REPORT

1. Nonlinearity

1.1 Why might be necessary to consider nonlinear relationships between variables.

Definition

Nonlinearity is defined as an environment in which an independent variable and a dependent variable do not have a direct or straight-line relationship. This could indicate that there is little or no relationship between the two entities. Though more complicated than in a linear relationship, nonlinear entities may still possess reasonably predictable relationships with one another.[1]

Reason why nonlinear relationship between variable can be considered

Working with nonlinear models presents greater challenges compared to linear models, demanding increased amounts of training data, computational resources, and tuning efforts to achieve optimal results. Nonetheless, nonlinear models prove superior for addressing real-world problems, enabling data scientists to uncover trends that would be otherwise undetectable.

Nonlinear models manifest in diverse structures, ranging from polynomial models adept at fitting data curves to neural networks designed to comprehend intricate patterns within high-dimensional datasets. Their potency surpasses that of linear models, as they can encapsulate intricate relationships between variables. In classification scenarios, nonlinear models excel in delineating complex decision boundaries that effectively separate distinct classes, particularly in cases where these boundaries defy linearity.

While linear models serve as a solid initial approach, they may fall short when confronted with more intricate problems. The necessity for nonlinear relationships becomes apparent as they offer increased power and flexibility, allowing for a more refined and effective representation of complex relationships within the data.[2]

1.2 The mathematical equation for a nonlinear model and provide an example of an application where it might be appropriate

Nonlinear regression models are highly beneficial in complex engineering applications for several reasons. Firstly, their flexibility surpasses that of linear models, as they can represent curved or exponential relationships between variables, making them adept at describing intricate behaviours. Secondly, nonlinear regression enables the incorporation of engineering knowledge into the model; if a known relationship, such as an exponential one, exists between variables, it can be explicitly included. Lastly, the capacity to fit complex models to data enhances the accuracy and precision of predictions, proving invaluable in various engineering scenarios.

Nonlinear regression models are represented by a mathematical equation. The formula varies depending on the relationship between your variables. However, a simple generic nonlinear regression model is given by:

$$y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$$

Where:

y: Is the dependent variable.

X: Is the independent variable.

$\beta_0, \beta_1, \beta_2$: Are the parameters of the model.

ε: Is the error term.

The quadratic equation represents just one variant of nonlinear regression models. Numerous other functions are available, such as logarithmic, exponential, power functions, and more. The selection often hinges on engineering intuition, specific use cases, and domain expertise.

A simple nonlinear regression model is expressed as follows:

$$y = f(X, \beta) + \varepsilon$$

Where:

- **X:** is a vector of P predictors
- $\boldsymbol{\beta}$**:**is a vector of k parameters
- $f(-)$: is the known regression function
- ε :is the error term

Nonlinear regression involves employing a mathematical function, usually a curve, to establish an equation that fits given data. The fitness of a regression model is evaluated using the sum of squares, obtained by determining the difference between the mean and each data point. Nonlinear regression models are preferred for their capability to adapt to various mean functions.[3]

Application of nonlinear model

A nonlinear model can be used to predict the stock market based on historical data and indicators, or to classify an image as a cat or a dog based on its pixels.

In chemical engineering, nonlinear regression models are utilized to depict chemical kinetics, considering variables like temperature and reactant concentration in a nonlinear fashion. For instance, in batch chemical processes, where temperature, concentration, and pressure nonlinearly influence reaction kinetics, applying a nonlinear regression model allows for a comprehensive understanding of individual variables' behaviour. This approach proves valuable in designing enhanced and more efficient chemical reactors.

An instance illustrating the application of nonlinear regression involves predicting population growth over time. Observing a scatterplot of evolving population data reveals a non-linear relationship between time and population growth. To accurately capture this relationship, a nonlinear regression model, such as a logistic population growth model, can be employed to generate population estimates.[4], [5]

1.3 Can a nonlinear model be more parsimonious than a linear model? Write down mathematical formulae for both the linear and nonlinear models to support your answer.

I believe yes, nonlinear models can offer increased parsimony compared to linear counterparts when the underlying relationship in the data is inherently nonlinear. While the potential for complexity and overfitting exists in nonlinear models, well-regularized approaches can effectively balance capturing the true pattern without succumbing to unnecessary intricacies. In contrast, forcing a linear model to approximate a nonlinear relationship may necessitate a convoluted representation with numerous parameters, diminishing interpretability and potentially compromising performance. The choice between linear and nonlinear models hinges on the nature of the data, and a judiciously regularized nonlinear model can often provide a more concise and accurate portrayal of the underlying complexity while avoiding the pitfalls of overfitting associated with linear models attempting to emulate nonlinear patterns.

Certainly, a nonlinear model can be more parsimonious than a linear model. Parsimony in statistical modelling refers to achieving a good fit with the data using a minimal number of parameters. Let's consider the logistic equation as an example of a nonlinear model and a polynomial of degree five as an example of a linear model.

The logistic equation, a common nonlinear model, is given by:

$$Y(t) = \frac{L}{1 + e^{-k(t-t_m)}}$$

Here, $Y(t))$ is the response variable, L is the asymptotic parameter, t_m is the inflection point, and k determines the steepness of the curve.

Now, let's consider a fifth-degree polynomial as a linear model:

$$Y(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

In this polynomial, $a_0, a_1, a_2\ a_3, a_4$ and a_5 are coefficients associated with the terms.

In many cases, a nonlinear model like the logistic equation can be more parsimonious because it captures complex relationships with just a few parameters (in this case, three: (L, t_m) and k). On the other hand, a fifth-degree polynomial would require six parameters, and the interpretation of each parameter becomes more challenging and less intuitive. The advantage of parsimony in nonlinear models is particularly evident when the data can be adequately described by a model with fewer parameters, leading to a more interpretable and efficient representation of the underlying biological or physical processes.

Nonlinear models play a pivotal role in agriculture, particularly in capturing dynamic relationships within soil and plant sciences. Exponential decay and growth functions are widely employed to model nutrient decay in soils over time and describe processes like plant

population expansion or organic matter accumulation. These models offer valuable insights into optimal conditions for plant growth and nutrient management, guiding agricultural practices by predicting decay rates or maximum values.

Sigmoid curves, known for their distinctive S shape, are equally essential nonlinear models in agriculture. Applied to various plant characteristics such as height, weight, leaf area index, and seed germination, sigmoid curves help elucidate complex biological processes. Their versatility allows for a nuanced understanding of critical phases in plant development, offering insights into how these characteristics respond to factors like time, nitrogen application rate, or herbicide dose. The integration of these nonlinear models enhances precision in scientific insights, contributing to more informed and efficient agricultural management practices.[6]

1.4 Surrogate data are used for testing for nonlinearity. What characteristics are typically preserved when generating surrogates? Give the names of two surrogate techniques and describe the approaches for implementing them.

Surrogate data involve modifying a time series to retain certain characteristics while altering others. These alterations serve the primary purpose of testing hypotheses regarding the time series structure, such as its randomness or nonlinearity. By comparing the original data to the surrogate, researchers can assess and validate various aspects of the time series, gaining insights into its underlying properties. In other words, Surrogate data testing is an essential part of many of these methods, as it enables robust statistical evaluations to ensure that the results observed are not obtained by chance but are a true characteristic of the underlying system.[7]
The methodology entails defining a null hypothesis (H0) that characterizes a linear process, followed by the generation of multiple surrogate datasets based on H0 using Monte Carlo methods. Subsequently, a discriminative statistic is computed for both the original time series and the entire set of surrogate data. If the statistic's value significantly differs between the original series and the surrogate set, the null hypothesis is dismissed, and non-linearity is inferred.

Two Surrogate Technique
Phase Randomization
Shuffle Randomization

Phase randomization is a surrogate data method employed to create surrogate datasets by perturbing the phase information while maintaining the amplitude characteristics of the original data. The process entails extracting the phase details from the initial dataset, introducing randomness (e.g., through shuffling or applying random phase shifts), and subsequently reconstructing the surrogate data by amalgamating the randomized phase information with the unaltered amplitude details. The resultant surrogate data retains certain statistical properties of the original dataset but eliminates any temporal or phase-related associations. This technique proves valuable in evaluating the significance of observed temporal or phase relationships in the original dataset, facilitating hypothesis testing and uncovering underlying patterns in time series data.

The generation of randomly shuffled surrogates involves a specific procedure: first, the original signal is divided into blocks, and then these blocks are permuted or shuffled. During this process, the positions of the blocks are randomized, and blocks located at the end of the

signal are wrapped around to the start of the time series. The key characteristic of randomly shuffled surrogates is that they preserve short-term correlations within individual blocks, maintaining local relationships. However, any long-term dynamical information in the signal is disrupted, as the random shuffling alters the sequential arrangement of blocks. This technique is particularly useful when assessing the impact of short-term correlations in the data while eliminating or disrupting long-term patterns, providing insights into the temporal structure of the original signal. [8], [9]

1.5 Define information, entropy and mutual information using mathematical formulas. Describe how entropy can be used for constructing a feature for measuring regularity and give an example of an application. Explain how mutual information can be used for feature selection and why it might be better than correlation.

