Introduction

Data Science is a vast subject that uses a wealth of different techniques to perform analysis on data to generate insight. Data science combines the core components of computer science, mathematics, and business knowledge to turn different forms of data into insightful material. This report will focus on data mining components, in which building effective models using machine learning techniques to determine the likelihood of an event occurring. Data mining is all about learning patterns, making predictions, and building models, which essentially means taking an input, applying a machine learning technique on it, to produce an output. This report will focus on applying the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework, this framework is an industry standard one that covers the process of building a model that can be deployed in a real-world environment. The CRISP-DM model is very flexible and not linear, it encourages the process to return to different stages to ensure the task fits the requirements needed. Below is a visual depiction of each stage of the CRISP-DM framework.

Diagram

Description automatically generated

(IBM, 2021)

Business Understanding

The background of this task involves Sydney City Council (SCC), they are responsible for the operation of one of the cities beaches, their concern has arisen as there have been a string of shark attacks on Australian beaches, they are keen to minimise the risk of such attack happening on their beach and putting the public at risk. They are also concerned about the implications of such attacks on the local economy as they will be forced to close the beach, this will result in a loss of tourism and income to the local community. The task therefore is to use machine learning techniques to produce a system that would be able to tell SCC if a shark appearance is likely to occur, with such information SCC can determine if they should keep the beaches open. The main concern is that of public safety, the consequences of having a false negative are extremely significant, we don’t want to give the beach the all clear when indeed there is a shark present. We are therefore trying to achieve an effective model that puts emphasis of safety and addressing false negatives more so than the other combinations. A good solution would be able to determine to a high level of accuracy whether a shark appearance is likely, the public and the local economy are relying on this model to be effective, we must however be wary not to close the beach at every opportunity as this would destroy the local economy, therefore we must find the correct balance in the trade-off between bias and variance to produce the best possible model.

Machine learning is suitable for this task as there are many different factors at play, the relationship between the variables are very unclear to the human eye, this meaning that machine learning techniques can be implemented to establish any relationships between the variables. There is no easy formula to determine the likelihood of a shark attack, unless it is something glaringly obvious such as dumping excessive amounts of shark food by the beach, therefore we must use the rich dataset we have been presented with from the SCC to establish a plausible outcome. This task is clearly a regression one, we use regression analysis in this instance as it is useful for identifying which variables have an impact on the question at hand. We need to be able to establish the relationships between the different variables, with a regression model we can predict an outcome from a set of variables, this outcome can be numeric or categorical.

The model should be highly transparent for the sake of the public and the SCC, the public must be able to trust such a model as they are quite literally depending their lives on it, furthermore from a business perspective it must address the SCC’s concerns and do so with a high level of accuracy, this is because the social and economic consequences lie at their feet. Having said this, we must put emphasis on the implications of errors, this is defined as the difference between a predicted output from a model and the true observed value, therefore we must take into account unexplained variation, data collection problems, sampling sizes and overall model ability.

Some important terminology definitions are defined below so that this report is accessible to a non-technical audience.

Variable – Something that can hold different values and can be measured (In the context of Data Science it is essentially a column in a normal tabular dataset).

Model – A piece of software that takes input values, applies a machine learning technique, and then produces an output value.

Parameters – Factors or limits that determine the way something is done.

Hyper Parameters – Settings that control how a model learns.

Data Understanding

Data understanding is all about the first impressions of the data and the variables it possesses. The data provided by the SCC is well formatted and in a CSV file, each row in the data relates to a specific time and details whether a shark was present at that specified time. Breaking the data down, I will comment on my first impressions of each variable and if I can infer anything from them.

Continuous Numeric Variables:

‘murkiness.level’ - This has high importance in the model as it is a key determining factor in terms of visibility of a shark.

‘diversity.of.prey’ - This can give us indication of the types of food the shark likes and if there is a simple linear regression opportunity between increased diversity of prey and increased shark appearances.

