing data[5].
Scikit-Learn - a free machine learning library for Python that supports both supervised and unsupervised machine learning, providing diverse algorithms for classification, regression, clustering, and dimensionality reduction[6].

Introduction:

This report details the completion of Assignment 6. Assignment 6 requests answers to 4 critical thinking and data analytical questions.

Objectives include:

- Understanding and dealing with nonlinearity.
- Fitting classification models.
- Choosing optimal model parameters.
- Performing cross-validation.
- Evaluating and comparing model performance.

Question 1 Report:

1.1 Nonlinearity explained and its necessity in considering nonlinear relationships between variables

Nonlinearity describes a relationship between two variables where changes in one variable do not result in proportional changes in another. This means that the relationship cannot be represented by a straight line on a graph; instead, it typically forms a curve or more complex shape. Nonlinear relationships often involve intricate interactions and dependencies that simple linear equations cannot adequately capture.[7]

Reasons for considering nonlinear relationships between variables:

- Nonlinear models can capture complex patterns, dependencies and interactions that linear models might miss.
- Linear models can oversimplify complex interactions, leading to incorrect conclusions and predictions. This is avoided with nonlinear models.
- Knowledge of nonlinear relationships between models is key in constructing more accurate and realistic predictive models.
- Understanding nonlinearity is crucial for developing more realistic predictive models that reflect actual dynamics[7].

1.2 Nonlinear Model Equation and Example:

Mathematical Equation for a Nonlinear Model:

A general nonlinear model can be expressed as:

$$y = f(x1, x2, \ldots, xn)$$

In this equation, y is the dependent variable, $x1,x2,\ldots,xn$ are the independent variables, and ff represents a nonlinear function that describes their relationship.

Example of a nonlinear model application:

One common example of a nonlinear model is Logistic Regression, which is used in classification problems. The relationship between input variables and the probability of a certain outcome is nonlinear. The equation for logistic regression can be represented as:

$$P(y = 1) = \frac{1}{1 + e - z1}$$

where $z=\beta0+\beta1x1+...+\beta nxn$. This model helps predict binary outcomes based on various predictors

1.3 Parsimonious Nonlinear Models:

Parsimony in Linear vs. Nonlinear Models:

- A model is called parsimonious if it provides a good fit to the data with fewer parameters.
- Linear models are often viewed as more parsimonious due to their simplicity; nonlinear models can also be designed to be efficient.

Mathematical comparison:

- Linear Model:

$$y=\beta0+\beta1x1+...+\beta nxn$$

- Nonlinear Model:

$$y=f(x1,x2,...,xn)$$

- In some cases, a carefully chosen nonlinear function ff may require fewer parameters than a linear model while still effectively capturing the underlying relationships[8].

1.4 Surrogate Data for Nonlinearity Testing:

Characteristics preserved in surrogates:

- Surrogate data generation aims to maintain specific characteristics of original data while eliminating nonlinear relationships. Typically, this involves preserving marginal distributions and linear correlations among variables.

Surrogate techniques:

- Fourier Transform Surrogates: This method transforms data into the frequency domain, randomizes the phases, and then transforms it back to the time domain.
- Amplitude-Adjusted Fourier Transform (AAFT) Surrogates: Similar to Fourier Transform surrogates but adjusts amplitudes to align with the original data's distribution[9].

1.5 Information Theory Concepts and Applications:

Definitions:

- Information: Measure of how much uncertainty is reduced when observing an event.
- Entropy: Measure of uncertainty or randomness in a system.
- Mutual Information: Measures how much knowing one variable reduces uncertainty about another variable.

Mathematical formulas:

- Entropy (H):

$$H(X) = -i = 1\sum np(xi)log2(p(xi))$$

- Mutual Information (I):

$$I(X,Y) = x \in X \sum y \in Y \sum p(x,y) log2(p(x)p(y)p(x,y))$$

Entropy for measuring regularity:

- Entropy can quantify the complexity or regularity of time series data. For instance, approximate entropy (ApEn) assesses how similar patterns in observations remain over time. This is particularly useful in analyzing physiological signals like heart rate variability[10].

Mutual Information for feature selection:

- Mutual information helps identify informative features for predicting target variables. Unlike correlation, mutual information captures both linear and nonlinear dependencies, providing a more comprehensive understanding of relationships between variables[11].

Question 2 Report:

2.1 Decision trees

Components of a Decision Tree:

Nodes

A decision tree is made up of **nodes**, which are categorized into two types:

- **Decision Nodes**: These nodes represent the features or attributes used to split the data.
- **Leaf Nodes**: These nodes signify the final outcomes or decisions[12].

Branches

Branches are the connections between the nodes. They illustrate the possible values or conditions of the attributes, leading to subsequent nodes based on the attribute values[12].

Pruning:

- Pruning is the technique of removing unnecessary nodes and branches from a decision tree. This process simplifies the model and helps prevent overfitting, which occurs when a tree becomes too complex[13].

Necessity of Pruning:

- Complex trees may perfectly fit training data but fail to generalize well to new, unseen data. Pruning reduces this complexity, enhancing the tree's performance on fresh data.

Advantages of Decision Trees:

- **Interpretability**: Decision trees are straightforward and easy to understand, making them ideal for applications where clarity is essential[12].

- **Non-parametric**: They do not assume any specific distribution for the data, allowing them to work with various types of datasets[13].
- **Handling Nonlinear Relationships**: Decision trees can effectively capture complex nonlinear relationships between variables, making them versatile for classification tasks.
- **Feature Importance**: They provide insights into which features are most significant, aiding in feature selection and data comprehension.

2.2 Steps to improve the existing rule-based classifier:

To improve upon an existing rule-based classifier, the following steps can be taken to develop a more robust data-driven classifier:

Data Collection and Preprocessing:

- **Data Collection:** Ensure the availability of a large and diverse dataset relevant to the classification problem.
- **Data Preprocessing:** Clean and preprocess the data to handle missing values, outliers, and inconsistencies. Encode categorical variables to make them suitable for the chosen classification algorithm.

Feature Engineering:

- Analyze the existing rules and domain knowledge to identify additional relevant features. Feature engineering involves creating new features or transforming existing ones to capture meaningful patterns in the data, potentially improving the model's performance.

Model Selection and Training:

- **Model Selection:** Choose an appropriate data-driven classification algorithm, such as Decision Trees, Random Forests, or Gradient Boosting, based on the problem's nature and complexity.

- **Model Training:** Split the data into training and validation sets. Train the selected model on the training set, optimizing hyperparameters using techniques like cross-validation to find the best model configuration.

Model Evaluation:

- Evaluate the new model's performance on the validation set using relevant evaluation metrics such as accuracy, precision, recall, and F1-score. Compare these results with the existing rule-based classifier to assess the improvements.

Testing the New Model's Validity:

- **Hold-out Testing:** Split the data into training, validation, and testing sets. Train and optimize the model on the training set, validate on the validation set, and finally assess its performance on the unseen testing set.
- **Cross-Validation:** Employ k-fold cross-validation to evaluate the model's performance across different subsets of the data, ensuring a more comprehensive assessment.
- **Statistical Significance Testing:** Conduct hypothesis tests (e.g., t-test, chi-square test) to compare the new model's performance with the existing one, ensuring that the improvements are statistically significant.

2.3 Decision Tree for Titanic dataset:

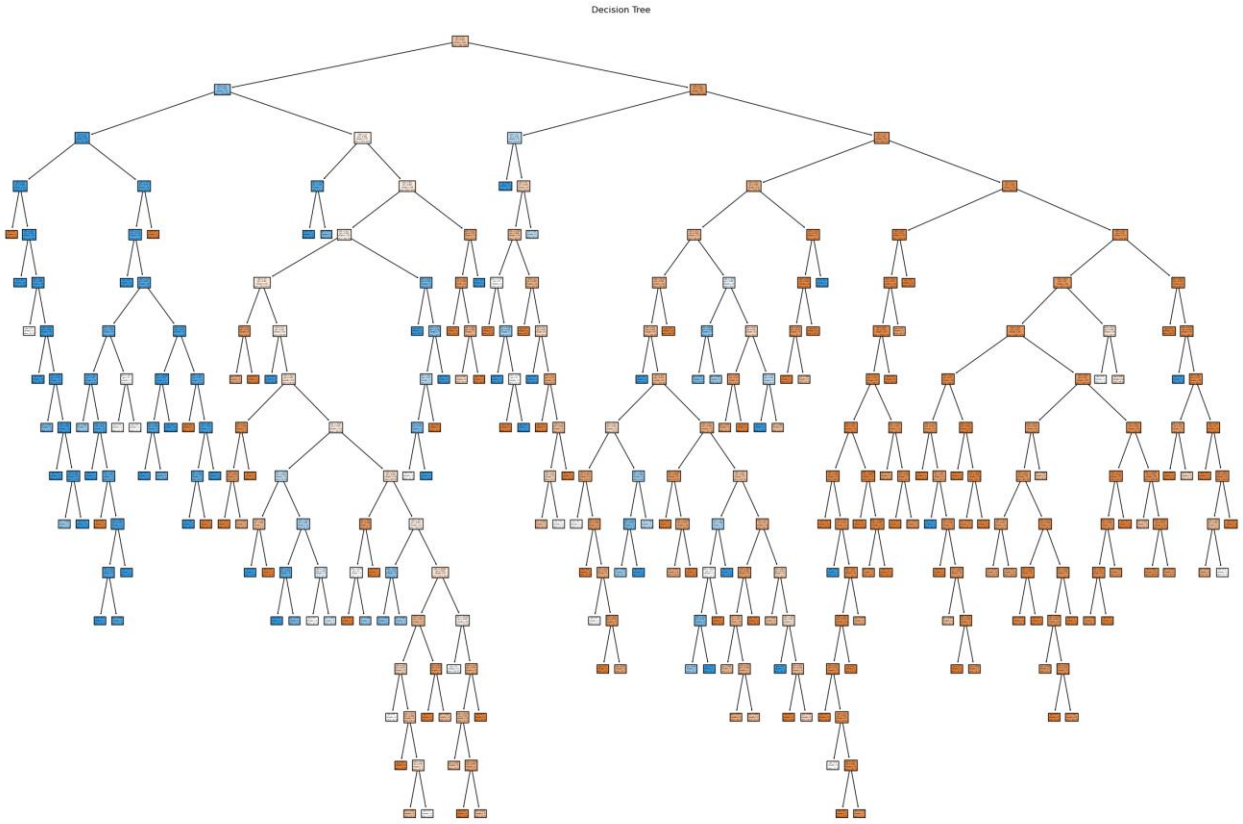
Steps followed:

- Used the Titanic dataset with features like passenger class, age, sex, and survived.
- Split the data into training and testing sets.

