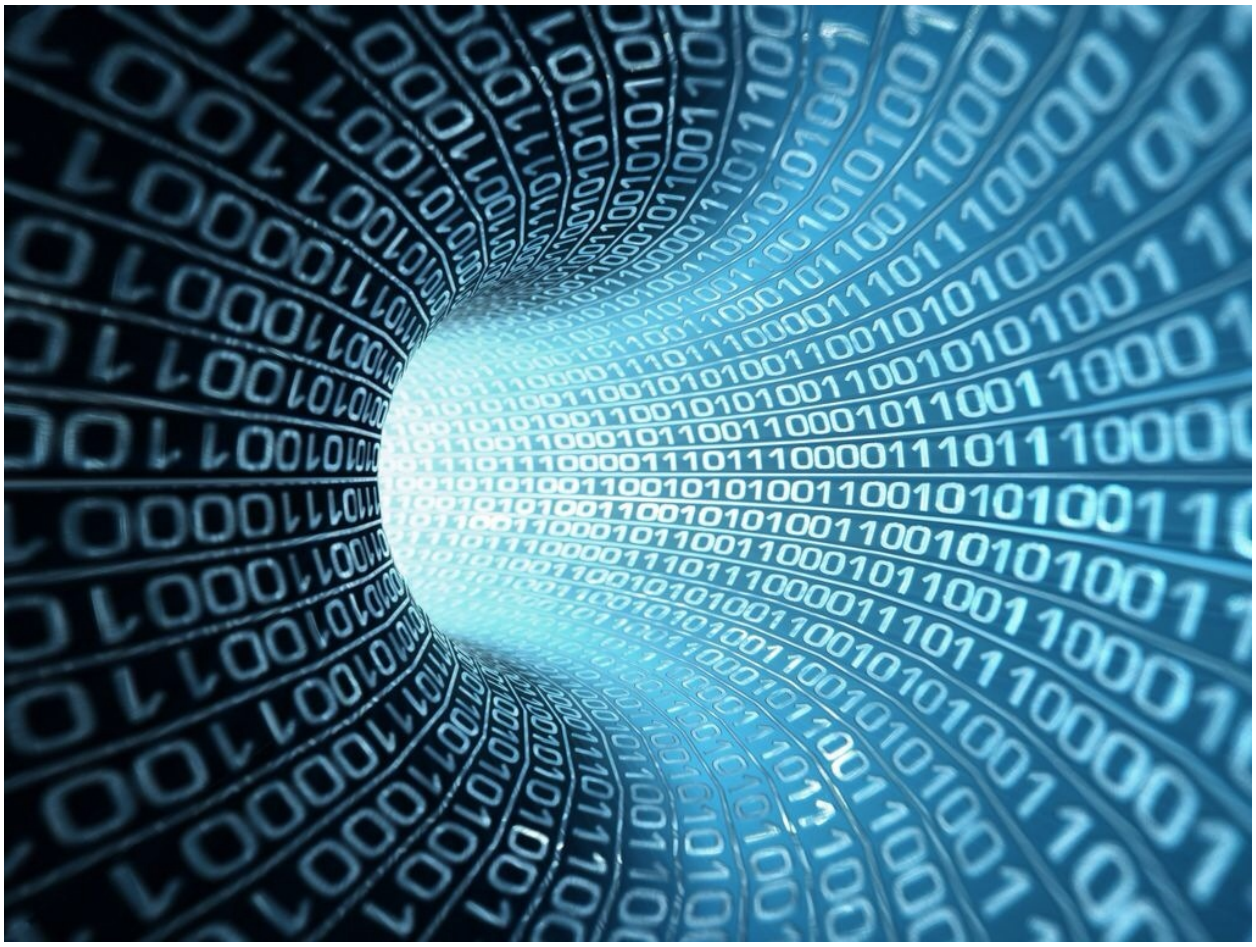
Report: DIAML Assignment 6



I, the undersigned, have read the entire contents of the syllabus for course 18-785 (Data Inference and Applied Machine Learning) and agree with the terms and conditions of participating in this course, including adherence to CMU's AIV policy.

Signature: Tunga Tessema

Andrew ID: tchamiss

Full Name: Tunga Tessema

Librairies

- **Pandas:** to read excel and csv files, convert them to dataframes and perform operations on them
- **numpy:** to create an array, calculate square roots and calculate correlation coefficients
- **matplotlib.pyplot:** to plot graphs
- **statsmodels.api**: to calculate linear and logistic regression models
- **sklearn**: to build decision tree, lasso model and to calculate metrics like accuracy_score