‘seels.seen’/’avg.dolphins’ - This could be significant as the shark could have these animals in its diet therefore giving us indication if the more there are the more likely a shark will be on the scene.

‘previous.week.fishing’ - This again could suggest the more fish the more likely a shark is to be present.

‘water.temperature’ - This is useful as we can establish what kind of climate the shark prefers; this kind of data would be useful in a classification problem if we were trying to determine what type of shark it is.

‘seagull.density’ - Although sharks don’t necessarily eat seagulls, I believe this variable is still important as it could indicate the density of fish, as seagulls follow the fish, as do the sharks.

Discrete Numerical Variables:

‘people.previous.day’ - This shows how many people were at the beach on each given day, this could suggest things such as the more people the more likely the shark is going to attack as they have a wealth of targets.

Nominal Categorical Variables:

‘comon.surfboard.colour’ - This again is a highly contested variable as research suggests that sharks are colour blind, however others suggest that in murky waters such vibrant colours would stand out to a shark.

Ordinal Categorical Variables:

‘time.of.day’ – This is an essential variable as we can determine when a shark is most likely to appear based of the time.

Both the overcast weather and shark appearances are binary numeric variables taking the value of zero or one, one being that the variable is evident.

For a task where the consequences are so severe, I believe that all variables provided by the SCC are important contributors to the final prediction and therefore must be included in the model. The initial view of the data is promising as it has some key variables which could be very telling, my only concern is the reliability of the collection, who is recording shark appearances? Can a human always tell the difference between a dolphin and a shark? These are some questions that must be answered before creating a model as they are massively influential in terms of the model’s accuracy.

It is important at this stage to gather some more information on the data, such as its distribution, we can infer from this if there are any general errors. Before starting any data mining task, it is important to also understand some summary statistics, from running the data into the Python terminal we can see that there are thirteen variables and four hundred and fifty-six rows of data, this is a nicely sized data set for the task at hand, if we need additional data it is important to converse with the SCC about the availability of additional collection. We can also run a short script to confirm that we are working with the correct types. Another great step is to establish a range of histograms, to briefly look at the distributions. Below are the histograms for each numeric variable, a histogram visualises how often a variable appears and is categorised by being distributed into bins.

Chart, box and whisker chart

Description automatically generated

Straight away we can see several issues in the dataset, the first being clear outliers present in the murkiness level and diversity of prey, this is evident as there are huge bars followed by small bars with a resulting bar far out of reach from the rest of the data set, this must be addressed before creating a model. The distribution of a continuous variable can be described by a probability density function, this is shown in a smooth curve that can be approximated with a histogram. I used a probability mass function for the ordinal categorical variables so we can see the distribution of times the data was recorded at; this is shown below, we can also see in this example more evidence of data input errors. In this stage I also checked for missing values defined as null values, this told me where missing data was and has given me the opportunity to remove or fill these rows.

Chart, bar chart

Description automatically generated

Data Preparation

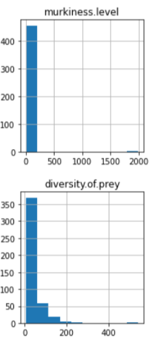
This stage is largely about taking what we have inferred from the data so far and correcting it to a degree that is sufficient to run through our model. Firstly, we will correct the spelling mistakes in the data collection, here we will set the incorrect spelling to equal the correct spelling, this shown in the updated histogram below.

Chart, bar chart, histogram

Description automatically generated

Chart, histogram

Description automatically generatedMurkiness level and diversity of prey both had outliers in the dataset provided, the number of outliers are extreamly small so I believe it is best to drop these rows from the dataset as they don’t contribute any value. Below shows those histograms updated once the outliers had been dropped.