- Built a decision tree classifier using DecisionTreeClassifier[14] library.
- Trained the decision tree on the training data.
- Visualized the tree structure using python Tree function.

Decision Tree Structure:

The decision tree for Titanic survival prediction is presented below:



2.4 Tree Performance Evaluation (before and after pruning):

Before Pruning:

Training and Testing Sets: The decision tree was trained on the Titanic dataset and evaluated using cross-validation.

Evaluation Metrics: The model's performance was assessed using accuracy, precision, recall, and F1-score.

Results: The decision tree achieved an accuracy of 0.74 on the test set, with a misclassification error rate of 0.26. The confusion matrix[15] provided insights into the model's performance for different classes.

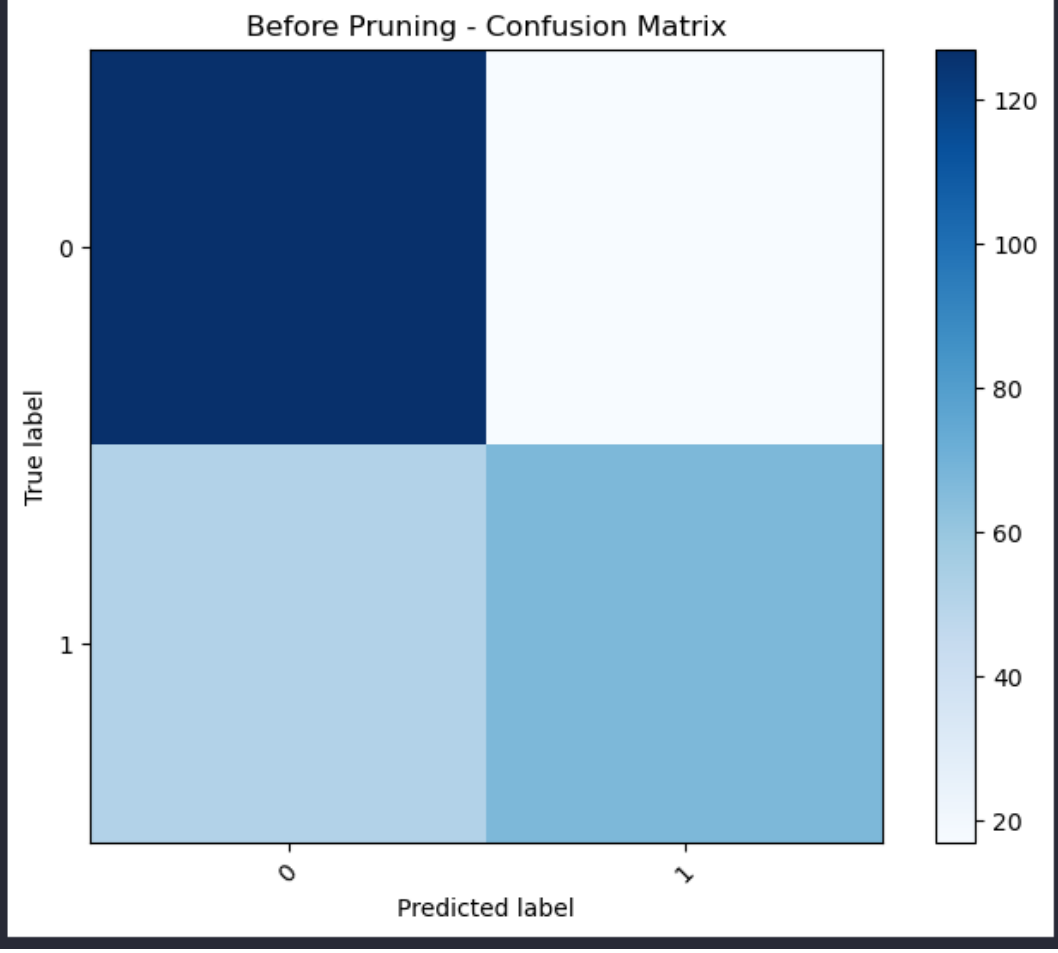
Results:

```
Before Pruning - Accuracy: 0.74
Before Pruning - Misclassification Error Rate:0.26
Before Pruning - Classification Report:
              precision    recall  f1-score   support

           0       0.71      0.88      0.79       144
           1       0.80      0.57      0.66       118

    accuracy                           0.74       262
   macro avg       0.76      0.72      0.73       262
weighted avg       0.75      0.74      0.73       262
```

Confusion Matrix

Actual/Predicted	Survived	Not Survived
Survived	127	17
Not Survived	51	67

```
Before Pruning - Confusion Matrix:

[[127  17]
 [ 51  67]]
```



Pruning:

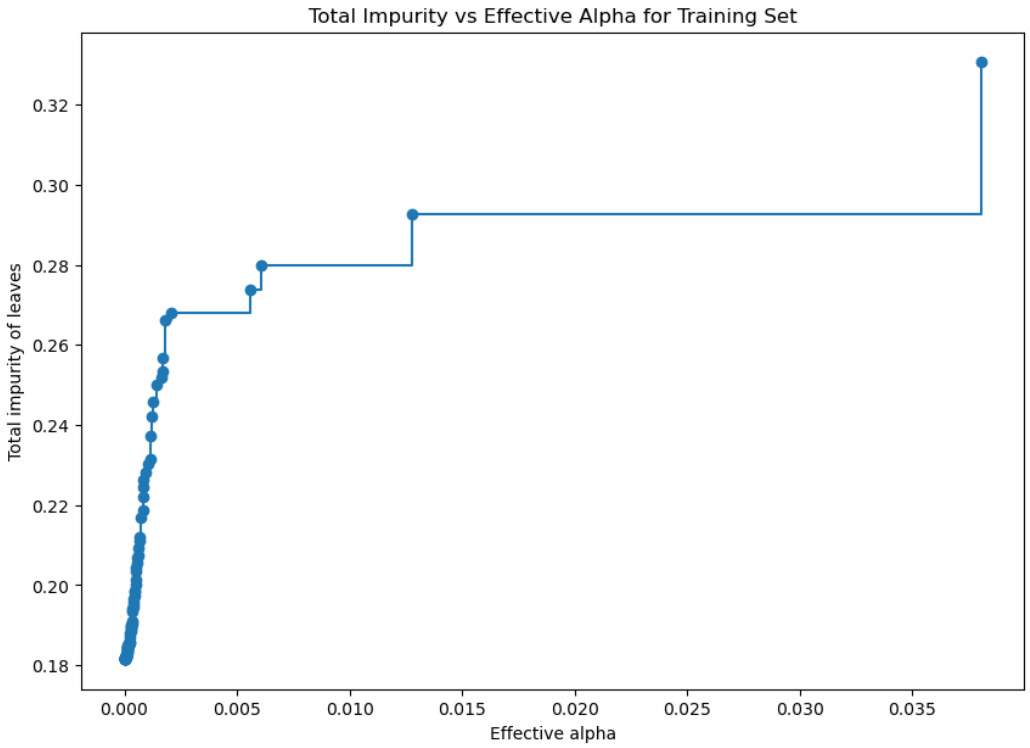
Pruning Technique: The decision tree was pruned using the Cost Complexity Pruning (CCP) algorithm, which optimizes the tree's complexity by minimizing the cost complexity measure.

Parameter Tuning: Various values of the CCP parameter (ccp_alpha)[16] were tested to find the optimal balance between model complexity and performance.

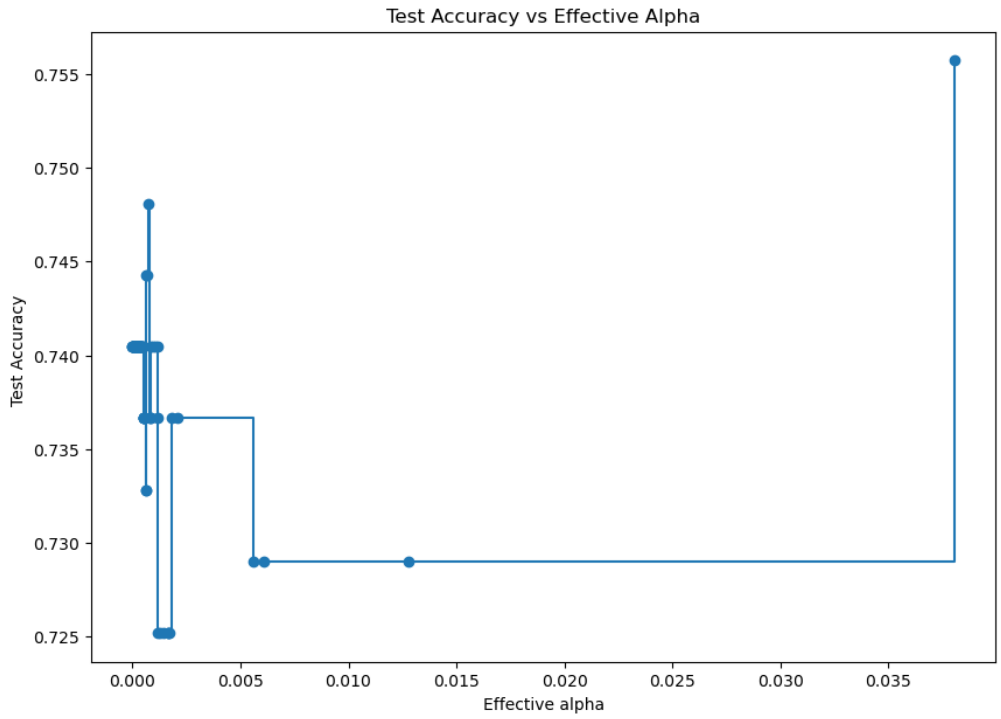
Best Pruning Parameter: The optimal ccp_alpha value was determined as 0.0380 with an accuracy of 0.76, resulting in the best-pruned tree.

```
Best ccp_alpha: 0.038055312783106645, with accuracy: 0.76
```