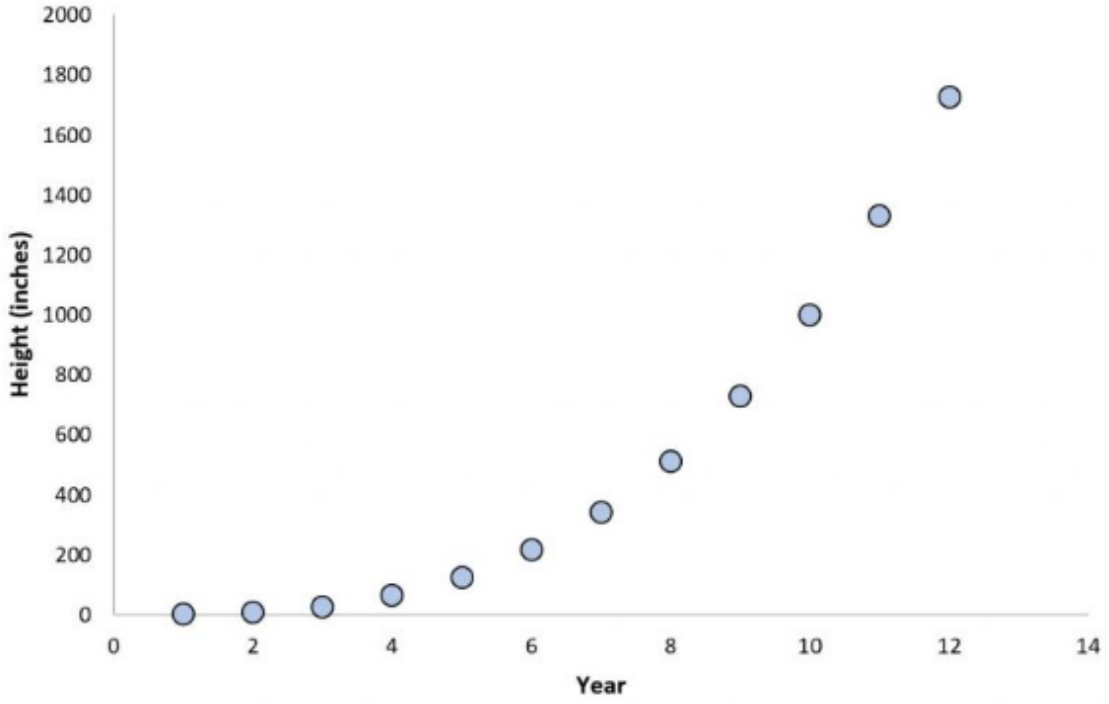
Question 1

1.1 Why might it be necessary to consider nonlinear relationships between variables?

Without studying nonlinear relationships, much of the world around us would not be understood. That is because most physical and statistical situations in the real world are nonlinear. For example, it is often believed that the more effort one puts into working out, the more one should lose weight.

However, this may not hold. Sometimes, doubling your exercise may not lead to an equivalent reducing effect on weight.

1.2 Write down the mathematical equation for a nonlinear model and provide an example of an application where it might be appropriate.



The above graph shows the lifespan of bamboo plants and their yearly growth. It has the following formula: y = x^3. We can see from the graph that their relationship is exponential. During the first few years of growth, a bamboo plant grows very slowly but once it reaches a certain age it explodes in height and grows at a rapid pace.

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.

In statistical modeling, there are occasions where a non-linear model with a limited number of parameters fits the data well, while using a linear model would require many more parameters to produce an acceptable fit. This might happen in a periodic regression model, for example:

$$Y_i = \beta_0 + \alpha \sin(2\pi\phi(x_i - \mu)) + \sigma\varepsilon_i \qquad \varepsilon_i \sim \text{IID } \mathrm{N}(0, 1).$$

The frequency parameter $0<\phi<1$ enters the model in a non-linear fashion (the amplitude and phase angle parameters are linearisable), and no matching linear model reaches the same shape. A linear regression model that approximates the sinusoidal signal is achievable, but it will only be a decent approximation throughout the data range if you have a large number of parameters. For instance, you may choose to represent this signal as a sum of periodic signals with harmonic frequencies, resulting in the linear regression model:

$$Y_i \approx \beta_0 + \sum_{i=1}^{m} \left[\beta_{1s} \sin\left(\frac{2\pi i}{m} x_i\right) + \beta_{1c} \cos\left(\frac{2\pi i}{m} x_i\right) \right] + \sigma\varepsilon_i \qquad \varepsilon_i \sim \text{IID N}(0, 1).$$

In this situation, you have a non-linear model with just four unknown coefficient parameters and a linear approximation with $1+2m$ unknown coefficient parameters. If m is large enough, the linear approximation is less parsimonious since it contains more parameters.

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 provides a means of null hypothesis testing The characteristics that are preserved are unconditional distribution and non-linear correlations.

The following are two surrogate techniques:

- IID surrogates: are independent identically distributed obtained by shuffling the original time series (preserves the unconditional distribution). This technique uses the random shuffle approach: surrogates generated by randomly shuffling the original data; obtaining same distribution and destroying linear correlations.

- FFT surrogates: obtained by Fourier decomposition of the original time series. FFT surrogates preserve the linear correlations (but not the nonlinear correlations). This technique uses the random phases approach: surrogate data are generated by the inverse Fourier Transform of the amplitudes of Fourier Transform of the original data with new (uniformly random) phases. This approach preserves the linear correlations in the data.

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.

The Information Content of a value measures how surprising that value is.
For example, suppose we have a biased coin which 90% of the time turns out to be head, while only 10% tail. If we toss that coin and it results in a head, is that event surprising? Not really. But if it shows a tail? Yes, because that event happens only 10% of the time, it is so unlikely to occur. Things that are surprising give us more information, thus they have higher Information Content.

$$\text{Information Content of x} = -\log p(x)$$

Entropy measures the Uncertainty of data. The intuition is that when the probability of entries are the same (i.e. they are uniformly distributed), the entropy is highest, meaning we know less about the data, or we are uncertain about the data. On the reserve side, when the distribution is skewed (i.e. some entries have significantly bigger probability while some have only small chance to occur), the entropy is low, meaning if we are given a random entry from the data, we are more certain what its value is.

$$H = \sum_{x} -p(x) \log p(x)$$

Approximate entropy (ApEn) is a technique used to quantify the amount of regularity and the unpredictability of fluctuations over time-series data. Standard formula for measuring entropy requires access to large amounts of data and is sensitive to noise. ApEn is a modification that allows estimation of entropy using empirical observations. Approximate Entropy was used to measure regularity in heart rate time series.
Heart rate approximate entropy decreased with age and was higher in women than men ($p < 0.05$).

The Mutual Information of A and B is the properties, or content that both A and B possess.

$$\text{The Mutual Information of A and B is denoted by I(A, B):}$$

$$I(A, B) = H(A) - H(A|B) = H(B) - H(B|A)$$

While (Pearson) correlation is the most commonly used metric to estimate the relationship between variables, it is in fact flawed because it can only recognize linear relationships. The mutual information, on the other hand, is stronger since it does consider every type of dependency.

Question 2

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?

A decision tree is a graphical representation of all possible solutions to a decision.
The components of a decision tree are:-

- **Parent node**: In any two connected nodes, the one which is higher hierarchically, is a parent node.
- **Child node**: In any two connected nodes, the one which is lower hierarchically, is a child node.
- **Root node**: The starting node from which the tree starts, It has only child nodes. The root node does not have a parent node.
- **Leaf Node/leaf**: Nodes at the end of the tree, which do not have any children are leaf nodes or called simply leaf.
- **Internal nodes/nodes**: All the in-between the root node and the leaf nodes are internal nodes or simply called nodes. Internal nodes have both a parent and at least one child.
- **Branch/Subtree**: a subsection of the entire tree is called a branch or sub-tree.