Defining with Mathematical formulas.

1. Information

Information is a measure of the uncertainty associated with a particular event. The more surprising an event, the more information it provides. It is often measured in bits.
The formula for information content $I(x)$ of an event x is given by:

$$I(x) = -\log(P)$$

Where:

- $P(x)$ is the probability of event x
- $\log(P)$ is the logarithm.

Properties of Information Content:

- **Non-negativity:**
  - $I(P) \geq 0$
  - Information content is always a non-negative quantity.
- **Zero for Certainty:**
  - $I(1) = 0$
  - Events that always occur (certainties) convey no information, and the information content is zero in such cases.
- **Additivity for Independent Events:**
  - $I(P_1P_2) = I(P_1) + I(P_2)$
  - The information content due to independent events is additive. If two events are independent, the information associated with both events occurring together is the sum of the information associated with each event individually.[10], [8]

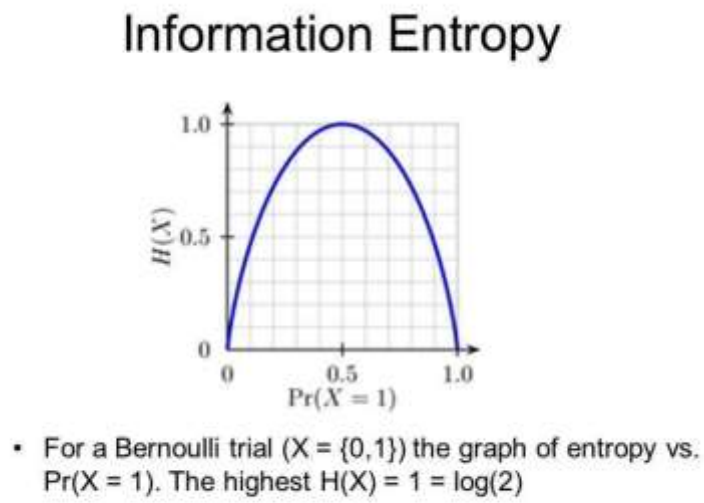
2. Entropy

Entropy is the measures the level of disorder or uncertainty in a given dataset or system. It is a metric that quantifies the amount of information in a dataset, and it is commonly used to evaluate the quality of a model and its ability to make accurate predictions.

$$H(x) = \sum(pi * \log_2 pi)$$

$$Entropy(P) = -\sum_{i=1}^{N} pi * \log_2(pi)$$

The representation of uncertainty or impurity is expressed through the logarithm to the base 2 of the probability of a given category (pi), with the index (i) corresponding to the number of potential categories. In the context of binary classification, where i equals 2, this mathematical formulation captures the inherent uncertainty associated with each category. The resulting equation is visually illustrated by a symmetric curve, showcasing the relationship between the probability of an event on the x-axis and the corresponding heterogeneity or impurity denoted by H(X) on the y-axis. This graphical representation provides a clear depiction of how varying probabilities influence the level of uncertainty or impurity within the classification system.



Source: Analytics Vidhya

Figure 1 Example of Entropy in Machine learning

3. Mutual information

Mutual Information (MI) is a non-negative metric quantifying the mutual dependence between two random variables. This statistical measure gauges the extent to which observing the values of one variable informs us about the corresponding information in the second variable. It provides a numerical representation of the shared information or dependency between the two variables, highlighting the degree to which changes in one variable correlate with changes in the other. A higher mutual information indicates a stronger association.

By utilizing the relative entropy, we can now define the MI. We define the MI as the relative entropy between the joint distribution of the two variables and the product of their marginal distributions.
MI is given by:

$$I(X,Y) = \sum_{X}\sum_{Y} P(x,y)\log\left(\frac{P(x,y)}{P(x)P(y)}\right)$$

- $I(X;Y)$ is the mutual information between X and Y.
- $P(x,y)$ is the joint probability mass function of X and Y.
- $P_X(x)$ and $P_Y(y)$ are the marginal probability mass functions of X and Y respectively.
- The double summation is over all possible values of x and y that the random variables X and Y can take.
- log is the logarithm.

Alternatively, the formula can be expressed in terms of entropies:

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

Where:

- $H(X)$ and $H(Y)$ are the entropies of X and Y respectively.
- $H(X,Y)$ is the joint entropy of X and Y

The Mitual Iinformation can be equal or greater than 0. When $p(x,y) = p(x)\,p(y)$, the MI is 0. The joint probability is equal to the product of the marginals when there is no association between the variables. When the MI is 0, then knowing the values of x does not tell us anything about y, and vice versa, that is knowing y, does not tell us anything about x.[11]

Correlation

The correlation coefficient is an important measure of the relationship between two random variables. Once calculated, it describes the validity of a linear fit. For two random variables, X and Y, the correlation coefficient, ρxy, is calculated as follows:

$$\sigma_{xy} = \frac{cov(X,Y)}{\sigma_x \sigma_y}$$

The correlation coefficient will take on values from 1 to -1. Values of 1 and -1 indicate perfect increasing and decreasing linear fits, respectively. As the population correlation coefficient approaches zero from either side, the strength of the linear fit diminishes. Approximate ranges representing the relative strength of the correlation are shown below. See the Correlation article on Wikipedia for more detailed information about the correlation theory establishing these ranges. The ranges also apply for negative values between 0 and -1.

Correlation analysis provides a quantitative means of measuring the strength of a linear relationship between two vectors of data. Mutual information is essentially the measure of how much “knowledge” one can gain of a certain variable by knowing the value of another variable.[12]

Relationship between Correlation and Mutual Information (MI)

Non-linear Relationships: While correlation measures just linear links, mutual information captures non-linear relationships between variables. As a result, mutual information is more adaptable and appropriate for a larger variety of interactions and data types[13]

Correlation analysis provides a quantitative means of measuring the strength of a linear relationship between two vectors of data. Mutual information is essentially the measure of how much "knowledge" one can gain of a certain variable by knowing the value of another variable.

Feature Importance: The amount of information a feature gives about the target variable is measured by mutual information. This can help choose the most informative features for a prediction task and order features according to relevance.[14]

2. Classification using trees

2.1 Decision trees are often used to transform a set of observations into a specific recommended action. Describe the components (nodes, branches) of a decision tree. Why might it be necessary to prune the tree? Why are decision trees an attractive method for classification in practical applications?

Definition of Decision tree

A decision tree is a type of supervised machine learning method that represents decisions, outcomes, and predictions through a tree-like structure resembling a flowchart. This tree is created through an algorithmic procedure involving a series of if-else statements, determining how to partition, categorize, and visually represent a dataset based on various conditions.

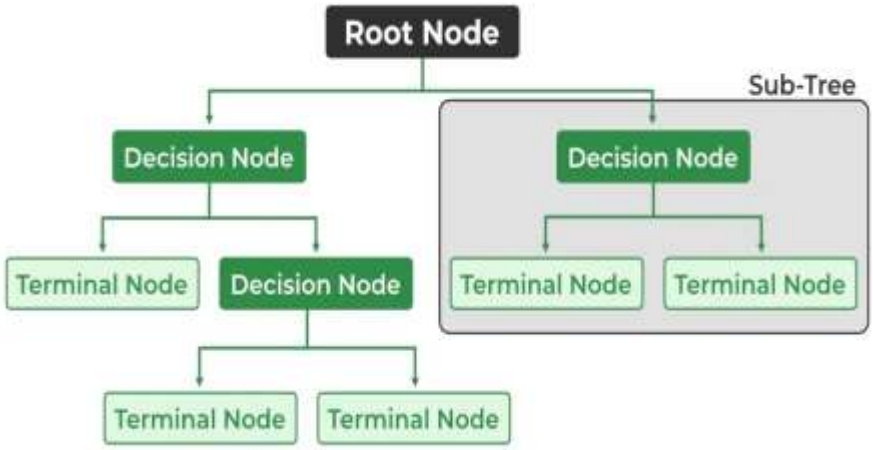
Decision Trees are a non-parametric supervised learning method that can be used for classification and regression tasks. The goal is to build a model that can make predictions on the value of a target variable by learning simple decision rules inferred from the data features.