Graph showing Total Impurities movement with different ccp_alpha values



Graph showing accuracies for each pruned tree on the test set



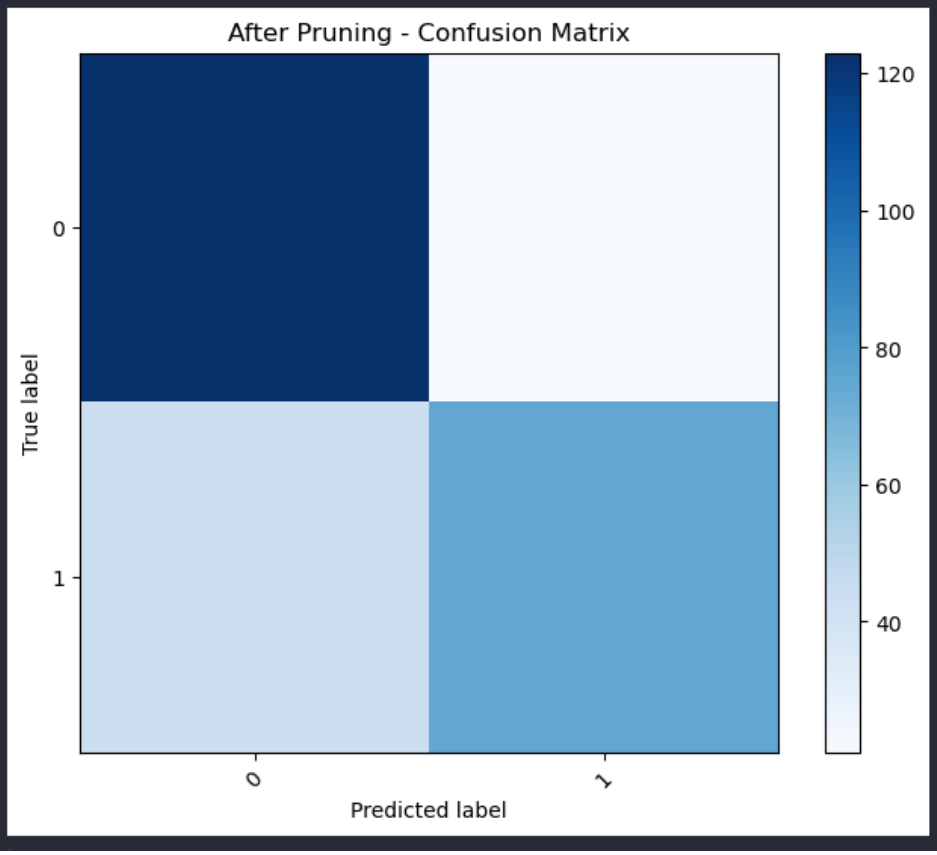
After Pruning:

Evaluation: The pruned tree's performance was evaluated using cross-validation on the same training and testing sets.

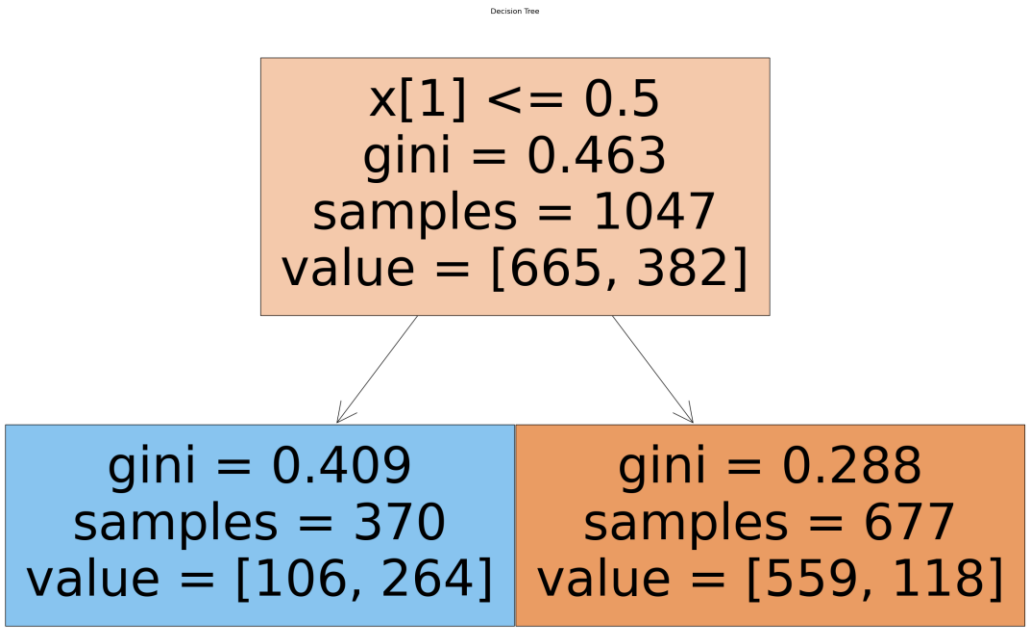
Results: The pruned tree achieved a cross-validation accuracy of 0.78, indicating an improvement over the unpruned tree.

```
After Pruning - Test Accuracy: 0.76
After Pruning - Misclassification Error Rate:0.24
After Pruning - Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.85      0.79       144
           1       0.78      0.64      0.70       118

    accuracy                           0.76       262
   macro avg       0.76      0.74      0.75       262
weighted avg       0.76      0.76      0.75       262
```

```
After Pruning - Confusion Matrix:

[[123  21]
 [ 43  75]]
```



Pruned tree:



2.5 Comparison Pruned Tree and Logistic Regression:

Logistic Regression Results:

Cross-Validation Accuracy: Logistic regression achieved a cross-validation accuracy of 0.80, with a standard deviation of 0.02.

Test Accuracy: On the test set, logistic regression obtained an accuracy of 0.77, with a misclassification error rate of 0.23.

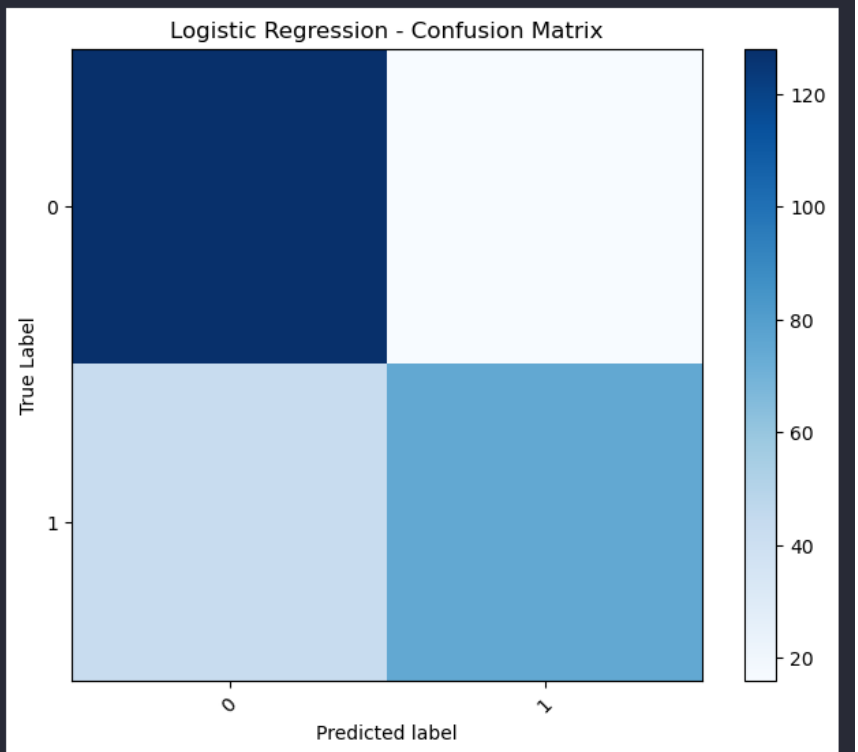
Confusion Matrix: The confusion matrix for logistic regression shows its performance across different classes.

Confusion Matrix:

Actual/Predicted	Survived	Not Survived
Survived	128	16
Not Survived	43	75

```
Logistic regression cross-validation accuracy: 0.80 ∓ 0.02
Logistic Regression - Test Accuracy: 0.77
Misclassification Error Rate: 0.23
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.89      0.81       144
           1       0.82      0.64      0.72       118

    accuracy                           0.77       262
   macro avg       0.79      0.76      0.77       262
weighted avg       0.78      0.77      0.77       262
```

```
Confusion Matrix:

[[128  16]
 [ 43  75]]
```



Decision Tree Classifier Model Results:

Cross-Validation Accuracy: The decision tree achieved a cross-validation accuracy of 0.78, with a standard deviation of 0.02.

Test Accuracy: On the test set, the decision tree obtained an accuracy of 0.76, with a misclassification error rate of 0.24.