When we remove the sub-node of a decision node, it is called pruning. Pruning is necessary to avoid overfitting. If no limit is set, it will give 100% fitting for the training data, because, in the worst-case scenario, it will end up making a leaf node for each observation. However, it will perform poorly on testing datasets.

Decision trees are an attractive method for classification because they have the following advantages:-

- Easy to visualize and interpret: Its graphical representation is very intuitive to understand and it does not require any knowledge of statistics to interpret it.
- Useful in data exploration: We can easily identify the most significant variable and the relation between variables with a decision tree. It can help us create new variables or put some features in one bucket.

- Less data cleaning required: It is fairly immune to outliers and missing data, hence less data cleaning is needed.
- The data type is not a constraint: It can handle both categorical and numerical data

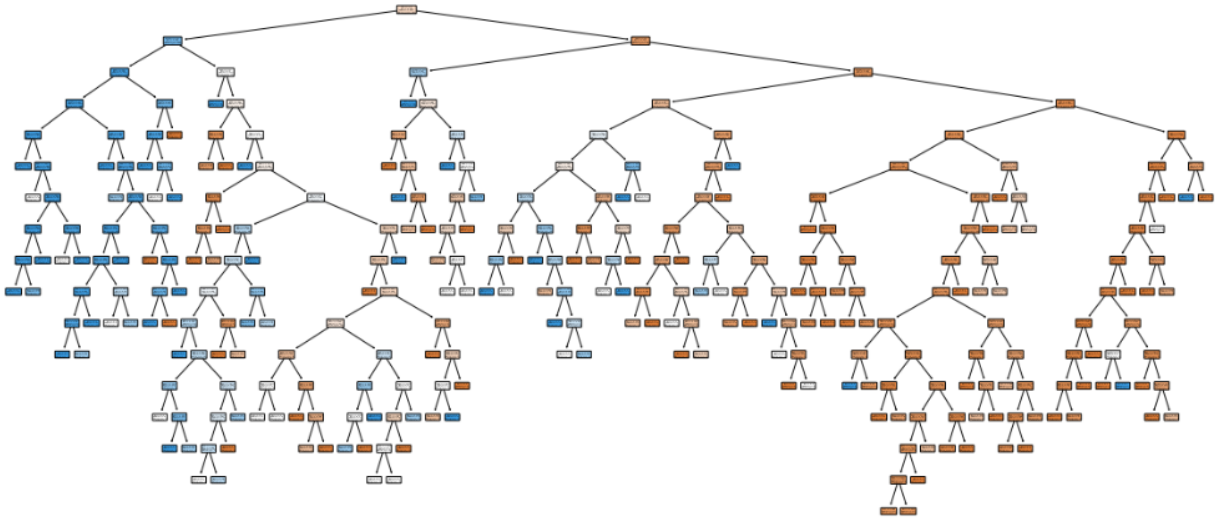
2.2. Suppose an organization has built a rule-based classifier using domain knowledge. After collecting a large amount of data, outline the steps required to improve upon the existing approach by constructing a data-driven classifier. How would you advise to test the validity of the new model?

To create a data classifier:-

- A training dataset must be constructed for which the true classifications are known
- Feature selection: choose only necessary parameters, omit irrelevant parameters and combine highly correlated parameters
- Create a testing data set for which the true classifications are known

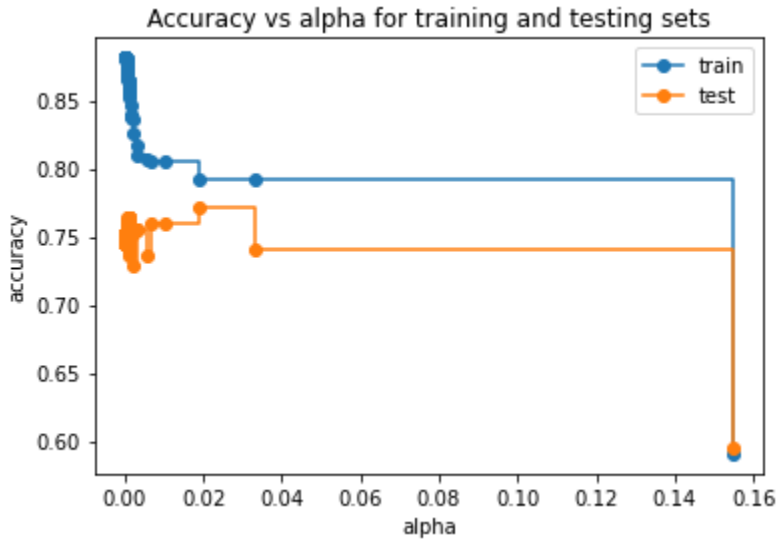
The validity of the new model should be tested using testing data and cross validation. Cross-validation makes sure that all objects in the training set get used both as test objects and as training objects. This ensures that the classifier is tested on both rare and common types of objects.

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.

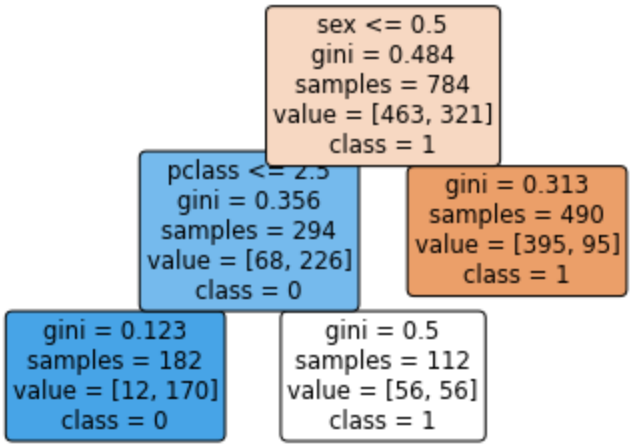


To build a decision tree, I used the DecisionTreeClassifier function. I fitted the model using 75% training data and predicted values using the 25% testing data.
As we can see on the above diagram, the tree has a depth of 15.