Component of decision tree

Decision trees ultimately consist of just three key nodes:

- Decision nodes: In a decision tree, decision nodes are represented by squares. They stand for choices that must be made based on particular requirements or standards. Every decision node has branches that can lead to additional choices or to several potential outcomes.[15]
- Chance nodes: Representing probability or uncertainty (typically denoted by a circle): A node that symbolizes a choice regarding an input feature. Branching off internal nodes connects them to leaf nodes or other internal nodes. In the decision tree, chance nodes are represented by circles. They represent hazy results or occurrences with corresponding probability. The likelihood of various possibilities or results is displayed by chance nodes.[16],[15]

- End nodes: In a decision tree, leaf nodes also referred to as endpoint or terminal nodes are shown as rectangles or other shapes. They stand in for the conclusions or choices made at the end of a particular decision tree path. No more branching or splitting occurs in the leaf nodes.[15]



Source: Geeksforgeeks

- **Branch / Sub-Tree***:* A subsection of the entire decision tree is referred to as a branch or sub-tree. It represents a specific path of decisions and outcomes within the tree.

Why might it be necessary to prune the tree?

1. Mitigating overfitting: Allowing a decision tree to be fully trained may lead to overfitting on the training data. To address this, employing techniques to prevent overfitting is essential. One effective approach involves fine-tuning the hyperparameters of the decision tree model, enabling the pruning of trees and thereby preventing overfitting.[17]
2. Enhanced Generalization Through Pruning: The goal of Decision Tree Pruning is to craft an algorithm with decreased performance on training data but improved generalization on test data. Strategic adjustments to the hyperparameters of your Decision Tree model play a pivotal role in boosting its overall effectiveness, leading to time and resource savings in the process.[17]
3. Size Reduction for Enhanced Adaptability: Pruning is essential as it reduces the size of a Decision Tree. While this may lead to a marginal increase in training error, it significantly decreases testing error. This trade-off enhances the model's adaptability, ensuring more robust performance on new, unseen data.[18]

Reason why decision trees an attractive method for classification in practical applications?

- Easy to read and interpret: Decision trees offer the advantage of easily interpretable outputs, making them accessible to individuals without statistical expertise. In applications such as presenting customer demographic data to a marketing team, the graphical representation allows straightforward interpretation, facilitating informed decision-making without the need for statistical knowledge.[19]
- Easy to prepare: Decision trees require less data preparation effort compared to alternative techniques, though users must possess relevant information for creating predictive variables. This method enables straightforward classification without intricate calculations. In complex scenarios, users can enhance decision trees by integrating them with other methods for more comprehensive analysis.[19]

- Decision trees facilitate considering all potential outcomes for a problem, and they demand less data cleaning compared to alternative algorithms. This makes them advantageous for scenarios where a comprehensive exploration of possible solutions is essential with minimal pre-processing efforts.[16]

2.2 Suppose an organization has built a rule-based classifier using domain knowledge. After collecting a large amount of data, outline the steps required to improve upon the existing approach by constructing a data-driven classifier. How would you advise to test the validity of the new model?

These are the steps to build a data-driven classifier to improve existing rule-based classifier.[20], [21], [22], [23]

- Analyse existing rule: Evaluate the current rules within a rule-based classifier comprehensively, considering their accuracy, data classification capabilities, coverage, and identifying and resolving redundancies or conflicts to improve overall performance.

- Collect more data: Increase classifier accuracy by expanding the training dataset, enabling better pattern recognition through the acquisition of additional labelled data or dataset supplementation.

- Refine Rule selection: Optimize classifier performance by reassessing rule-based feature selection and threshold criteria through experimentation with diverse sets for improved accuracy.

- Refine the rule-based classifier by incorporating specific rules to address challenges or misclassifications encountered by the base classifier, thereby improving its ability to capture patterns and resolve conflicts more effectively.

- Assess the rule-based classifier's performance through cross-validation techniques, dividing the training dataset into subsets for training and evaluation, aiming to identify areas for improvement and evaluate the classifier's generalization capabilities.

- Iteratively refine and update the rule-based classifier by incorporating feedback and new data, regularly monitoring its performance, and making necessary adjustments to adapt to evolving patterns or requirements.

- Utilize optimization techniques like Harmony Search or genetic algorithms to enhance the rule-based classifier's performance by identifying the optimal set of rules and parameters for improved accuracy and effectiveness.

2.3 Consider the challenge of classifying the likelihood of survival using the Titanic dataset. Construct a decision tree and display the structure of this tree using a graphic.

I utilized various libraries, starting with importing pandas to extract the Titanic dataset. The sklearn library played a crucial role in splitting the dataset into training and testing sets using its dedicated function. For handling missing values, particularly in the age column, I handled missing value by filling them with the mean age of the available data.

The sex column, containing categorical data, was appropriately encoded into binary values for compatibility with the decision tree algorithm. I divided data into training and testing subsets, as step in assessing the predictive performance of the decision tree model.

The decision tree was instantiated and trained on the training data using the DecisionTreeClassifier from sklearn. To gain insights into the structure of the decision tree and its decision-making process, I visualized it using the plot_tree function from the matplotlib library. The resulting plot illustrates the hierarchical structure of the decision tree, focusing on the sex, age, and pclass columns.