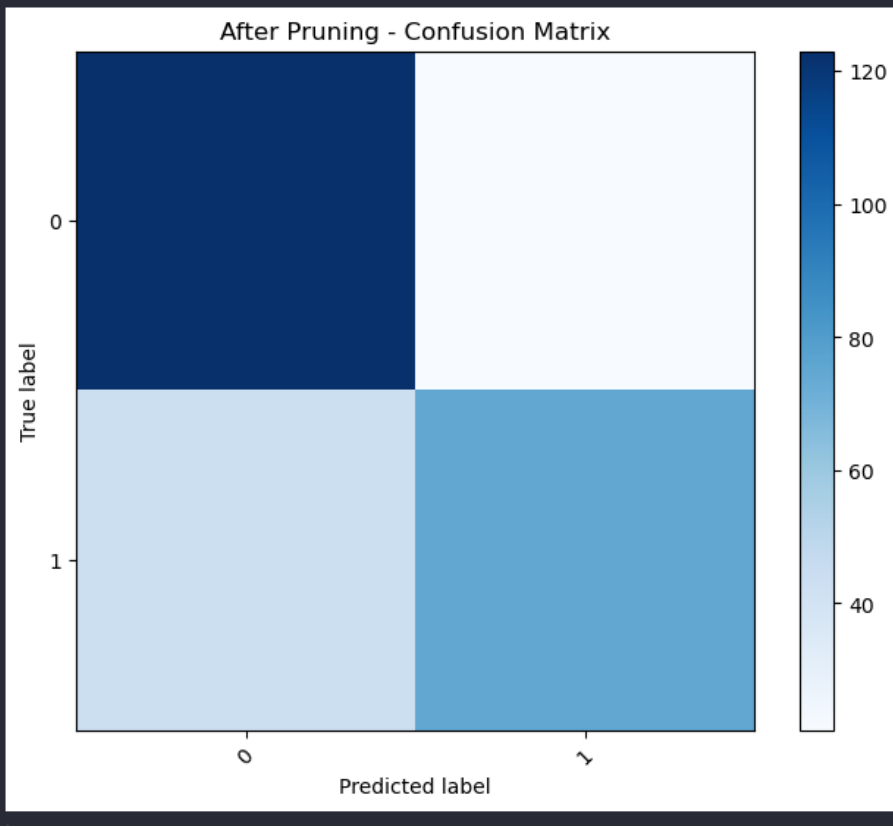
Confusion Matrix: The confusion matrix for the decision tree provides insights into its performance.

Confusion Matrix:

Actual/Predicted	Survived	Not Survived
Survived	123	21
Not Survived	43	75

```
Pruned Decision Tree Cross-Validation Accuracy: 0.78 ± 0.02
After Pruning - Test Accuracy: 0.76
After Pruning - Misclassification Error Rate:0.24
After Pruning - Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.85      0.79       144
           1       0.78      0.64      0.70       118

    accuracy                           0.76       262
   macro avg       0.76      0.74      0.75       262
weighted avg       0.76      0.76      0.75       262
```

```
After Pruning - Confusion Matrix:

[[123  21]
 [ 43  75]]
```



Evaluation:

Advantages and Disadvantages:

- Decision Tree:
  - Pros: Interpretability handles nonlinear relationships, feature importance.
  - Cons: May overfit, sensitive to small changes in data, can be complex to interpret for large trees.

- Logistic Regression:
    - Pros: Simple, well-understood, provides probability estimates.
    - Cons: Assumes linear relationships, may not capture complex patterns.

Kaggle Competition Choice:

- For the Kaggle Titanic competition, both models can be effective. However, the decision tree's ability to handle nonlinear relationships and provide feature importance insights gives it an edge.

2.6 Analysis and Insights

The analysis of the decision tree's performance on the Titanic dataset reveals several key insights:

Feature Importance: Gender and age are critical factors in predicting survival, with female passengers, especially younger ones, having a higher chance of survival.

Nonlinear Relationships: The decision tree effectively captures nonlinear relationships, as seen in the age and class interactions. This capability is a significant advantage over linear models like logistic regression.

Pruning Impact: Pruning the decision tree improved its performance, as evidenced by the accuracy score increasing from 0.74 to 0.76. This improvement demonstrates the importance of finding the right balance between model complexity and generalization.

Model Comparison: The decision tree and logistic regression models have comparable performance, with logistic regression slightly outperforming the decision tree in accuracy. However, the decision tree's interpretability and feature importance insights make it a valuable tool for understanding the data and making informed decisions.

2.7 Conclusion

Decision trees offer a powerful and interpretable approach to classification. The decision tree constructed in this analysis effectively captures the complex relationships between passenger attributes and survival outcomes. Pruning techniques improve the tree's performance and generalization ability, demonstrating the importance of model optimization.

Question 3 Report:

3.1 Parsimonious Models using Local Modeling

Concept:

Local modeling is a powerful technique that aims to construct models for small, localized regions of the state space rather than relying on a single global model for the entire dataset. This approach is based on the principle that local patterns and relationships within specific neighborhoods may be more accurately represented by simpler models, leading to increased parsimony and improved predictive performance[17].

Step-by-step procedure:

a. Divide the State Space: Partition the input space into smaller regions or neighborhoods based on the input variables. This can be done using techniques like k-means clustering.
b. Local Model Construction: For each neighborhood, construct a separate model using the data points within that region. This can be a simple linear model or a more complex nonlinear model, depending on the problem.
c. Model Complexity: Keep the models parsimonious by limiting the number of parameters or features used in each local model. This helps to avoid overfitting and improves interpretability.
d. Prediction: When a new data point is encountered, its neighborhood is determined based on its input values. The corresponding local model is then used to make predictions for that specific data point. This localized prediction approach allows the model to adapt to the local characteristics of the data.
e. Ensemble Prediction: In some cases, combining predictions from multiple local models can improve overall performance. Techniques like weighted averaging or voting can be employed to aggregate the predictions from different local models, resulting in a more robust final prediction[18].

3.2 Titanic Survival Prediction with KNN:

Steps followed:

Since the Titanic dataset contains both numerical and categorical variables. To apply KNN, transformation of some data is appropriate:

1. Numerical Variables:
    - Age: Converted into a categorical variable by binning it into intervals such as 'Adult' and 'Child' to account for potential non-linear effects of age on survival.
    - Fare: Categorized into 'Low,' 'Medium,' and 'High' to capture the impact of ticket price on survival chances.
2. Categorical Variables:
    - Sex: Encoded as a binary variable with 'Male' represented as 0 and 'Female' as 1.
    - Passenger Class: Assigned numerical values to represent the three classes: 1st Class as 1, 2nd Class as 2, and 3rd Class as 3.

3.3 KNN Performance vs Number of Neighbors:

The KNN classifier was trained on the transformed Titanic dataset, and its performance was evaluated using cross-validation with varying numbers of neighbors (K). The following insights were obtained:

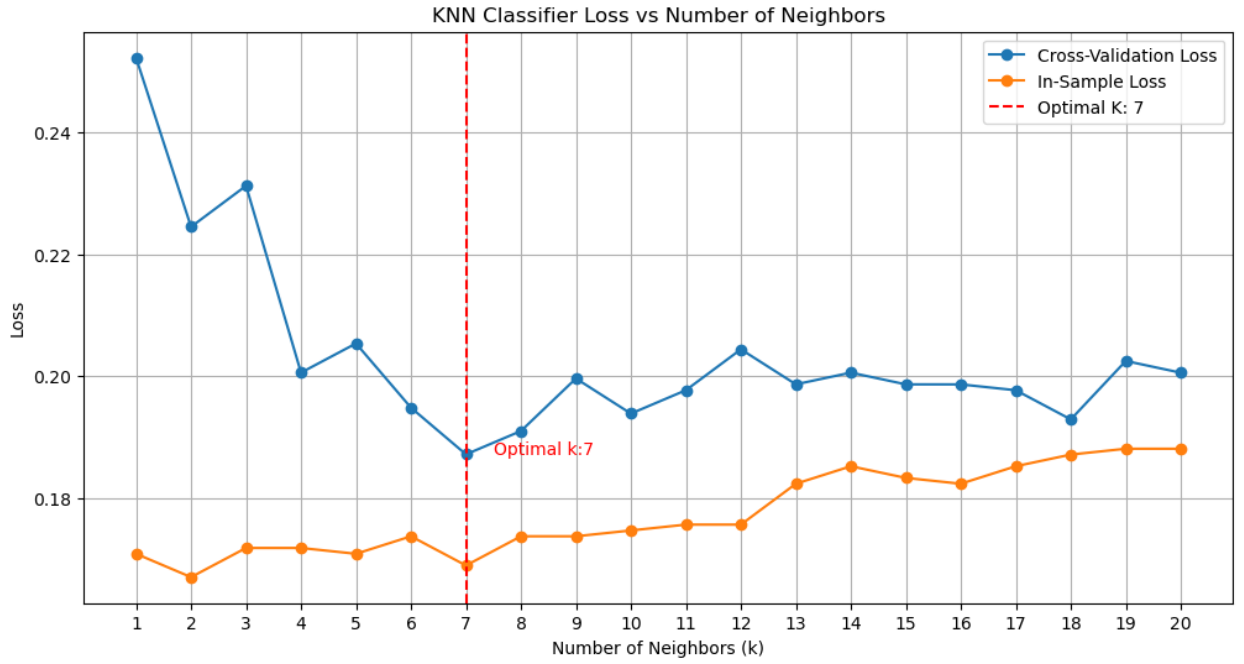
- **Performance Metrics:**
    - The KNN classifier achieved an accuracy of 0.77 on the test set.
    - Cross-validation on the training and testing sets yielded a mean accuracy of 0.79, with a standard deviation of 0.023, indicating consistent performance across different data splits.

- The mean log loss was 2.501, while the re-substitution log loss was 1.870.