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



I pruned the tree using the cost_complexity_pruning_path function. This function gives a list of cost complexity pruning alphas (ccp_alphas). I built a decision tree classifier for each alpha. I then calculated the accuracy for each decision tree classifier for both training and testing data. I plotted the respective accuracy for each alpha value to determine the alpha value with the highest accuracy. I found that alpha 0.03 gives the highest accuracy. I used that alpha value to prune the tree and got the following tree:




```
Accuracy of the model before pruning (train) is 88%
Accuracy of the model before pruning (test) is 74%

Accuracy of the model after pruning (train) is 79%
Accuracy of the model after pruning (test) is 77%
```

As we can see the depth of the tree got reduced from 15 to 2. In addition, even though the accuracy of the training dataset was reduced from 88% to 79%, the accuracy of the testing dataset increased from 74% to 77%. And what counts is how the model performs with new data.

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?

```
Accuracy of the model after pruning (train) is 79%
Accuracy of the model after pruning (test) is 77%

Accuracy of the logistic regression model (train) is 74%
Accuracy of the logistic regression model (test) is 80%
```

For the training dataset, the accuracy of the tree at 79% is greater than the logistic regression at 74%. However, for the testing dataset, the accuracy of the tree is 77% but the accuracy of the logistic regression is 80%. The logistic regression is best for competing in the Kaggle competition.

Logistic Regression assumes that the data is linearly (or curvy linearly) separable in space. Decision Trees are non-linear classifiers; they do not require data to be linearly

separable. When the data set divides into two separable parts, then use a Logistic Regression else go with a Decision Tree. A Decision Tree will take care of both.
In addition, categorical data works well with Decision Trees, while continuous data work well with Logistic Regression. If we enumerate the labels eg. Mumbai — 1, Delhi — 2, Bangalore — 3, Chennai — 4, then the algorithm will think that Chennai (2) is twice as large as Mumbai (1).
Furthermore, Decision Trees handle skewed classes nicely if we let it grow fully. So, there is bias in a dataset, then let the Decision Tree grow fully and identify max depth according to the skew. Logistic Regression does not handle skewed classes well. So we need some adjustments such as increasing the weight to the minority class.
When our data contains outliers we should use trees or remove the outliers for a logistic regression. Logistic regression will push the decision boundary towards the outlier. While a Decision Tree won't be affected by an outlier, since an impure leaf will contain nine +ve and one –ve outlier. The label for the leaf will be +ve, since the majority are positive.
Finally, Logistic Regression does not handle missing values; we need to impute those values by mean, mode, and median. Decision Trees work with missing values.

Question 3

3.1 By focusing on small neighborhoods 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.

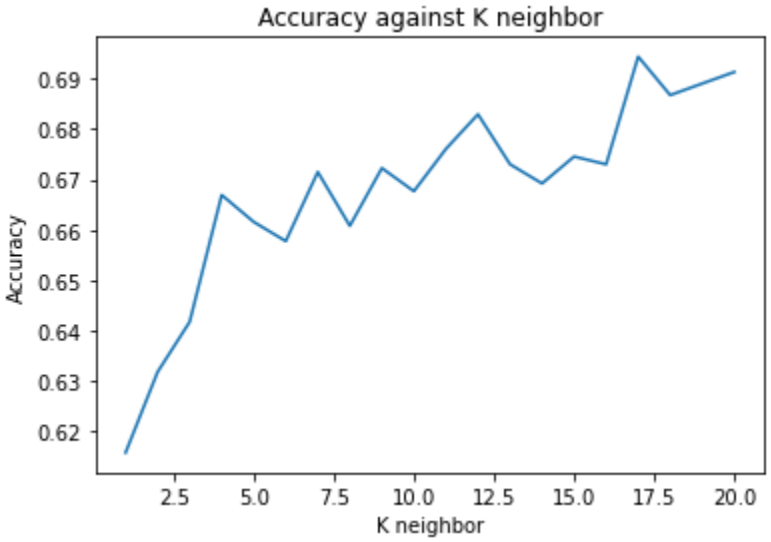
The nearest neighbor approach uses ‘feature similarity’ to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.
The following steps are followed to implement such model:

- the distance between the new point and each training point is calculated.
- The closest k data points are selected (based on the distance)
- The average of these data points is the final prediction for the new point.

3.2 Consider the challenge of classifying the likelihood of survival using the Titanic dataset. In order to construct a KNN classifier, how will you transform the available variables?