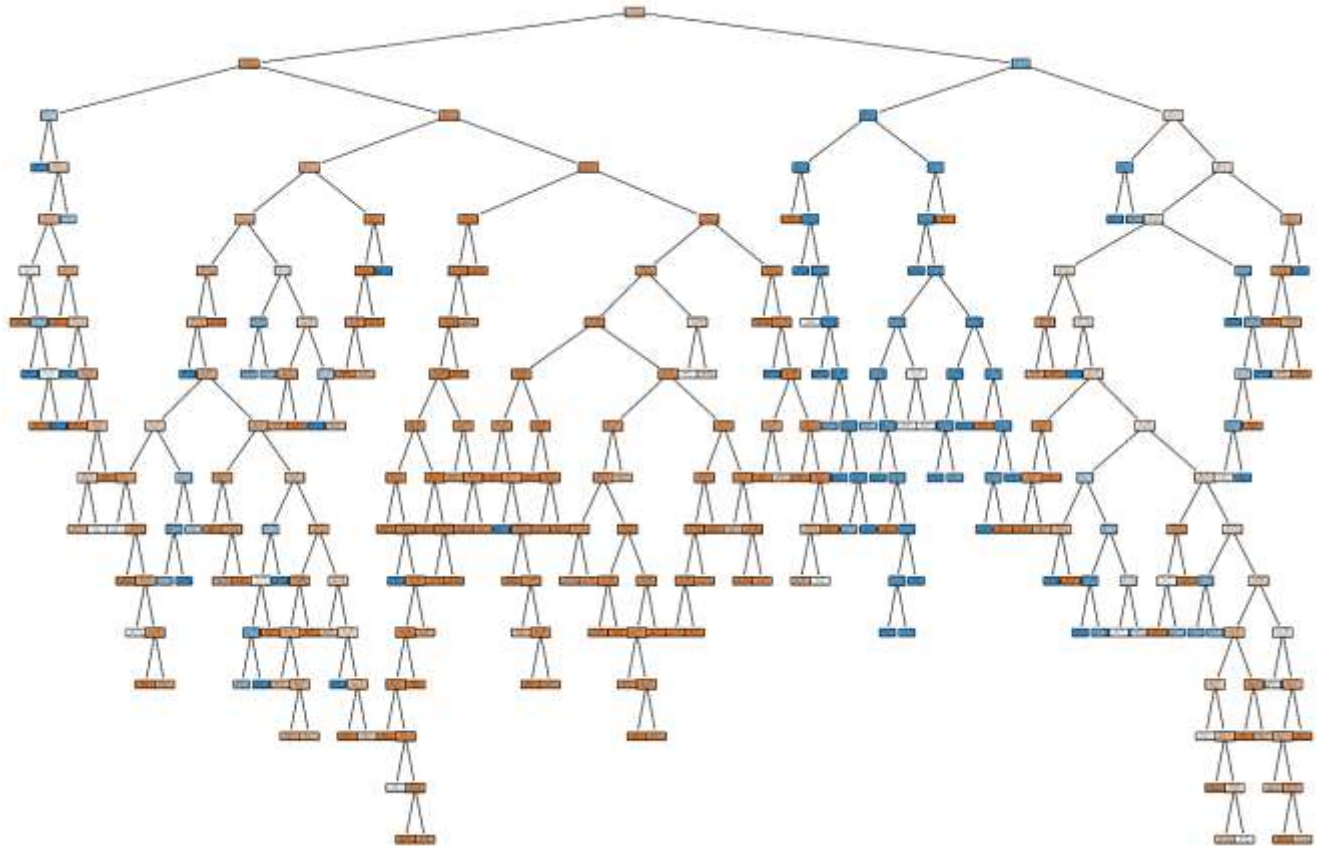


Figure 2 Decision tree

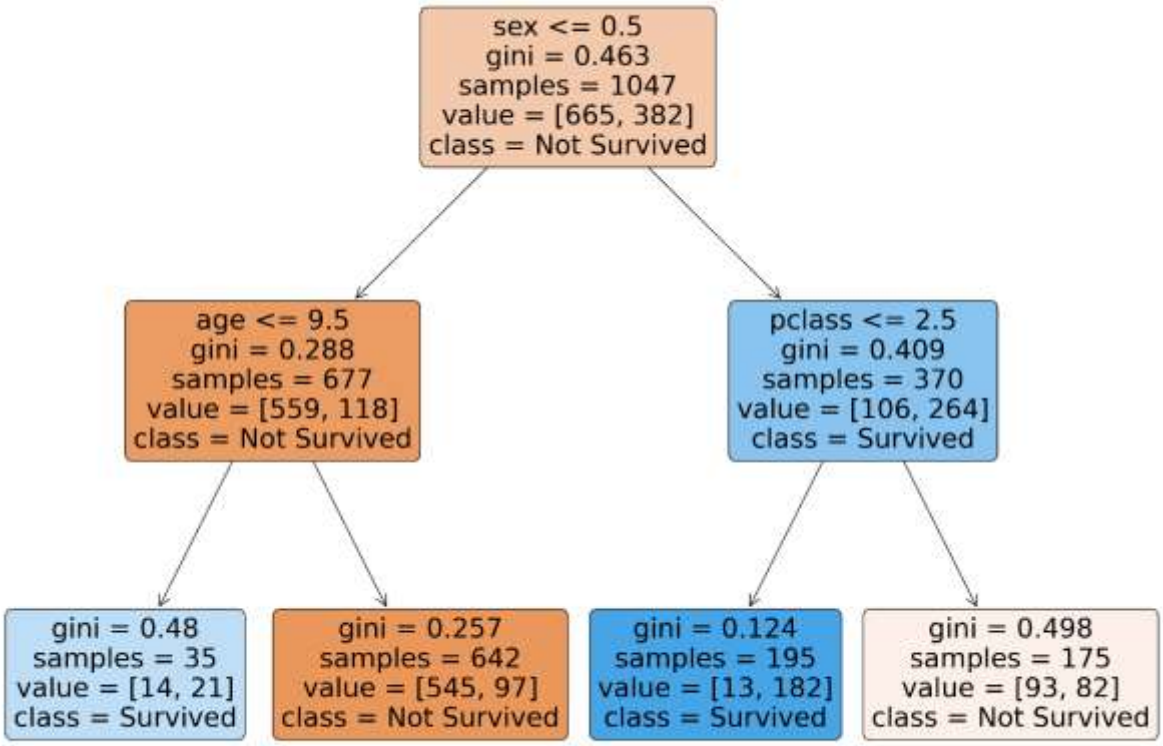
Plot the decision tree for sex, age, pclass column. Shows the node and branches helps to show and interpreting the titanic dataset.

2.4 Evaluate the performance of the tree (before and after pruning) and provide results using cross-validation.

I utilized the scikit-learn library to perform the task of assessing the performance of a Decision Tree classifier on the Titanic dataset. The initial step involved creating an instance of the DecisionTreeClassifier with a fixed random state (42). The algorithm's performance was evaluated before pruning through cross-validation scores and the calculation of the misclassification error, providing valuable insights into potential imbalance within the model. After obtaining the initial performance metrics, we proceeded to prune the Decision Tree using cost-complexity pruning, setting the alpha parameter to 0.01. The subsequent evaluation of the pruned Decision Tree involved cross-validation to gauge its accuracy. The results, including cross-validation accuracy scores and the misclassification error after pruning, were then compared with the initial metrics.

The Decision Tree was visualized post-pruning using the **plot_tree** function from scikit-learn and displayed using Matplotlib. This visualization allowed for a clear understanding of the refined structure of the pruned model.

```
Cross Validation Performance Before running

---------------------------------------------
Cross validation scores before pruning:[0.50763359 0.75954198 0.67557252 0.66412214 0.63601533]

Mean accuracy before pruning: 0.6485771109356263

Misclassification error before pruning:0.2595419847328244

Cross Validation Performance After Prunning

---------------------------------------------
Cross validation scores after pruning: [0.51526718 0.80152672 0.79770992 0.67938931 0.60536398]

Mean accuracy after pruning: 0.6798514228890644

Misclassification error after pruning:, 0.27099236641221375

Accuracy of the model reduced.

                        DECISION TREE AFTER PRUNNING
```



The results indicate that the Mean Accuracy before pruning is approximately 0.648, accompanied by a misclassification error of 0.259. After applying pruning to the Decision Tree, the accuracy increased to 0.679, but the misclassification error also rose to 0.27. This suggests that the pruning process might have led to a trade-off, where the overall accuracy improved slightly, but at the expense of an increase in misclassification errors. It's important to consider the specific goals and trade-offs in model performance when deciding whether pruning is beneficial in this context

2.5 Compare the final tree with logistic regression and comment on the advantages and disadvantages of both. Which model is best for competing in the Kaggle competition?

Logistic Regression model was built and evaluated through cross-validation, serving as a benchmark against the Decision Tree model's performance. The Logistic Regression model demonstrated a mean accuracy of approximately 0.703 and a misclassification error of around 0.225, showcasing its competence in classifying instances within the Titanic dataset. Subsequently, the Decision Tree model, pruned for enhanced simplicity, exhibited a slightly higher misclassification error of approximately 0.271. The comparison revealed that, with respect to misclassification error, Logistic Regression outperformed the pruned Decision Tree, indicating that, in terms of error rates, Logistic Regression is more favourable choice for this specific task, As it indicated with Higher accuracy compare to Decision tree

```
Cross-validation scores: [0.52290076 0.82442748 0.80152672 0.70610687 0.65900383]

Mean accuracy: 0.7027931326957386

Logistc Regression Misclassification error: 0.22519083969465647

Misclassification error after pruning Decision tree:, 0.27099236641221375

Logistic regression has better performance compare to pruned decision tree considering  error.
```

When to use Decision tree over logistic regression[24]

a. Decision tree is better for handling complex and non-linear relationship
b. Decision tree can be more used for data with outlier while logistic cannot.
c. Can be used when the target variable has more than two classes.

When to use Logistic regression over Decision tree[24]

a. For predicting binary outcome logistic regression is the best model compared to decision tree
b. Logistic regression is used for linear relationship between predictor and predict.
c. For small sample size logistic regression is better than decision tree

3. Classification using KNN

3.1 By focusing on small neighbourhoods of state space it is possible to construct parsimonious models. Describe the concept behind this general approach and a step-by-step procedure for implementing such a model.

Parsimonious models aim for simplicity, reflecting the principle of Occam's razor, which suggests that, when confronted with two explanations, the simpler one is often more accurate. In the context of statistical models, the principle of parsimony strives to elucidate data using the minimal number of parameters, offering significant advantages.[25],[26],[27]

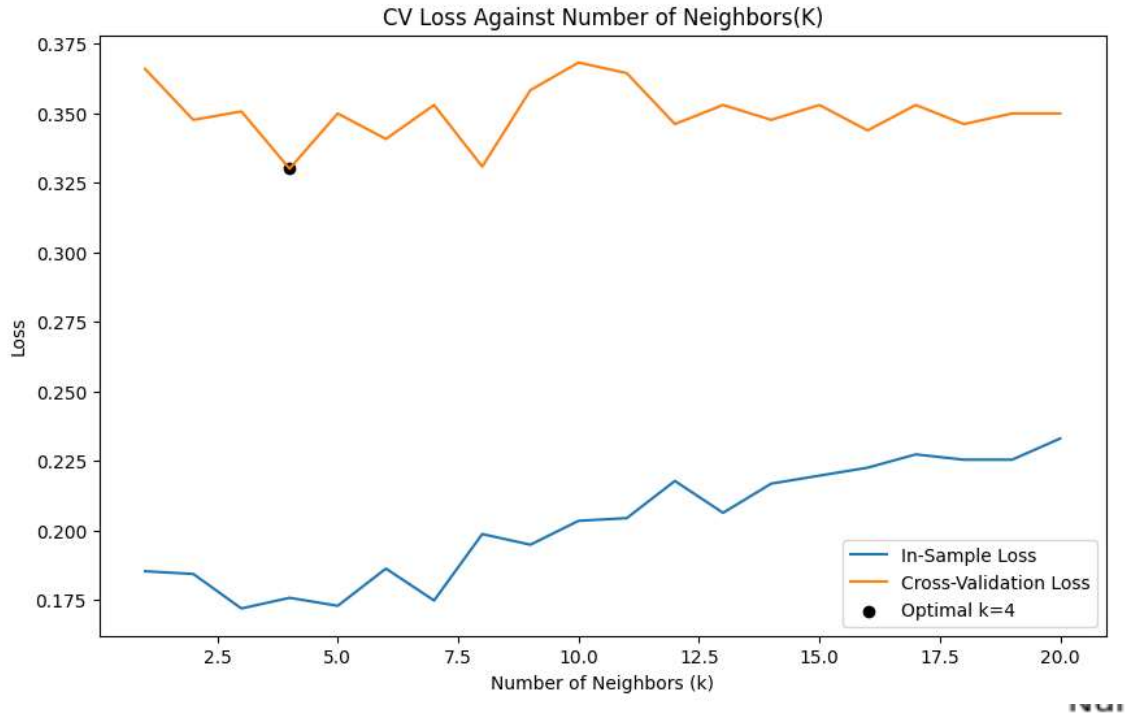
- Define the scope and purpose: The scope and purpose of a model involve clearly defining its intended use and addressing a specific problem or phenomenon. This includes setting boundaries, specifying key variables, and outlining relationships to ensure a focused and effective application of the model.
- Define state space: Define the state space as the encompassing set of potential values for the relevant variables in the problem, capturing the entire spectrum of possible states or conditions.
- **Identify Neighbourhoods:** Identifying neighbourhoods involves categorizing data subsets with similar properties based on characteristics, clustering, and domain knowledge to facilitate the construction of localized models in a parsimonious modelling approach.
- **Choose Model**: Select appropriate models for each neighbourhood based on the characteristics of the data, considering options like linear models, decision trees, or other algorithms depending on the specific nature of the data.
- **Validate and Refine Model:** Validate and refine the parsimonious models by employing validation techniques like cross-validation or comparing their performance against independent datasets, iteratively improving accuracy and predictive power through necessary model adjustments.
- Iterate and improve the model development process by continuously refining the model in response to new data, emerging insights, or evolving dynamics within the problem or system being modelled.