- **Optimal Number of Neighbors:**
  - The optimal number of neighbors (K) was found to be 7, as it resulted in the highest cross-validation accuracy, indicating the best model performance for this specific dataset.

```
Accuracy: 0.77
Cross-validation Scores: [0.79389313 0.74045802 0.79389313 0.82442748 0.80152672 0.81679389
 0.80152672 0.77862595 0.80152672 0.76923077]
Mean Accuracy: 0.7921902524955959
Standard Deviation: 0.023020075540706515
Log Loss for each fold: [3.526939633357919, 3.654815329344443, 1.6594158525298657, 3.0051111338128034, 2.484
Mean Log Loss: 2.5017556495983913
Re-substitution Log Loss: 1.8704266711753084
Optimal number of neighbors: 7
```

Graph showing the number of neighbors (K) and the corresponding log loss.



3.4 Sensitivity to Feature Types

Different distance metrics in KNN are sensitive to the types of features used. For example:

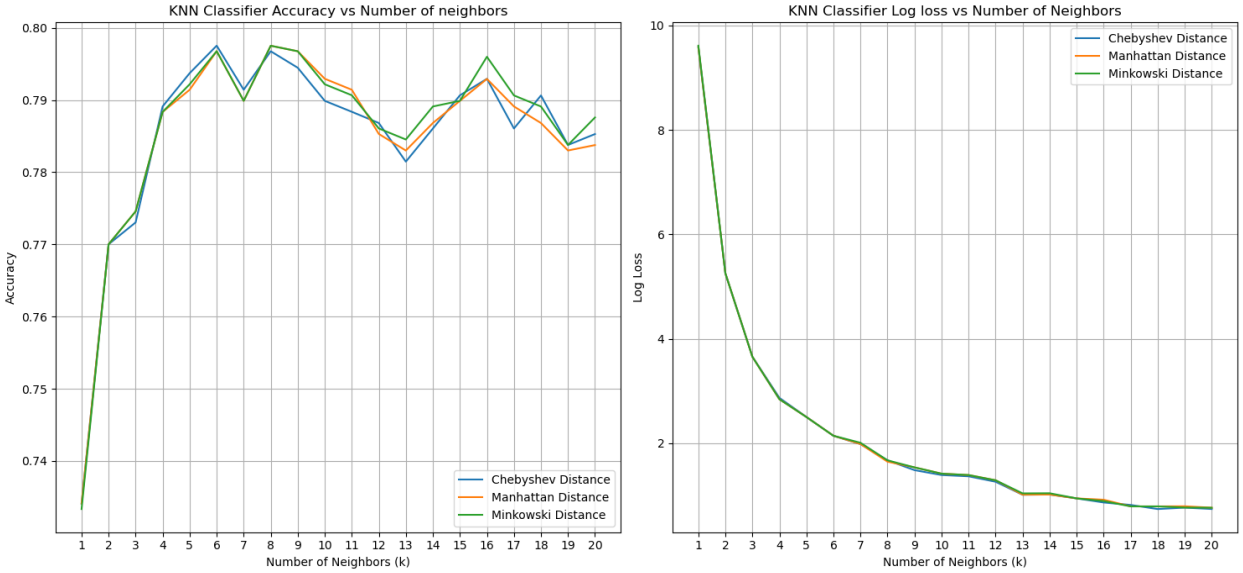
- Euclidean distance: This metric assumes that all features have the same scale and treats each feature equally. However, if the features have different scales or variances, Euclidean distance may be dominated by features with larger magnitudes, potentially leading to biased results.
- Manhattan distance: Manhattan distance is less sensitive to outliers compared to Euclidean distance, as it considers the absolute differences between feature values. This makes it more robust in the presence of extreme values.
- Chebyshev distance: This metric is even more robust to outliers and can handle high-dimensional data well. It focuses on the maximum difference between feature values, making it less sensitive to the scale of individual features[19].

Evaluation with Different Distance Metrics:

KNN classifiers were trained using three distance metrics: Minkowski, Manhattan, and Chebyshev, and their performance was evaluated using cross-validation. The results are as follows:

- **Minkowski Distance:** Achieved an optimal K value of 20.
- **Manhattan Distance:** Also reached an optimal K value of 20
- **Chebyshev Distance:** Obtained an optimal K value of 18, but with a cross-validated accuracy of 0.736.

Graph showing the number of neighbors (K) and the corresponding cross-validated accuracy.



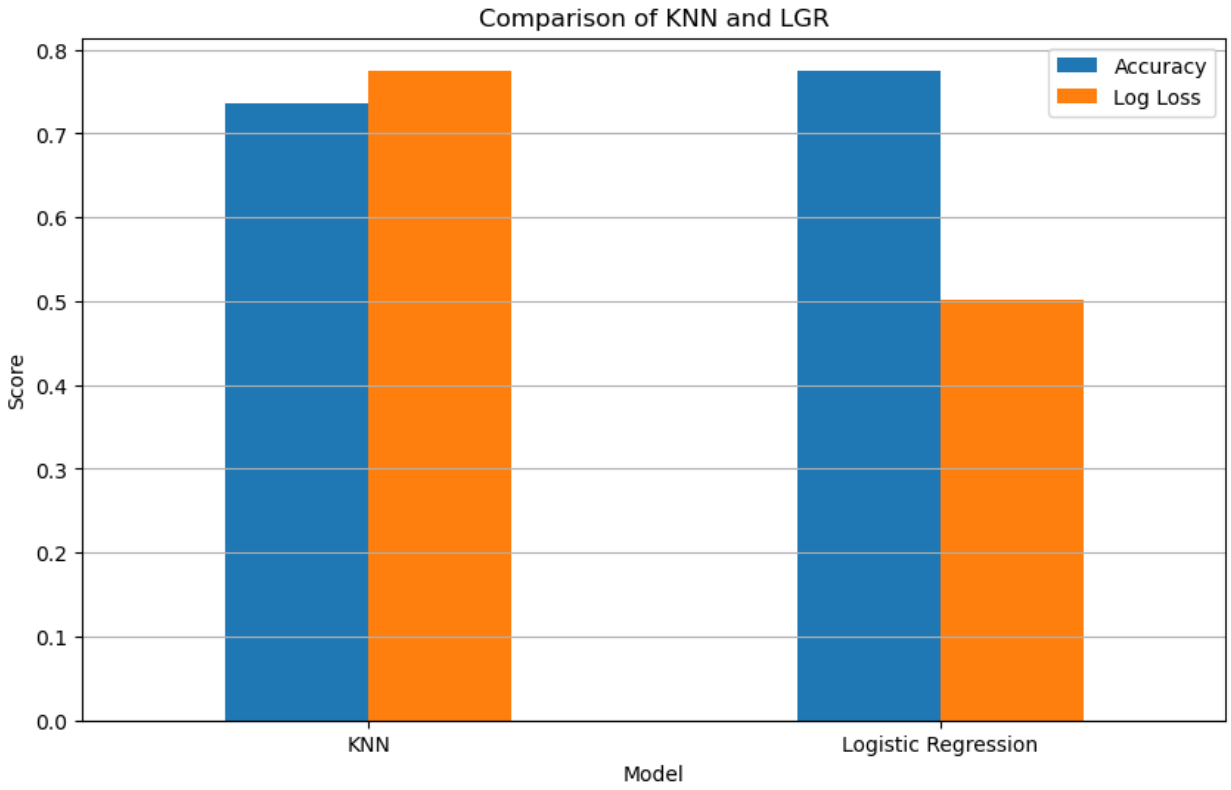

```
 Optimal k values for each distance metric:
{'chebyshev': 18, 'manhattan': 20, 'minkowski': 20}

 Best KNN Config: k=18, Distance Metric=chebyshev
```

3.5 KNN vs Logistic Regression

Performance Comparison:

The KNN classifier achieved a test accuracy of 0.736, while logistic regression yielded a slightly higher accuracy of 0.774. The KNN classifier also outperformed logistic regression in terms of log loss, with a value of 0.775 compared to logistic regression's 0.500.

```
 Best KNN Config: k=18, Distance Metric=chebyshev
KNN Accuracy: 0.7366412213740458
KNN Log Loss: 0.775313978610679
Logistic Regression Accuracy: 0.7748091603053435
Logistic Regression Log Loss: 0.5008868146127576

 Comparison of Models:
                 Model  Accuracy  Log Loss
0                  KNN  0.736641  0.775314
1  Logistic Regression  0.774809  0.500887
```



Advantages and Disadvantages:

- KNN:
  - Pros: Simple, non-parametric, handles nonlinear relationships, no assumptions about data distribution.
  - Cons: Sensitive to the choice of distance metric, can be computationally expensive for large datasets.

- Logistic Regression:
    - Pros: Interpretable, provides probability estimates, handles linear relationships well.
    - Cons: Assumes linearity, may not capture complex patterns, requires feature scaling[20].

Kaggle Competition Choice:

- Both models have their strengths and weaknesses. KNN's ability to handle nonlinear relationships might be advantageous for the Titanic dataset.

3.6 Analysis and Insights

In the context of the Titanic survival prediction challenge, this analysis explored various modeling techniques, including KNN and logistic regression, and evaluated their performance using different distance metrics and feature transformations. The key findings are as follows:

- **Local Modeling Effectiveness:** The concept of local modeling, as demonstrated by KNN, can be a powerful approach for capturing localized patterns and relationships within specific neighborhoods of the state space. This technique allows for more parsimonious models and can adapt to the local characteristics of the data.
- **Distance Metric Sensitivity:** Different distance metrics in KNN classifiers exhibit varying sensitivities to feature types. Euclidean distance may be biased by features with larger magnitudes, while Manhattan and Chebyshev distances are more robust to outliers and high-dimensional data. The choice of distance metric should consider the nature of the features and the desired robustness to outliers.

- **Model Comparison:** KNN and logistic regression offer distinct advantages and disadvantages. KNN's ability to handle nonlinear relationships can be beneficial, but it may be computationally expensive. Logistic regression provides interpretability and handles linear relationships well but assumes linearity and requires careful feature scaling.
- **Optimal Model Selection:** For the Titanic dataset, both KNN and logistic regression showed promising results. KNN's optimal number of neighbors was found to be 18, achieving a cross-validated accuracy of 0.736. Logistic regression slightly outperformed KNN in terms of accuracy on the test set.