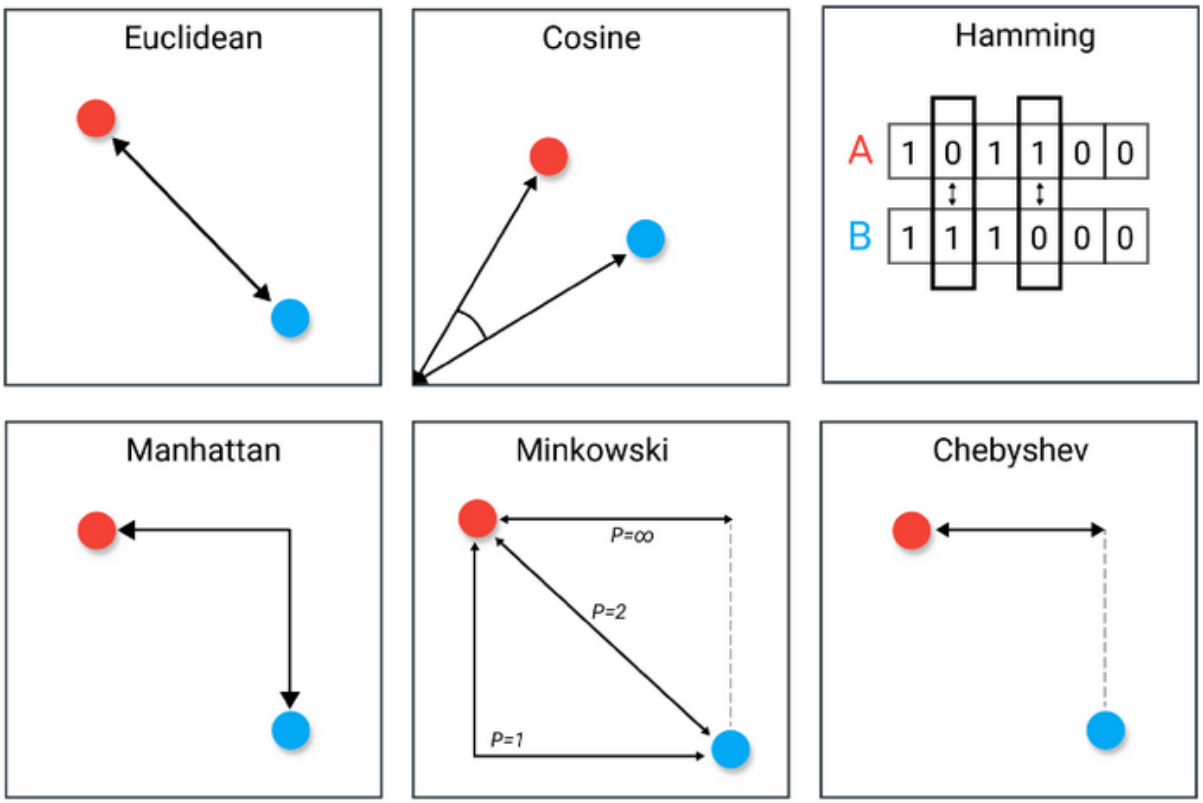
In order to classify the likelihood of survival, I first choose the sex, pclass and age variables. I then changed the sex variable from categorical string to numeric values using pd.get_dummies() function. I also used the min max scaler to normalize the age and pclass values. The scale of the features is very different so normalization is required. This is because the distance calculation done in KNN uses feature values. When one feature values are larger than the others, that feature will dominate the distance hence the outcome of the KNN.

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



I first used the KNeighborsClassifier() function to create the KNN classifier. I used a for loop to pass k values from 1 to 20. For each k value, I used the cross_val_score function to calculate the accuracy. I stored each accuracy and the corresponding k values in their respective arrays. I then plotted the graph using matplotlib.

3.4 Explain why some distance metrics are sensitive to the kind of features used. Evaluate the performance using different distance metrics.



To evaluate the model using different metrics I used the GridSearchCV function and gave it an array of metrics of the above metrics.
Euclidean distance can be explained as the length of a segment connecting two points. Euclidean distance is not scale in-variant which means that distances computed might be skewed depending on the units of the features. Typically, one needs to normalize the data before using this distance measure. Euclidean distance works great when you have low-dimensional data and the magnitude of the vectors is important to be measured.
The cosine distance is simply the cosine of the angle between two vectors.One main disadvantage of cosine similarity is that the magnitude of vectors is not taken into account, merely their direction. In practice, this means that the differences in values are not fully taken into account.We use cosine similarity often when we have high-dimensional data and when the magnitude of the vectors is not of importance.
Hamming distance is the number of values that are different between two vectors. It is typically used to compare two binary strings of equal length. Hamming distance is difficult to use when two vectors are not of equal length. You would want to compare same-length vectors with each other in order to understand which positions do not match.

Moreover, it does not take the actual value into account as long as they are different or equal. Therefore, it is not advised to use this distance measure when the magnitude is an important measure.
Manhattan distance then refers to the distance between two vectors if they could only move right angles. Although Manhattan distance seems to work okay for high-dimensional data, it is a measure that is somewhat less intuitive than euclidean distance, especially when using in high-dimensional data.
Chebyshev distance is defined as the greatest difference between two vectors along any coordinate dimension. It is simply the maximum distance along one axis.
Minkowski is a metric used in n-dimensional real space. Its formula contains a p-value
Common values of p are:

p=1 — Manhattan distance
p=2 — Euclidean distance
p=∞ — Chebyshev distance

Minkowski has the same disadvantages as the distance measures they represent, so a good understanding of metrics like Manhattan, Euclidean, and Chebyshev distance is important. Moreover, the parameter p can actually be troublesome to work with as finding the right value can be quite computationally inefficient. The upside to p is the possibility to iterate over it and find the distance measure that works best for your use case.

3.5 Compare the best KNN classifier with logistic regression and comment on the advantages and disadvantages of both. Which model is best for competing in the Kaggle competition?

```
the best K is 16
KNeighborsClassifier(metric='euclidean', n_neighbors=16)
The accuracy of the knn model:  0.7099236641221374

accuracy for model using logistic regression: 0.7404580152671756
```

The best K for the KNN model is 16. The KNN model has an accuracy of 70% while the logistic regression model has an accuracy of 74%. The logistic regression model is best for competing in the kaggle model.

The benefits of the KNN model are:

- A quick and straightforward model of machine learning.
- A few tuneable hyperparameters.

The drawbacks of the KNN model are:

- K should be chosen wisely.
- High runtime computing costs if the sample size is large.
- For equal treatment between features, proper scaling should be given.

The benefits of the Logistic regression model are:

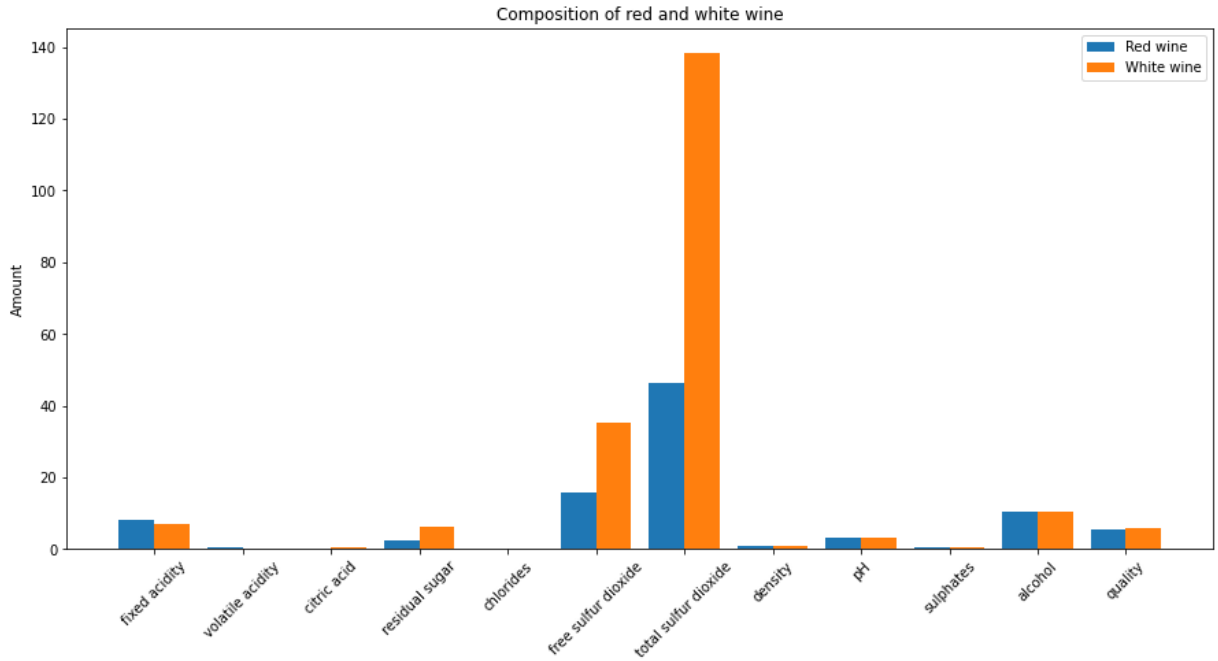
- A convenient, quick and straightforward method of classification.
- Parameters explain the direction and intensity of significance of the independent variables over the dependent variable.
- Can be used for multiclass classifications also.

The drawbacks of the logistic regression model are:

- It can not be extended to problems of non-linear classification.
- Proper feature selection is required.
- A good ratio of signal to noise is required.

Question 4

4.1 Calculate the average of each feature for the red and white wines separately and make a comparison using a bar graph showing the two wines together. How do the results relate to common sense (or the intuition of a wine expert) based on the features that are available?



I first calculated the mean of each feature using the mean function of pandas dataframe and plotted the bar graph using matplotlib.pyplot.bar function. On the graph we can see that the quality of the white wine is a bit greater than the quality of the red wine. When we look at the features we see that most of them are equal: chlorides, density, pH, sulphates, alcohol. There are instances where the red wine has a greater amount of a specific element such as with fixed acidity and volatile acidity. On the other hand, there are instances where the white wine has a greater amount of a specific element such as with citric acid, residual sugar, free sulfur dioxide and total sulfur dioxide.

The intuition of the wine expert might probably be affected by the fixed acidity, residual sugar, free sulfur dioxide and total sulfur dioxide because these are the variables that show a significant difference between red and white wine.

4.2 What is the correlation between each feature and the dependent variable using a separate analysis for white and red wine? Which variable is most relevant for each wine?

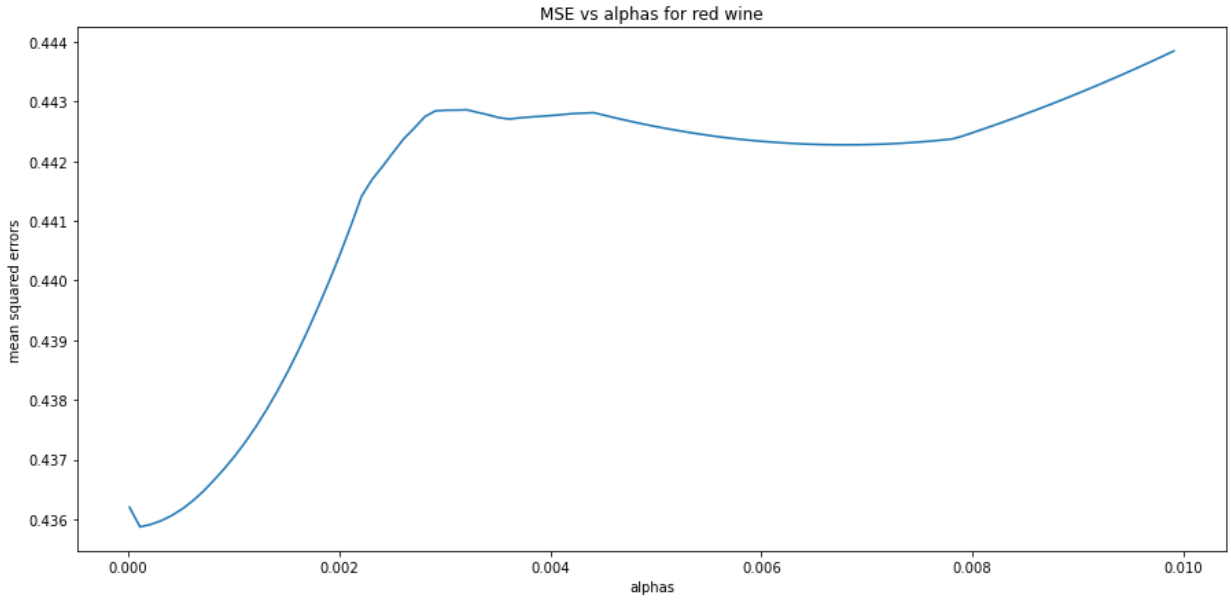
```
Correlation for Red wine
fixed acidity          0.124052
volatile acidity      -0.390558
citric acid            0.226373
residual sugar         0.013732
chlorides             -0.128907
free sulfur dioxide   -0.050656
total sulfur dioxide  -0.185100
density               -0.174919
pH                    -0.057731
sulphates              0.251397
alcohol                0.476166
quality                1.000000
Name: quality, dtype: float64
```

```
Correlation for White wine
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
Name: quality, dtype: float64
```