3.2 Explain how you will transform the available variables in order to construct a KNN classifier

- Choose the value K for the neighbours
- Compute the Euclidean distance for K neighbours.
- Select the K nearest neighbours based on the computed Euclidean distance.
- Count the occurrences of data points in each category among these K neighbours.
- Assign the new data point to the category for which the count of neighbours is the highest.

3.2 Calculate the performance of the classifier versus the number of neighbours used and provide a graphic to display the result. What is the optimal number of neighbours using cross-validation?

The task is to use k-Nearest Neighbours (KNN) classifier to predict survival on the Titanic dataset. I started by cleaning and pre-processing, including encoding the sex column into binary values and handling missing values in the age column by filling with mean. The KNN classifier was initially trained with default parameters and evaluated for in-sample and cross-validation losses. Subsequently, a fine-tuning process was conducted, exploring various values for the number of neighbours (k). The optimal number of neighbours was identified based on the k value that minimized the cross-validation loss. A visual representation of in-sample and cross-validation losses across different k values was plotted to facilitate a comprehensive understanding of the model's performance characteristics.




```
The optimal number of neighbors is: 4
```

I utilized the Matplotlib library to graphically represent the losses associated with different numbers of neighbours in the k-Nearest Neighbours (KNN) classifier. The plot spanned a range of 1 to 20 neighbours, providing a visual understanding of how the model's in-sample and cross-validation losses evolved with varying complexities. Significantly, the visualization showcased a distinct point at which the cross-validation loss reached its minimum, pinpointing the optimal number of neighbours as 4 within the explored range. This graphical approach not only facilitated a nuanced interpretation of the model's performance across different hyperparameter values but also emphasized the importance of selecting an optimal k to strike a balance between in-sample and cross-validation losses, ensuring effective generalization to new data.

3.3 Evaluate the performance of the KNN classifier using different distance metrics

I assessed the performance of the k-Nearest Neighbors (KNN) classifier using various distance metrics, such as Euclidean, Manhattan, Chebyshev, minkowski, and hamming. For each specified metric, the KNN classifier was trained on the training data, and its accuracy was evaluated on the testing set. The results were stored in a dictionary, I created for loop to associate each distance metric with its corresponding accuracy score.

```
The performance distance Metric for KNN classifier

The performance Accuracy of euclidean distance Metric: 0.7442748091603053
The performance Accuracy of manhattan distance Metric: 0.7480916030534351
The performance Accuracy of chebyshev distance Metric: 0.683206106870229
The performance Accuracy of minkowski distance Metric: 0.7442748091603053
The performance Accuracy of hamming distance Metric: 0.7519083969465649
```

The results revealed distinct performance outcomes, with the hamming metric achieving the highest accuracy at approximately 0.752, surpassing the accuracies obtained with other metrics such as Euclidean 0.744, Manhattan 0.748, minkowski 0.744, and Chebyshev 0.683. This analysis underscores the significance of selecting an appropriate distance metric tailored to the dataset's characteristics, as it profoundly influences the KNN classifier's efficacy in accurately categorizing instances.

Through this performance can help to analyse and choose the better distance metric to use when testing the algorithm, where this 5 are the most popular distant metric.

Machine learning algorithms, whether supervised or unsupervised, leverage distance metrics to discern patterns in input data for informed decision-making. An effective distance metric plays a crucial role in enhancing the performance of classification, clustering, and information retrieval processes.[28]

- Euclidean distance is less effective for higher-dimensional data beyond 2D, or 3D space, and without feature normalization, unit variations can distort distance calculations.
- The Manhattan distance has drawbacks: it lacks intuition in high-dimensional space and doesn't represent the shortest path.
- Minkowski distance, covering diverse measures, faces challenges in higher-dimensional space and unit dependency. Additionally, the flexibility of the p-value may lead to computational inefficiency in finding the optimal value.
- Hamming distance limitations include its restriction to comparing vectors of the same length and its inability to convey the magnitude of differences, making it unsuitable when magnitude matters.

3.5 Compare best KNN with Logistic regression

In evaluating the performance of both the k-Nearest Neighbors (KNN) classifier and Logistic Regression through cross-validation, the KNN model with the optimal number of neighbors (k = 4) achieved a cross-validation accuracy of approximately 0.769. Meanwhile, the cross-validation accuracy for Logistic Regression was slightly higher, reaching approximately 0.790.

```
Cross-Validation Accuracy for KNN with optimal k (4): 0.7688539530644795
Cross-Validation Accuracy for Logistic Regression: 0.7898382319434951
```

This result shows that the Logistic regression has high chance of giving the accurate result than KNN which is the one that can be used for Kaggle challenge.
Compare the Best KNN classifier with logistic regression[29]

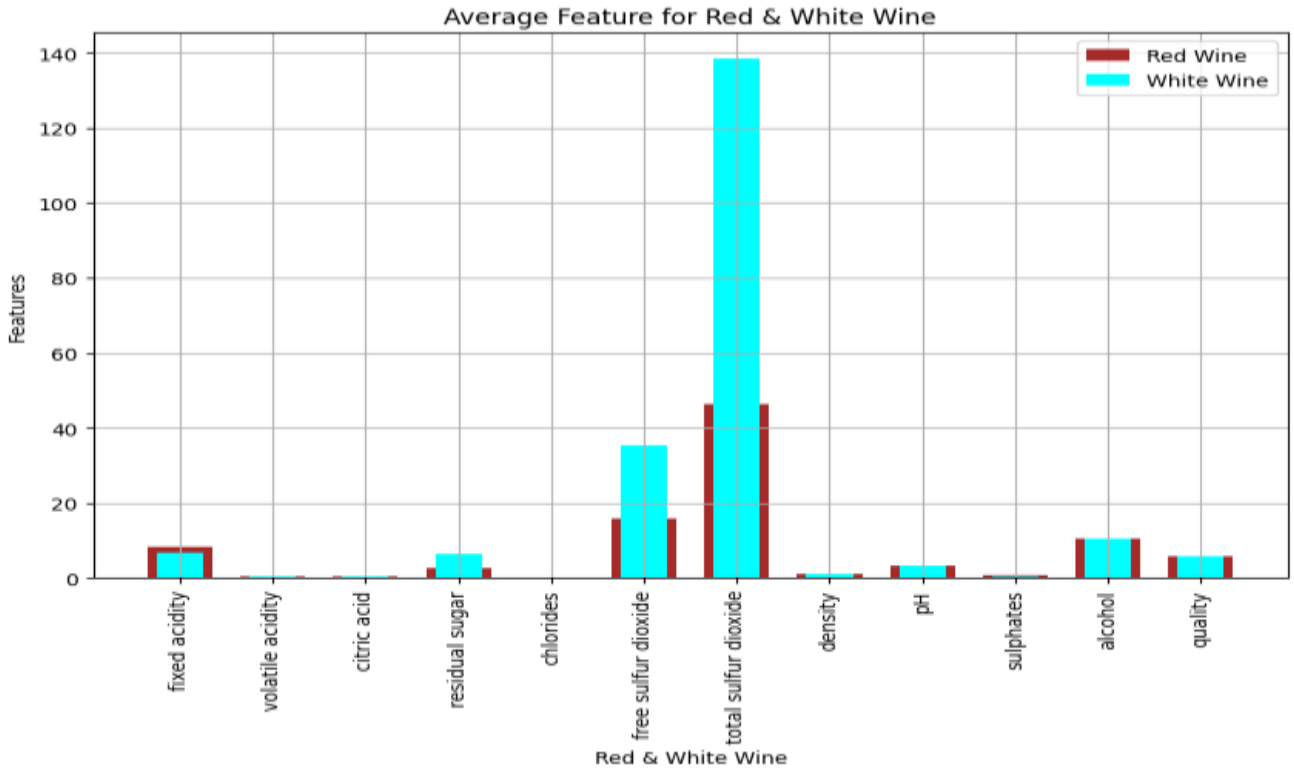
a. Logistic Regression is a classification algorithm based on maximum likelihood estimation.
b. It excels in handling linear data and is straightforward to implement. And it is faster, highly scalable

1. The KNN algorithm, characterized as a lazy learning approach, makes predictions during testing, and only retains data during training.
2. KNN is better when decision boundary is irregular.
3. Outliers significantly impact the performance of the KNN algorithm, emphasizing the importance of addressing outliers before training on the data.