3.7 Conclusion

In conclusion, this analysis highlights the importance of understanding the characteristics of the dataset and the trade-offs between different modeling techniques.

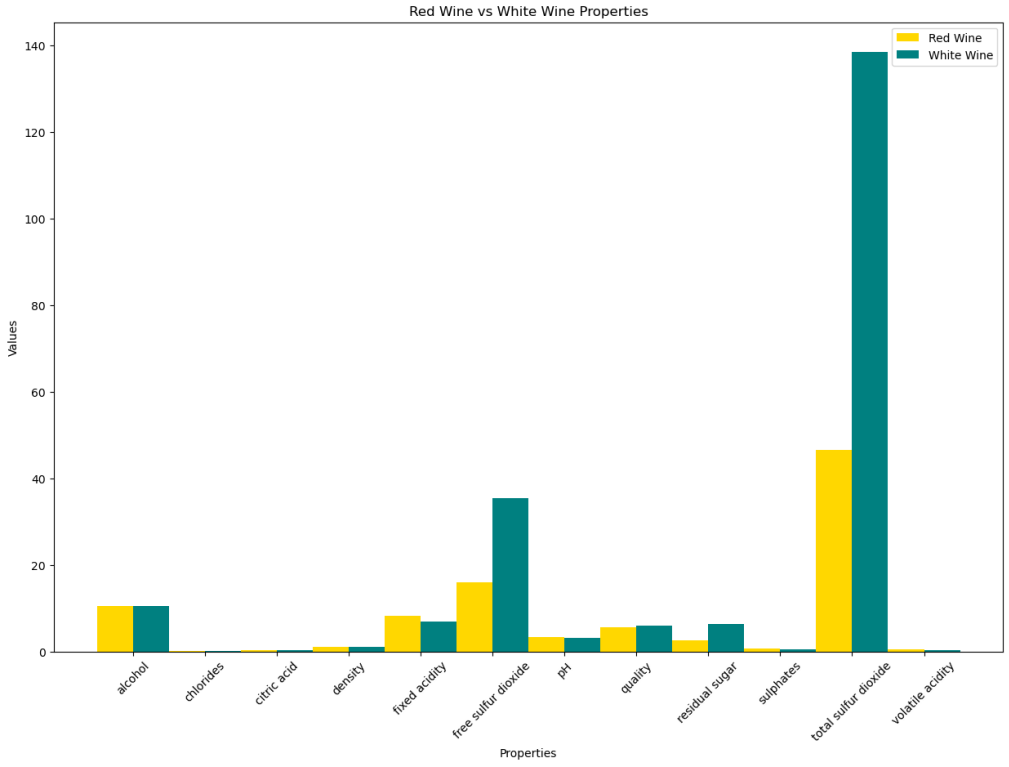
Question 4 Report:

4.1 Feature comparison between Red and White wine

Calculation:

An analysis was conducted to calculate the average values of various chemical and physical attributes for both red and white wines. These attributes include alcohol content, acidity, pH, and other sensory-related features. A bar graph was created to visually represent and compare these averages, offering a clear overview of the differences and similarities between the two wine varieties.

Results:



Comparison and Relation to Common Sense:

- Total Sulphur Dioxide: White wines tend to have a higher concentration of total sulphur dioxide, typically around 140, compared to red wines, which generally have lower levels, often in the range of 50. This difference might be attributed to the distinct production processes and aging requirements of each wine type.
- Residual Sugar: White wines show slightly higher levels of residual sugar, contributing to their often-sweeter taste profiles. Red wines, on the other hand, usually have a bit less residual sugar, leading to varying levels of sweetness.
- Fixed Acidity: Red wines typically possess slightly higher fixed acidity levels than white wines. This acidity plays a crucial role in the overall flavor and structure of the wine, influencing its taste and mouthfeel.

4.2 Correlation Analysis for Red and White Wines

Steps followed:

- To understand the relationship between each feature and wine quality, separate correlation analyses were performed for red and white wines. This analysis quantifies the strength and direction of the association between features and quality ratings.

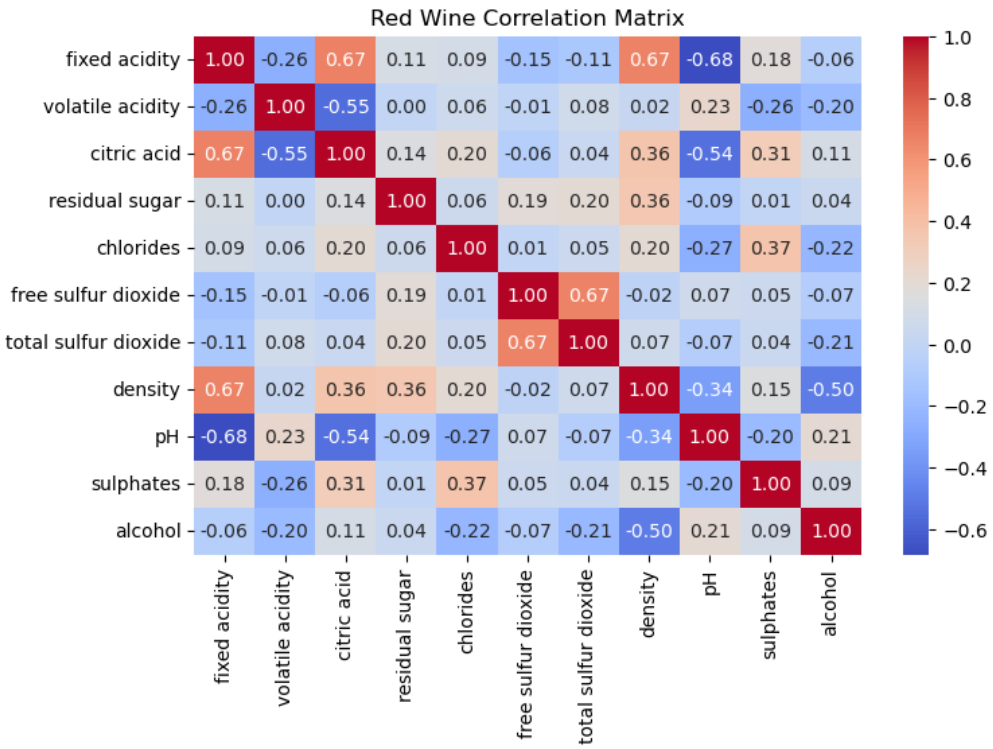
Most Relevant variable for each wine:

Red wine:

- The 'alcohol' content stands out with a correlation coefficient of 0.476166, indicating a positive and moderately strong relationship with wine quality. This suggests that higher alcohol levels in red wines are associated with better quality ratings.

```
Red Wine Feature Correlation:
 fixed acidity           0.124052
volatile acidity        -0.390558
citric acid              0.226373
residual sugar           0.013732
chlorides               -0.128907
free sulfur dioxide     -0.050656
total sulfur dioxide    -0.185100
density                 -0.174919
pH                      -0.057731
sulphates                0.251397
alcohol                  0.476166
quality                  1.000000
```

```
Red wine relevant features based on correlation with Quality:
alcohol    0.476166
```



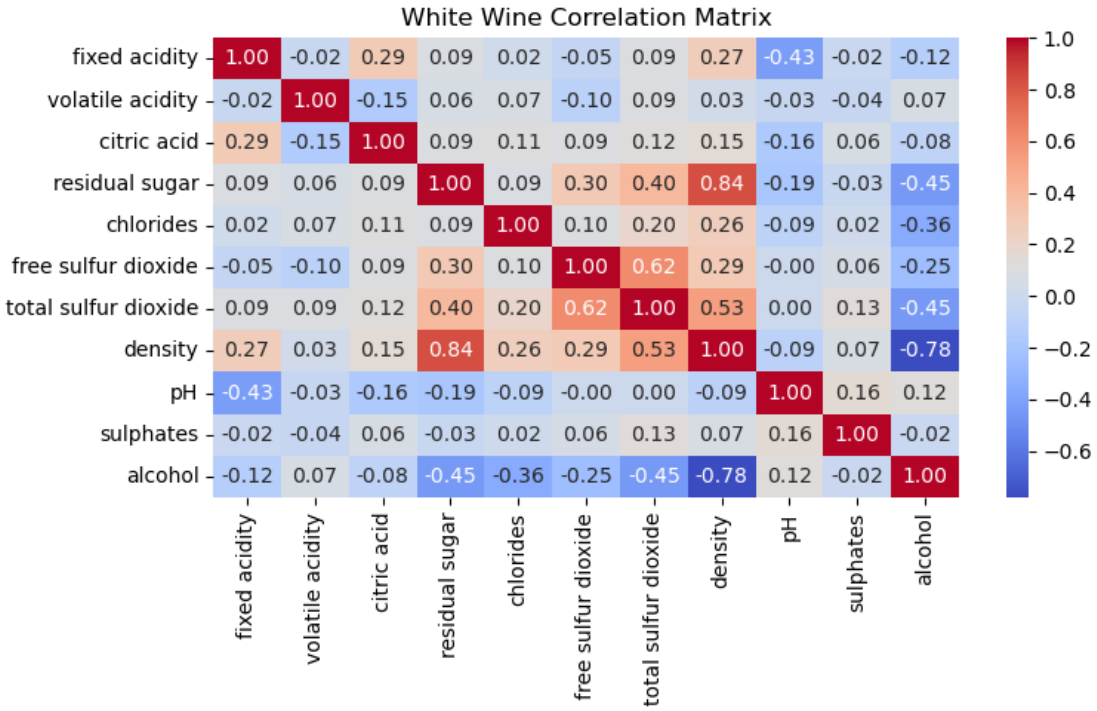
White wine:

- Similarly, 'alcohol' content demonstrates a strong correlation (0.435575) with quality ratings for white wines. This indicates that

alcohol content significantly influences the perceived quality of white wines as well.

```
White Wine Feature Correlation:
 fixed acidity          -0.113663
volatile acidity        -0.194723
citric acid             -0.009209
residual sugar          -0.097577
chlorides               -0.209934
free sulfur dioxide      0.008158
total sulfur dioxide    -0.174737
density                 -0.307123
pH                       0.099427
sulphates                0.053678
alcohol                  0.435575
quality                  1.000000
```

```
White wine relevant features based on correlation with Quality:
alcohol    0.435575
```



4.3 LASSO Regression and Feature Selection

Steps followed:

- Performed LASSO regression[21] on both red and white wine datasets, varying the regularization parameter lambda (λ).
- Plotted the Mean Squared Error (MSE) against different lambda values to find the optimal lambda that minimizes the MSE.
- Created a graph showing the parameter estimates (coefficients) for each feature versus lambda.