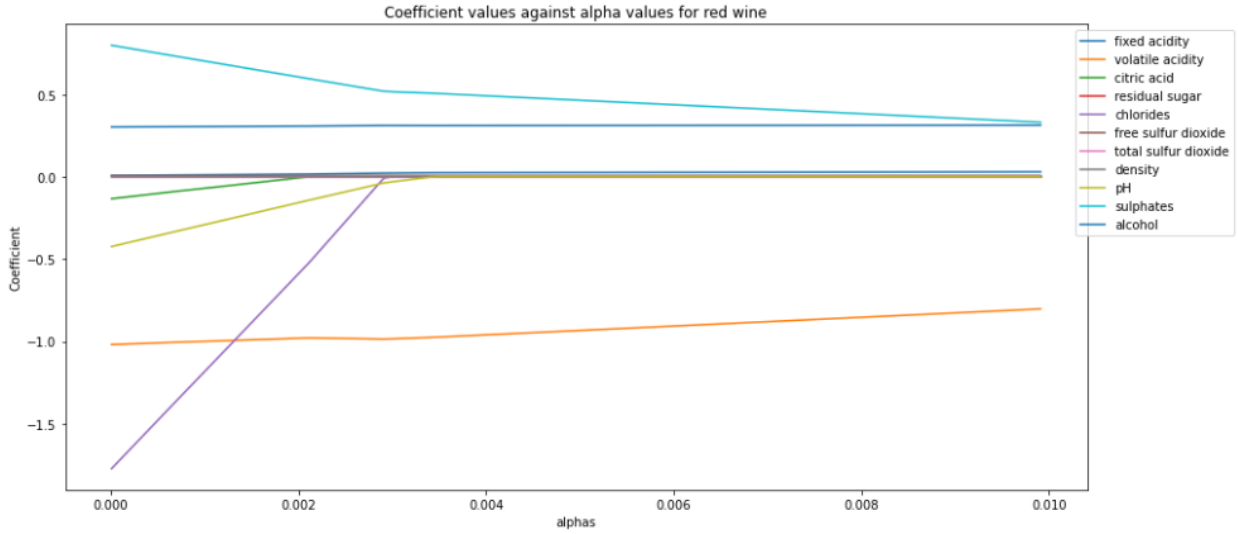
The above tables show the correlation between each variable and the quality. To calculate the correlation, I used the corr function of pandas and selected the correlations that correspond to the quality column.
For red wine we can see that the most relevant features are: alcohol (0.47), volatile acidity (-0.39), sulphates (0.25) and citric acid (0.22). For the white wine, we can see that the most relevant variables are: alcohol (0.43), density (-0.30), chlorides (-0.20) and total sulfur dioxide (-0.17).

4.3 Use Lasso and cross-validation to provide a plot of MSE against lambda and the parameter estimates versus lambda. How do the features selected by LASSO

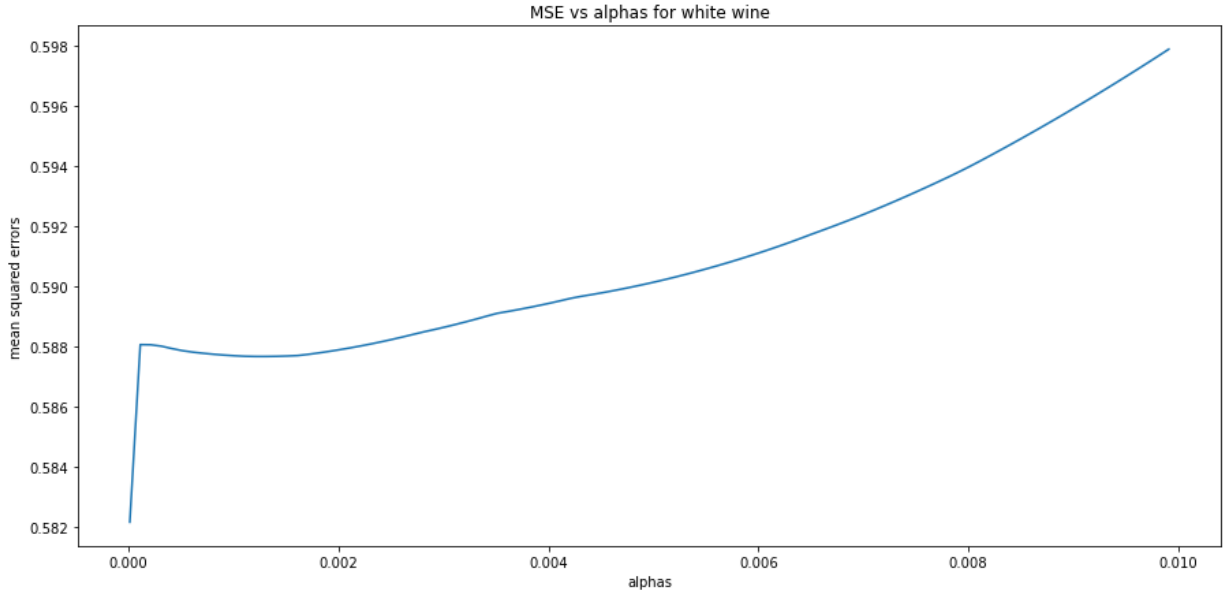
compare with an approach of setting a threshold on the absolute correlation coefficient?