4. Wine Quality Regression

4.1 Calculate the average of each feature for the red and white wines separately using mean() function. Plot bar graph to show comparison. Infer on the results.

I started this question by suing pandas library to extract the dataset then compute the mean values for each feature in both datasets as it was requested to compare their mean. I proceeded with creating a bar graph using Matplotlib library, showing the average feature values for red and white wines. Red wine features are represented by brown bars, while white wine features are depicted in cyan



Based on the graph the peak mean is visualized on Total sulphur dioxide for both red and white wine and White wine is the one with high average in all feature plotted.

4.2 Calculate correlation of these features with the dependent variable and identify the most relevant feature based on the correlation values.

I began the task by determining the correlation between each feature and dependent variable for red wine dataset and for white wine dataset separately as instructed. For red wine, after extracting the independent features and the dependent variable. I computed the correlation matrix. Subsequently, I focused on the correlation between each feature and the red wine dependent variable, with the results revealing that the most influential variable is ['alcohol'] positively correlated. In the case of white wine, a parallel procedure was executed, extracting independent features and the dependent variable. The correlation matrix was then computed, highlighting the variable as the most relevant, displaying a positive correlation.

```
Correlation Between Red Wine Features Dependent Variable:

alcohol                1.000000
pH                     0.205633
citric acid            0.109903
sulphates              0.093595
residual sugar         0.042075
fixed acidity         -0.061668
free sulfur dioxide   -0.069408
volatile acidity      -0.202288
total sulfur dioxide  -0.205654
chlorides             -0.221141
density               -0.496180
Name: alcohol, dtype: float64


The Correlation between White Wine Features Dependent Variable:

alcohol                1.000000
pH                     0.121432
volatile acidity       0.067718
sulphates             -0.017433
citric acid           -0.075729
fixed acidity         -0.120881
free sulfur dioxide   -0.250104
chlorides             -0.360189
total sulfur dioxide  -0.448892
residual sugar        -0.450631
density               -0.780138
Name: alcohol, dtype: float64
```

As shown in the figure in ascending order for red wine alcohol exhibit strong correlation with dependent variable with correlation of 1.00 and followed by Ph, where density does not have relationship with dependent variable quality with negative correlation of -0.49.

While for white wine Alcohol has relationship with dependent variable with correlation of 1.00 which is strong relation, and density also come as the last with negative correlation of -0.78.

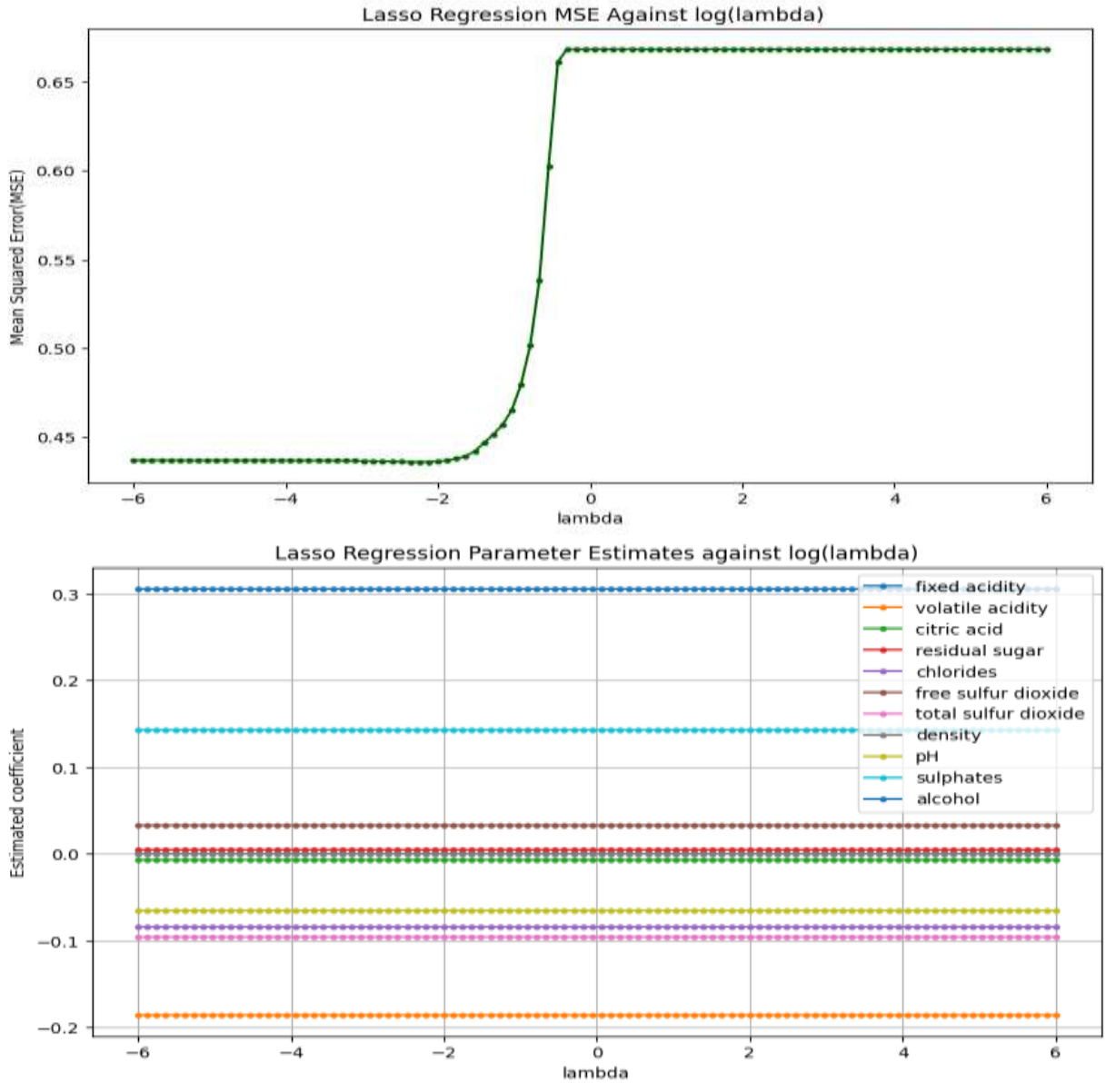
The Positive correlation shows that there is relationship between features and dependent variable where change of one variable led to the change of other variable.

4.3 Use Lasso and cross validation to provide a plot of MSE for each wine type. Provide a plot of parameter estimates versus lambda

I applied Lasso regression with cross-validation to the scaled features of the red wine dataset. The regularization parameter (lambda) was varied across a range of values, and the Mean Squared Error (MSE) was plotted against the logarithm of lambda. The resulting graph provides insight into the trade-off between model complexity and accuracy. Additionally, another plot illustrates how the estimated coefficients of each feature change with varying lambda values, offering a visualization of feature selection through Lasso regularization.