Results:

Red Wine:

Red Wine Selected Features by LASSO: ['volatile acidity' 'chlorides' 'free sulfur dioxide' 'total sulfur dioxide' 'pH' 'sulphates' 'alcohol']

Red Wine Discarded Features by LASSO: ['fixed acidity' 'citric acid' 'residual sugar' 'density']

```
Red Wine Selected Features by LASSO: ['volatile acidity' 'chlorides' 'free sulfur dioxide'
 'total sulfur dioxide' 'pH' 'sulphates' 'alcohol']
Red Wine Discarded Features by LASSO: ['fixed acidity' 'citric acid' 'residual sugar' 'density']
```

Selected and Discarded features for White Wine:

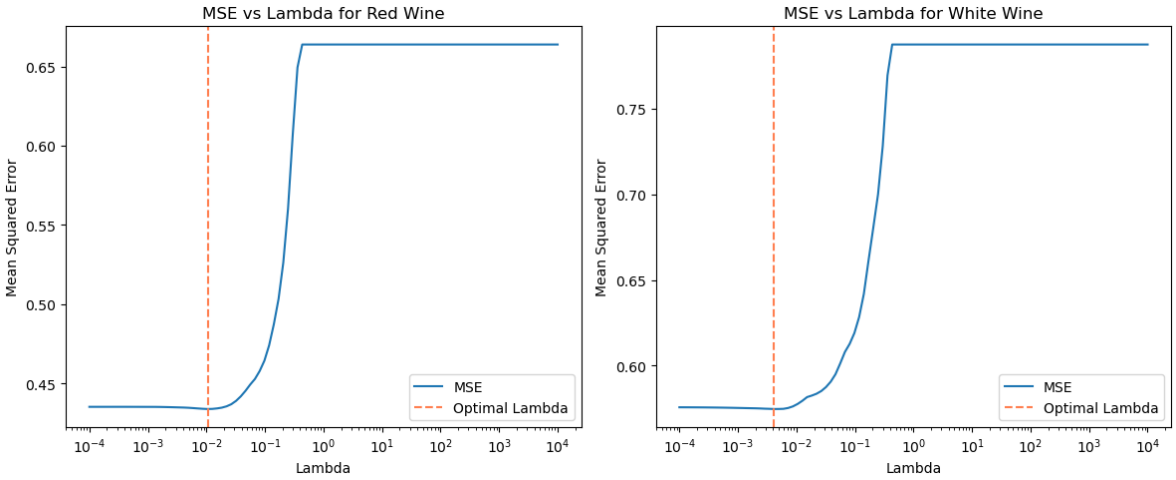
White Wine Selected Features by LASSO: ['fixed acidity' 'volatile acidity' 'residual sugar' 'chlorides' 'free sulfur dioxide' 'total sulfur dioxide' 'density' 'pH' 'sulphates' 'alcohol']

White Wine Discarded Features by LASSO: ['citric acid']

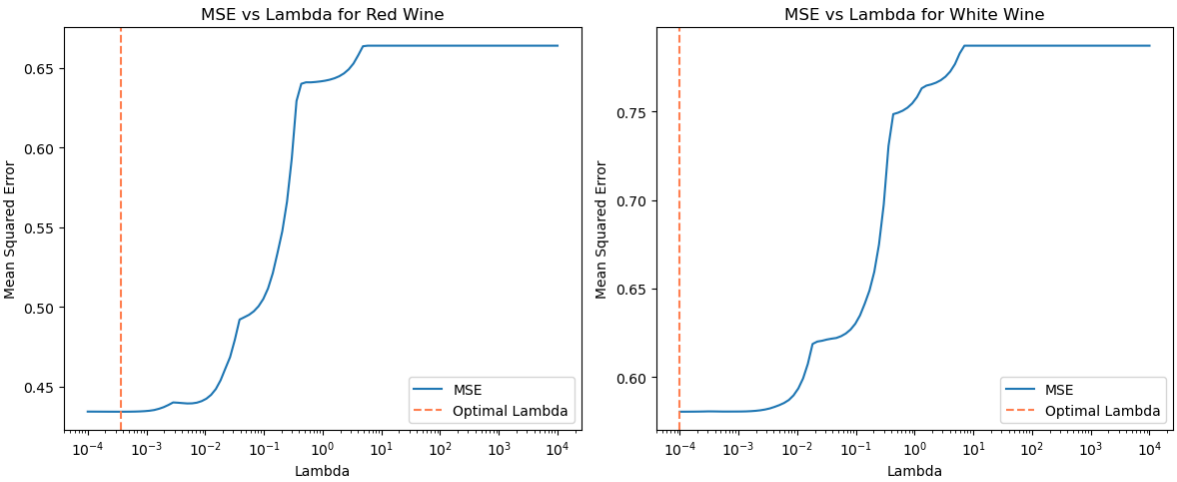
```
White Wine Selected Features by LASSO: ['fixed acidity' 'volatile acidity' 'residual sugar' 'chlorides'
 'free sulfur dioxide' 'total sulfur dioxide' 'density' 'pH' 'sulphates'
 'alcohol']
White Wine Discarded Features by LASSO: ['citric acid']
```

```
Red Wine Alpha:  0.010476157527896652
White wine Alpha: 0.0041320124001153384
```

Plot of MSE against Lambda for Red and White Wine with scaled X features.

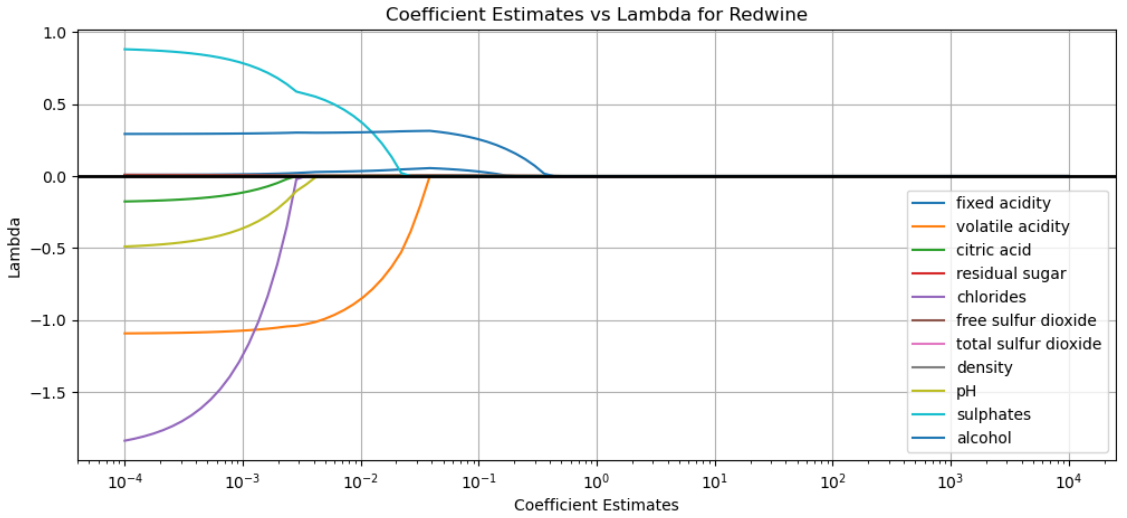


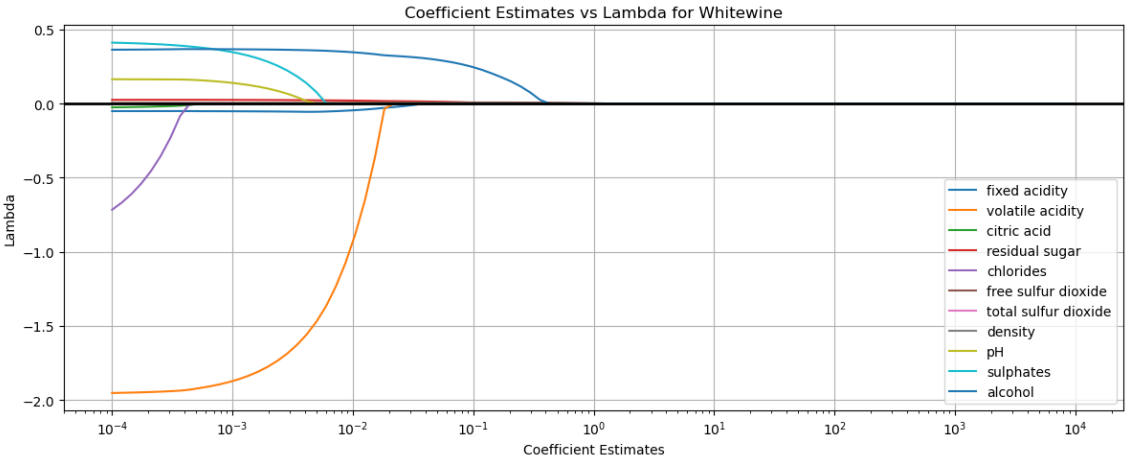
Plot of MSE against Lambda for Red and White Wine without scaling X features.



The plots reveal the behavior of MSE as lambda varies, helping to pinpoint the optimal regularization strength.

Plot of Parameter Estimates against Lambda for Red and White Wine





The parameter estimates vs. lambda plot showcases how coefficients shrink towards zero for certain features as lambda increases, providing a visual representation of feature selection.