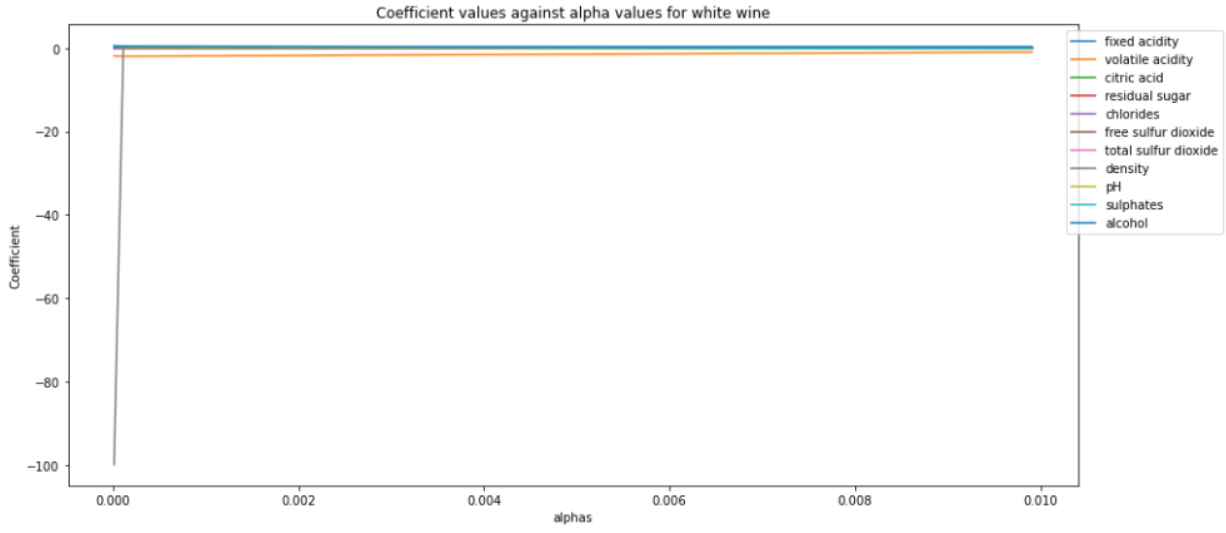
The above graph is a graph that shows the mean squared error of the lasso model (red wine) as the value of alpha increases. We can see that initially when alpha increased the MSE decreased slightly. However, after that as alpha increases the MSE of the lasso model increased and reached its peak at approximately 0.003. The MSE started to decrease slightly until approximately an alpha of 0.008. After 0.008, the MSE started to increase again.



The above graph shows the coefficient value for the lasso model as the value of alpha increases. We see that as the value of alpha increases, the coefficient values approach zero. We can see that after a value of 0.004, the coefficients that still have non-zero values are alcohol, sulfates and volatile acidity. If we set an absolute correlation coefficient of |0.25| or more, we would have selected these three variables.



The above graph shows us the MSE of the lasso model against alpha values for the white wine. We can see that the MSE starts at around 0.582 and increases as the value of alpha increases.



The above graph shows the coefficient values against alpha for the white wine. We can see that starting from a very small value of alpha, most of the coefficient values are zero. Volatile acidity is tangent to zero and continues to get closer to zero as alpha increases. Chlorides was around -100 but as soon as alpha starts increasing, it becomes zero. In the white wine case, selecting the variable using an absolute threshold is more difficult. The only way we would have gotten the volatile acidity variable is if we set the absolute correlation coefficient threshold value to be |0.19|.

4.4 Use the features identified by LASSO to construct a KNN regression model for red wine.

```
[ 0.02561365 -0.9172699  -0.         -0.00118279 -0.          0.00542362
 -0.00349791 -0.         -0.          0.44667493  0.310629  ]
7
['fixed acidity', 'volatile acidity', 'residual sugar', 'free sulfur dioxide', 'total sulfur dioxide', 'sulphates', 'alcohol']
```

I used the LassoCV function to build the Lasso model. The Lasso model selected 7 features: fixed acidity, volatile acidity, residual sugar, free sulfur dioxide, total sulfur dioxide, sulphates and alcohol. I then used only these features while constructing the KNN model using the KNeighborsRegressor() function.

4.5 What is the performance of a linear regression model and the KNN model, measured by MSE and R2? Describe the advantages and disadvantages of both models.

```
mse of knn model (red wine): 0.537735457063712
r-squared of knn model (red wine): 0.13071307774494656
```

```
mse of linear regression model (red wine): 0.39945596500356007
r-squared of linear regression model (red wine): 0.3542515342200595
```

The KNN model has an MSE value of 0.537 while the linear regression model has an MSE of 0.399. The linear model has fewer errors. The r-squared value of the KNN model is 0.13 while the r-squared value of the linear regression model is 0.35. This means that the linear regression model explains more variance in the quality variable than the KNN model.

Linear regression has the following advantages:

- Easy to fit. We only need to estimate a small number of coefficients.
- Often easy to interpret.

The disadvantages of linear regression are:

- They make strong assumptions about the form of f (X). If we assume a linear relationship between X and Y but the true relationship is far from linear, then the

resulting model will provide a poor fit to the data, and any conclusions drawn from it will be suspect

KNN has the following advantage:

- It does not assume an explicit form for f (X), providing a more flexible approach.

KNN has the following disadvantages:

- They can be often more complex to understand and interpret
- If there is a small number of observations per predictor, then linear regression tends to work better.

References

Lasso Regression Fundamentals and Modeling in Python | by Kerem Kargın | Analytics Vidhya | Medium
Lasso Regression in Python (Step-by-Step) - Statology
lassoPlot in python | Notes on engineering math (sprjg.github.io)
How to Develop LASSO Regression Models in Python - MachineLearningMastery.com

Effect Of Alpha On Lasso Regression (chrisalbon.com)
K-Nearest Neighbors Algorithm | KNN Regression Python (analyticsvidhya.com)

Building and Visualizing Decision Tree in Python | by Nikhil Adithyan | CodeX | Medium
Build Your Own Decision Tree Using Python | by Laxman Singh | Analytics Vidhya | Medium
Decision Tree Implementation in Python From Scratch (analyticsvidhya.com)
https://www.linkedin.com/pulse/decision-tree-node-calculation-laxman-singh

Decision Tree Pruning Techniques In Python (cloudyml.com)
Decision Tree: build, prune and visualize it using Python | by Gustavo Hideo | Towards Data Science
Cost Complexity Pruning in Decision Trees | Decision Tree (analyticsvidhya.com)
Post pruning decision trees with cost complexity pruning — scikit-learn 1.1.3 documentation

Steps in Developing a Classifier (stsci.edu)
Comparison of Linear Regression with K-Nearest Neighbors (duke.edu)

Logistic Regression vs K-Nearest Neighbours vs Support Vector Machine (globaltechcouncil.org)
https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa
Logistic Regression vs. Decision Tree - DZone Big Data