Furthermore, I compared the features selected by Lasso with an alternative approach based on setting a threshold on the absolute correlation coefficient. The correlation coefficient threshold was defined as 0.3, and features with correlation coefficients above this threshold were selected. The final comparison highlights the features chosen by each method, shedding light on the similarities and differences in feature selection between Lasso regression and the correlation coefficient threshold approach. This was for Red wine and I performed the similar task for white wine dataset

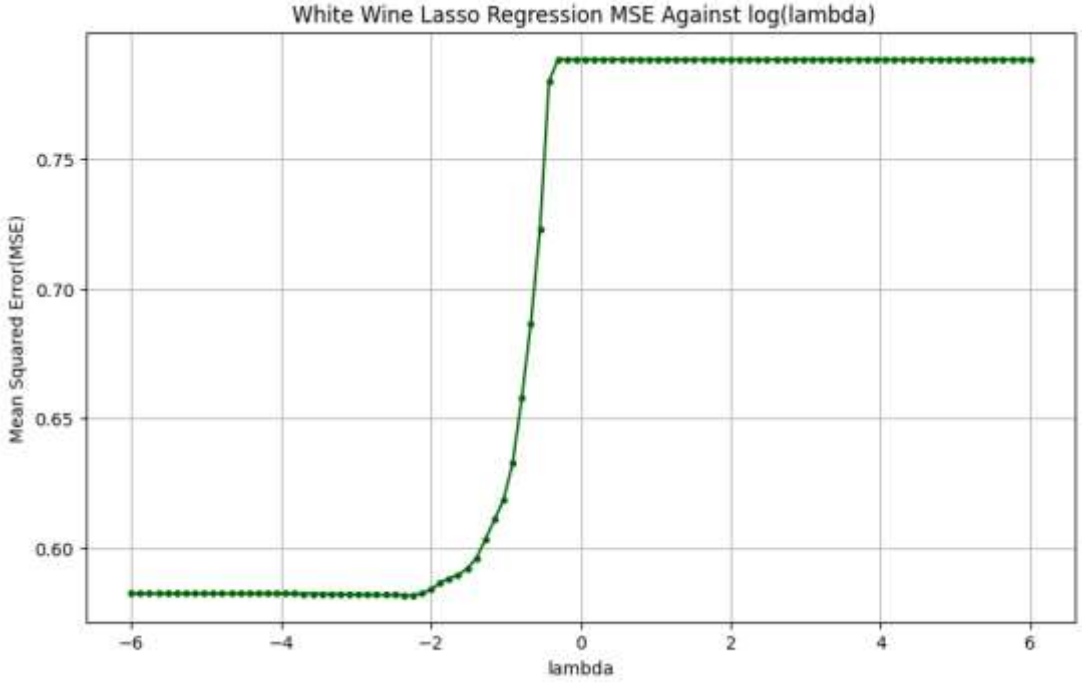
Graph of Red Wine

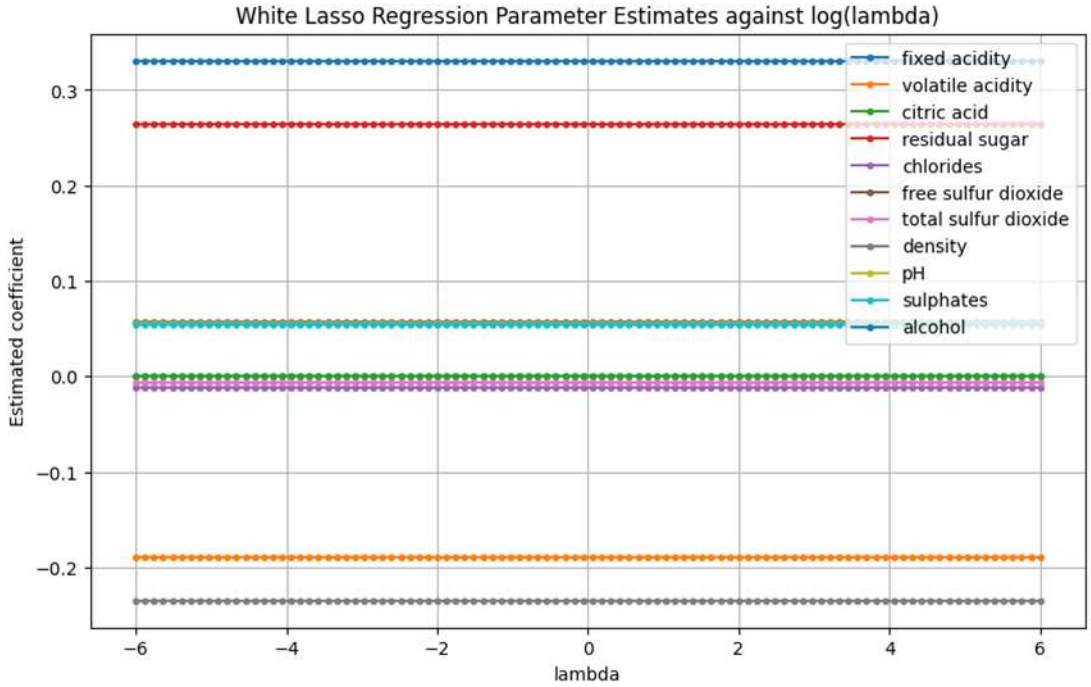


```
Features selected by Lasso:
----------------------------
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
pH
sulphates
alcohol

Features selected by correlation threshold:
----------------------------------------
volatile acidity
alcohol
```

Lasso regression with cross-validation, the features selected for the red wine dataset were found to be: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, pH, sulphates, and alcohol. These features were identified based on the non-zero coefficients obtained during the Lasso regularization process.

Comparing to when using a correlation coefficient threshold approach with a threshold set at 0.3 which is adjustable, only volatile acidity and alcohol were selected as relevant features for the red wine dataset. This contrast in selected features highlights the differing criteria and sensitivities of Lasso regularization and the correlation coefficient threshold approach in feature selection. While Lasso considers the overall contribution of features to the model's performance, the correlation threshold approach focuses solely on the strength of the linear relationship between each feature and the dependent variable.






```
Features selected by Lasso:
------------------------------
volatile acidity
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol

Features selected by correlation threshold:
------------------------------------------
density
alcohol
```

For white wine dataset selecting features using Lasso regularization with cross-validation and the correlation coefficient threshold approach led to distinctive sets of selected features. Through Lasso regularization, a broader array of features was identified, as shown in graph above it encompassing volatile acidity, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol. Conversely, the correlation coefficient threshold approach, employing a threshold of 0.3, pinpointed density and alcohol as the sole selected features. This divergence in feature selection emphasizes the nuanced differences in the criteria employed by Lasso regularization and correlation-based thresholding, highlighting the adaptability of Lasso in capturing a more extensive set of influential features for predicting white wine characteristics

4.4 Use the features identified by LASSO to construct a KNN regression model for red wine. By employing the features selected using LASSO to construct the model separately for red wine dataset and for white wine dataset but the approach remain the same.

In constructing a KNN regression model for red wine, the features identified by Lasso were utilized. The selected features from Lasso regularization were extracted and assigned to the variable X_lasso. Subsequently, the dataset was split into training and testing sets using the train_test_split function, with 0.2 of the data testing. The features were then standardized using the StandardScaler to ensure consistent scaling across the dataset.

To construct the KNN regression model, the value of k (number of neighbors) was set to 5 and the KNeighborsRegressor was instantiated and fitted to the standardized training data. Predictions were made on the test set using the predict method, and the model's performance was evaluated by calculating the Mean Squared Error (MSE) between the predicted and actual values. The resulting MSE provides a quantitative measure of the model's accuracy in predicting the red wine quality.

Red wine

```
Mean Squared Error (MSE) on the test set: 0.41937499999999994
```

For White wine

```
Mean Squared Error (MSE) on the test set: 0.48804081632653057
```

The Mean Squared Error (MSE) on the test set for the KNN regression model constructed using the features identified by Lasso for red wine is approximately 0.4194. This value represents the average squared difference between the predicted and actual values in the test set, providing a quantitative measure of the model's accuracy. This is for Red wine. For white wine the Mean squared error for test set using lasso selected feature to predict using KNN is 0.488 which is high error for prediction. This indicate that the KNN is not good in predicting the quality of the wine.

4.5 What is the performance of a linear regression model and the KNN model, measured by MSE and R-squared

In this code, I constructed and evaluated two regression models, linear regression, and k-nearest neighbors (KNN) regression, for predicting the quality of red wine. Utilizing the features identified by Lasso regularization, I first standardized the data and split it into training and testing sets. For linear regression, I fitted the model to the standardized training data, made predictions on the test set, and calculated the Mean Squared Error (MSE) and R-squared (R^2) to gauge its performance. Simultaneously, I implemented the KNN regression model with k=5 neighbors, predicting the test set and computing MSE and R^2. The resulting metrics revealed that the linear regression model, which achieved a lower MSE, demonstrated superior predictive accuracy compared to the KNN regression model.
For Red Wine comparison

```
Red Wine Linear Regression Performance:
------------------------------
Mean Squared Error (MSE): 0.5714404831038473
R-squared: 0.26215583678908017
Red Wine KNN Regression Performance:
------------------------------
Mean Squared Error (MSE): 0.48804081632653057
R-squared: 0.3698415173889896
```

For white wine comparison

```
Mean Squared Error (MSE): 0.5714404831038473
R^2: 0.26215583678908017
Mean Squared Error (MSE): 0.48804081632653057
R^2: 0.3698415173889896
```

In the evaluation of red wine and white wine separately prediction models, the linear regression model and k-nearest neighbors (KNN) regression model were assessed using the features identified by Lasso. The linear regression model exhibited a mean squared error (MSE) of 0.5714 for red wine, 0.571 for white and an R-squared of 0.2622, indicating moderate predictive accuracy and explanatory power. On the other hand, the KNN regression model demonstrated a lower MSE of 0.4880 for both white and red, suggesting improved predictive accuracy compared to linear regression. The KNN model also displayed a higher R-squared of 0.3698, indicating a better fit to the data and a more substantial proportion of variance explained. These results highlight the comparative performance of the two models, with the KNN regression model showing better predictive accuracy and a stronger ability to capture the underlying patterns in red wine quality.