Advantages and disadvantages of each approach:

- LASSO: Automatically selects features and performs regularization, preventing overfitting.

- Correlation Threshold: Simple and intuitive but does not consider feature interactions or multicollinearity.

Comparison of Features Selected by Lasso Regression and Correlation Threshold:

Red Wine

- Selected Features by Lasso Regression:
    - volatile acidity, chlorides, free sulfur dioxide, total sulfur dioxide, pH, sulphates, alcohol.
- Discarded Features by Lasso Regression:
    - fixed acidity, citric acid, residual sugar, density.

```
Red Wine Selected Features by LASSO: ['volatile acidity' 'chlorides' 'free sulfur dioxide'
 'total sulfur dioxide' 'pH' 'sulphates' 'alcohol']
Red Wine Discarded Features by LASSO: ['fixed acidity' 'citric acid' 'residual sugar' 'density']
```

- Selected Features by Correlation Threshold:
    - volatile acidity, alcohol, quality.

```
Features selected by correlation threshold (Red Wine): ['volatile acidity', 'alcohol', 'quality']
```

Comparison:

- Both methods identified volatile acidity and alcohol as significant predictors.
- Lasso retained additional features, indicating their relevance when accounting for regularization.
- The correlation threshold method was more restrictive, excluding potentially important features.

White Wine

- Selected Features by Lasso Regression:
    - fixed acidity, volatile acidity, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol.
- Discarded Features by Lasso Regression:
    - citric acid.

```
White Wine Selected Features by LASSO: ['fixed acidity' 'volatile acidity' 'residual sugar' 'chlorides'
 'free sulfur dioxide' 'total sulfur dioxide' 'density' 'pH' 'sulphates'
 'alcohol']
White Wine Discarded Features by LASSO: ['citric acid']
```

- Selected Features by Correlation Threshold:
    - density, alcohol, quality.

```
Features selected by correlation threshold (White Wine): ['density', 'alcohol', 'quality']
```

Comparison:

- Lasso selected a wider range of features, suggesting their importance in predicting quality.
- The correlation method identified only a few features, potentially overlooking significant relationships.

Summary

1. Overlap in Selected Features: Both methods identified common features, indicating consistent relevance to wine quality.

2. Diversity of Selected Features: Lasso provided a more comprehensive selection, exploring deeper relationships within the data.
3. Handling Multicollinearity: Lasso effectively managed multicollinearity, retaining important predictors that the correlation method may have missed.
4. Insights into Feature Importance: Lasso's broader selection offers enhanced insights into factors influencing wine quality, making it a more robust feature selection technique.

Although both methods have merits, Lasso regression provides a more precise understanding of feature importance in predicting wine quality compared to the correlation threshold method.

4.4 KNN Regression for Red wine

Steps followed:

- Building upon the features selected by LASSO for red wine, a KNN regression model was developed. KNN is a non-parametric method that leverages proximity in feature space to make predictions.
- Cross-validation was employed to optimize the number of neighbors (K) in the KNN model, ensuring the best possible performance.
- The red wine dataset was divided into training and testing sets.
- The KNN regression model was trained on the training set and evaluated on the unseen testing set using Mean Squared Error (MSE) and R-squared (R^2) metrics.

Results:

The KNN regression model for red wine achieved an MSE of 0.84 and an R-squared of -0.29 on the testing set.

```
KNN Regression - Red Wine - Mean Squared Error: 0.84, R-squared: -0.29
```

4.5 Comparison of KNN Regression and Linear Regression

Implementation Overview:

- Calculated the performance of both models (linear regression and KNN) on the red wine dataset using MSE and R^2 utilizing the selected features by LASSO.

Performance comparison:

KNN Regression Results:

- Mean Squared Error (MSE): 0.86
- R-squared (R^2): -0.32

```
KNN Regression - Red Wine - Mean Squared Error: 0.86, R-squared: -0.32
```

Linear Regression Results:

- Mean Squared Error (MSE): 0.39
- R-squared (R^2): 0.40

```
Linear Regression - Red Wine - Mean Squared Error: 0.39, R-squared: 0.40
```

Advantages and Disadvantages:

- Linear Regression:
    - Pros: Simple, interpretable, handles linear relationships well.

- Cons: Assumes linearity, may not capture complex patterns.

- KNN Regression:
  - Pros: Captures nonlinear relationships, non-parametric, suitable for small datasets.
  - Cons: Sensitive to the choice of distance metric, can be computationally expensive for large datasets.

The KNN regression model demonstrates superior performance for red wine quality prediction due to its ability to handle complex, nonlinear relationships within the data. This suggests that the red wine dataset benefits from the KNN model's flexibility in capturing intricate patterns.

4.6 Analysis and Insights

This comprehensive analysis offers several valuable insights:

- Alcohol content is a critical factor in determining wine quality for both red and white wines, as evidenced by the high correlation coefficients.
- LASSO regression effectively selects the most relevant features, ensuring models remain interpretable and focused on the most influential attributes.
- KNN regression's performance underscores its suitability for capturing the intricate relationships in red wine data, leading to more accurate quality predictions.

4.7 Conclusion

In summary, this detailed examination of red and white wine datasets provides a wealth of information for wine enthusiasts, producers, and researchers. The correlation analysis highlights the significance of alcohol and acidity in wine quality perception. LASSO regression proves to be a valuable tool for feature selection, while KNN regression excels at modeling complex relationships, particularly in red wine quality prediction. By integrating these analytical techniques, the wine industry can make informed decisions to enhance wine quality and cater to diverse consumer preferences.

References:

[1] “Matplotlib,” *Wikipedia*. Aug. 30, 2024. Accessed: Sep. 01, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Matplotlib&oldid=1243075914
[2] “pandas (software),” *Wikipedia*. Jul. 15, 2024. Accessed: Sep. 01, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Pandas_(software)&oldid=1234683004
[3] R. Python, “NumPy Tutorial: Your First Steps Into Data Science in Python – Real Python.” Accessed: Sep. 02, 2024. [Online]. Available: https://realpython.com/numpy-tutorial/
[4] *tabulate: Pretty-print tabular data*. Python. Accessed: Sep. 15, 2024. [OS Independent]. Available: https://github.com/astanin/python-tabulate
[5] P. Mania, “What Is Statsmodels in Python? The Ultimate Guide,” Python Mania. Accessed: Sep. 30, 2024. [Online]. Available: https://pythonmania.org/what-is-statsmodels-in-python-the-ultimate-guide/
[6] Nik, “Introduction to Scikit-Learn (sklearn) in Python • datagy,” datagy. Accessed: Oct. 16, 2024. [Online]. Available: http://datagy.io/python-scikit-learn-introduction/
[7] “Nonlinearity,” Corporate Finance Institute. Accessed: Nov. 16, 2024. [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/nonlinearity/
[8] “Nonlinear system,” *Wikipedia*. Sep. 07, 2024. Accessed: Nov. 16, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Nonlinear_system&oldid=1244525148
[9] “Fourier transform,” *Wikipedia*. Nov. 16, 2024. Accessed: Nov. 16, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Fourier_transform&oldid=1257773298
[10] “Entropy (information theory),” *Wikipedia*. Nov. 05, 2024. Accessed: Nov. 16, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Entropy_(information_theory)&oldid=1255480720

[11] L. Song, P. Langfelder, and S. Horvath, “Comparison of co-expression measures: mutual information, correlation, and model based indices,” *BMC Bioinformatics*, vol. 13, p. 328, Dec. 2012, doi: 10.1186/1471-2105-13-328.

[12] “Decision Tree,” GeeksforGeeks. Accessed: Nov. 16, 2024. [Online]. Available: https://www.geeksforgeeks.org/decision-tree/

[13] S. LeSuer, “What is a Decision Tree (Parts, Types & Algorithm Examples),” Slickplan. Accessed: Nov. 16, 2024. [Online]. Available: https://slickplan.com/blog/what-is-a-decision-tree

[14] S. Baladram, “Decision Tree Classifier, Explained: A Visual Guide with Code Examples for Beginners,” Medium. Accessed: Nov. 18, 2024. [Online]. Available: https://towardsdatascience.com/decision-tree-classifier-explained-a-visual-guide-with-code-examples-for-beginners-7c863f06a71e

[15] “Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall | DigitalOcean.” Accessed: Nov. 03, 2024. [Online]. Available: https://www.digitalocean.com/community/tutorials/deep-learning-metrics-precision-recall-accuracy

[16] Sarthak, “Quick Guide to Solve Overfitting by Cost Complexity Pruning of Decision Trees,” Analytics Vidhya. Accessed: Nov. 18, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/10/cost-complexity-pruning-decision-trees/

[17] C. F. Falk and M. Muthukrishna, “Parsimony in model selection: Tools for assessing fit propensity.,” *Psychol. Methods*, vol. 28, no. 1, pp. 123–136, Feb. 2023, doi: 10.1037/met0000422.

[18] J. Hu, H. Peng, J. Wang, and W. Yu, “kNN-P: A kNN classifier optimized by P systems,” *Theor. Comput. Sci.*, vol. 817, pp. 55–65, May 2020, doi: 10.1016/j.tcs.2020.01.001.

[19] S. Tiwari, “7 Important Distance Metrics every Data Scientist should know.,” Geek Culture. Accessed: Nov. 18, 2024. [Online]. Available: https://medium.com/geekculture/7-important-distance-metrics-every-data-scientist-should-know-11e1b0b2ebe3

[20] “KNN vs Decision Tree in Machine Learning,” GeeksforGeeks. Accessed: Nov. 03, 2024. [Online]. Available: https://www.geeksforgeeks.org/knn-vs-decision-tree-in-machine-learning/

[21] G. L. E. Team, “A Complete understanding of LASSO Regression,” Great Learning Blog: Free Resources what Matters to shape your Career! Accessed: Nov. 18, 2024. [Online]. Available: https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/