Describe the advantages and disadvantages of both models.

Linear regression[30]

Advantage

- Straightforward model: The linear regression model represents the most basic equation for expressing the relationship between multiple predictor variables and the predicted variable.
- Computationally efficient: Linear regression is characterized by high modeling speed since it avoids intricate calculations, making it particularly fast when dealing with large datasets for predictions.

Disadvantage

- The linear regression model is overly simplistic and fails to capture the complexity inherent in real-world scenarios.
- Linear regression relies on strong assumptions, including the assumption that the predictor (independent) and predicted (dependent) variables are linearly related, a condition that may not hold true in all cases.

KNN[31]

Advantage

- No Training Required: KNN modeling skips the training phase as the data itself serves as the model, acting as a reference for future predictions. This approach is highly time-efficient for adapting to random modeling on the existing dataset.

- Simple Implementation: KNN is straightforward to implement, as the only calculation involved is determining the distance between various points based on the data from different features. This distance can be easily computed using formulas like the Euclidean or Manhattan distance.

Disadvantage

- Performance Issues with Large Datasets: KNN does not perform well with large datasets because calculating distances between each data instance becomes computationally expensive and resource-intensive.
- Sensitivity to Noisy and Missing Data: KNN is sensitive to noisy and missing data, and its performance can be adversely affected in the presence of such data issues.

Reference

[1] "What Is Nonlinear? Definition, Vs. Linear, and Analysis," Investopedia. Accessed: Nov. 09, 2023. [Online]. Available: https://www.investopedia.com/terms/n/nonlinearity.asp
[2] "(27) Machine Learning: Linearity vs Nonlinearity | LinkedIn." Accessed: Nov. 09, 2023. [Online]. Available: https://www.linkedin.com/pulse/machine-learning-linearity-vs-nonlinearity-reday-zarra/?trackingId=gcqvTLmJQb6r6baGs8TpOg%3D%3D
[3] "Nonlinear Regression: Formula & Examples | StudySmarter," StudySmarter UK. Accessed: Nov. 09, 2023. [Online]. Available: https://www.studysmarter.co.uk/explanations/engineering/engineering-mathematics/nonlinear-regression/
[4] "Nonlinear Regression: Formula & Examples | StudySmarter," StudySmarter UK. Accessed: Nov. 09, 2023. [Online]. Available: https://www.studysmarter.co.uk/explanations/engineering/engineering-mathematics/nonlinear-regression/
[5] "What Is Nonlinear Regression? Comparison to Linear Regression," Investopedia. Accessed: Nov. 09, 2023. [Online]. Available: https://www.investopedia.com/terms/n/nonlinear-regression.asp
[6] S. V. Archontoulis and F. E. Miguez, "Nonlinear Regression Models and Applications in Agricultural Research," *Agron. J.*, vol. 107, no. 2, pp. 786–798, 2015, doi: 10.2134/agronj2012.0506.
[7] G. Lancaster, D. Iatsenko, A. Pidde, V. Ticcinelli, and A. Stefanovska, "Surrogate data for hypothesis testing of physical systems," *Phys. Rep.*, vol. 748, pp. 1–60, Jul. 2018, doi: 10.1016/j.physrep.2018.06.001.
[8] "CMUdiaml9.pdf: Data, Inference, and Applied Machine Learning - DIAML." Accessed: Nov. 12, 2023. [Online]. Available: https://canvas.cmu.edu/courses/36029/files/10293443?module_item_id=5635264
[9] "Shuffle-based · TimeseriesSurrogates.jl." Accessed: Nov. 12, 2023. [Online]. Available: https://juliadynamics.github.io/TimeseriesSurrogates.jl/dev/methods/randomshuffle/
[10] J. Brownlee, "A Gentle Introduction to Information Entropy," MachineLearningMastery.com. Accessed: Nov. 13, 2023. [Online]. Available: https://machinelearningmastery.com/what-is-information-entropy/
[11] Sole, "Mutual information with Python," Train in Data Blog. Accessed: Nov. 13, 2023. [Online]. Available: https://www.blog.trainindata.com/mutual-information-with-python/
[12] "13.13: Correlation and Mutual Information," Engineering LibreTexts. Accessed: Nov. 15, 2023. [Online]. Available: https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls_(Woolf)/13%3A_Statistics_and_Probability_Background/13.13%3A_Correlation_and_Mutual_Information
[13] P. Laarne, M. A. Zaidan, and T. Nieminen, "ennemi: Non-linear correlation detection with mutual information," *SoftwareX*, vol. 14, p. 100686, Jun. 2021, doi: 10.1016/j.softx.2021.100686.
[14] S. Asaithambi, "Why, How and When to apply Feature Selection," Medium. Accessed: Nov. 15, 2023. [Online]. Available: https://towardsdatascience.com/why-how-and-when-to-apply-feature-selection-e9c69adfabf2
[15] "What Is a Decision Tree and How Is It Used?" Accessed: Nov. 15, 2023. [Online]. Available: https://careerfoundry.com/en/blog/data-analytics/what-is-a-decision-tree/

[16] "Decision Tree," GeeksforGeeks. Accessed: Nov. 15, 2023. [Online]. Available: https://www.geeksforgeeks.org/decision-tree/
[17] S. Kumar, "3 Techniques to avoid Overfitting of Decision Trees," Medium. Accessed: Nov. 15, 2023. [Online]. Available: https://towardsdatascience.com/3-techniques-to-avoid-overfitting-of-decision-trees-1e7d3d985a09
[18] S. Arora, "Let's Solve Overfitting! Quick Guide to Cost Complexity Pruning of Decision Trees," Analytics Vidhya. Accessed: Nov. 15, 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/10/cost-complexity-pruning-decision-trees/
[19] "Decision Tree - Overview, Decision Types, Applications." Accessed: Nov. 15, 2023. [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/decision-tree/
[20] T. Kliegr and E. Izquierdo, "QCBA: improving rule classifiers learned from quantitative data by recovering information lost by discretisation," *Appl. Intell.*, vol. 53, no. 18, pp. 20797–20827, Sep. 2023, doi: 10.1007/s10489-022-04370-x.
[21] H. Hasanpour, R. Ghavamizadeh Meibodi, and K. Navi, "Improving rule-based classification using Harmony Search," *PeerJ Comput. Sci.*, vol. 5, p. e188, Nov. 2019, doi: 10.7717/peerj-cs.188.
[22] "Text Classification: What it is And Why it Matters," MonkeyLearn. Accessed: Nov. 16, 2023. [Online]. Available: https://monkeylearn.com/text-classification/
[23] V. NarasimhaMurthy, "DATA-DRIVEN APPROACH TO IMAGE CLASSIFICATION," University of Massachusetts Amherst. doi: 10.7275/14208575.
[24] G. Willig, "Decision Tree vs Logistic Regression," Medium. Accessed: Nov. 20, 2023. [Online]. Available: https://gustavwillig.medium.com/decision-tree-vs-logistic-regression-1a40c58307d0
[25] J. Frost, "What is a Parsimonious Model? Benefits and Selecting," Statistics By Jim. Accessed: Nov. 17, 2023. [Online]. Available: https://statisticsbyjim.com/regression/parsimonious-model/
[26] C. F. Daganzo, V. V. Gayah, and E. J. Gonzales, "The potential of parsimonious models for understanding large scale transportation systems and answering big picture questions," *EURO J. Transp. Logist.*, vol. 1, no. 1, pp. 47–65, Jun. 2012, doi: 10.1007/s13676-012-0003-z.
[27] A. Christopher, "K-Nearest Neighbor," The Startup. Accessed: Nov. 20, 2023. [Online]. Available: https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4
[28] J. Dancker, "A brief introduction to Distance Measures," Medium. Accessed: Nov. 20, 2023. [Online]. Available: https://medium.com/mlearning-ai/a-brief-introduction-to-distance-measures-ac89cbd2298
[29] P. email@hotmail.com, "Quality Python articles," PythonKitchen. Accessed: Nov. 20, 2023. [Online]. Available: https://www.pythonkitchen.com/logistic-regression-vs-k-nearest-neighbors-in-machine-learning
[30] Satyavishnumolakala, "Linear Regression -Pros & Cons," Medium. Accessed: Nov. 21, 2023. [Online]. Available: https://medium.com/@satyavishnumolakala/linear-regression-pros-cons-62085314aef0
[31] A. Soni, "Advantages And Disadvantages of KNN," Medium. Accessed: Nov. 21, 2023. [Online]. Available: https://medium.com/@anuuz.soni/advantages-and-disadvantages-of-knn-ee06599b9336