I also decided to implement some normalisation just for good practice. Normalisation in machine learning allows for greater efficiency when the variables are on a smaller scale, I applied this normalisation to a continuous numeric variable of diversity of prey. To normalise I first had to convert said data into a numpy array and then apply a min max scaler to reduce the size of the numeric values. I also took this opportunity to balance out some of the data, this was needed where the data was heavily skewed to to the left.

In the data preperation phase it is important to split our data into the necessery sets, it is often recommended to split our data sixty percent training set, twenty percent validation set and twenty percent test, however there are no hard rules for this split. It is important to split the data in this way as if we constantly evaluating our data on the training set our model will start to tune to that set, therefore we must further introduce the validation set so that the test set is fresh. The training data is where the majority of the operations happen and is used for setting the model parameters. We may however run into issues where all the hard to predict data points are in the final test set, to get around this problem we can make several training and validation sets, the sytematic way to achieve this is to use k fold cross validation, this process splits the data into five folds giving us a greater insght into the variance and helps reduce the risk of relying on a single valisdation set. Sklearn was used to split the data in the way outlined above, I made sure to use the .shape function to ensure that the correct split had been made.

Modelling

Here in the modelling stage is where we put our data to use. This report will focus on three separate models, a tree based model, a logistic regression based model and a neural network based model, the highest performing model will be the one implemented in the evaluation stage. As previously stated our data is split and ready to be worked with. We must chose a cost function, the most appropriate would be the mean squared error as we are trying to find out how far out our model is.

To begin a tree based model will be constructed, this is better known as decision tree, a decision tree progressively divides the dataset into smaller subsets with the intention of having each subset of the data set containing only one label. One of the only hyperparameters that can be used with this model is deciding the depth of the tree. It is important to select the right level of depth to avoid underfitting or overfitting the data. A little more about decision trees are that they are made up of two types of node, one is a branching node and represents a single variable and has a branch for each discrete value, the other is a leaf node that represents a class label. The algoritm in a decision tree makes the split for us and we use a selection of measures to determine the quality of that split. For this model the use of one hot encoding must implement as sklearn cannot take categorical variables as imput for a decision tree model.

Having constructed and completed the first run of the model the cross validation score returned an average mean of 0.827, it is important to note the higher the output number is to one the greater the level of accuracy. Looking at the visualisation of the tree we can see how the internal node contains a variable name, and a rule to splist the associated variable on, this is shown below.

Diagram, engineering drawing

Description automatically generated

Having received an initial average for this model I will train the model on three separate values to determine which output gives us the highest level of accuracy, the distribution of the depths will be low, middle and high values.

The values I used to test the range of depth were five, fifty and one hundred. I found that both the five and fifty imputs had around a 0.85 level of accuracy however the larger depth measure fell to 0.801. I think the split of the decision tree largely depends on the amount of variables and their associated values. We can also use a grid search to give us the best set of parameters, so far I have been manually adjusting the parameters but using a certain script will give us the optimum split results. We can test the gini coefficient and the entropy, these are two both give us a greater perspective on quantifying a random value, having run a grid search on the decision tree the optimum cross validation accuracy returned at 0.862, this has not varied to far from the initial run of the model and suggests that the cross validation is doing a good job in keeping the mean averages within a small range.

Before we move to the next model it is important to note that model that fits the training data too well but generalises poorly has low bias and high variance, this is to be avoided. The next model to be produced is the logistic regression model, this model is used when there are a variety of input vatiables ranging from numeric to categorical ones, in logistical regression the output number is between zero and one, this is the probability of the input pattern belonging to the positive class, this form of regression ensures that the output is between zero and one by taking the inputs and mapping them into the stated range. Furthermore logistic regression provides a way of learning the coefficants of a model to be interpretted as a probabilty.

Having run the logistical regression model on my dataset the test accuracy returned as 0.803, this is marginally lower than the previous model. We can evaluate the fit of this regression using some performance metrics the main ones that can be used are mean squared error, mean absolute error and R-squared . I will use Mean squared error in this report, this will give me the mean of the square of all the errors. The mean squared error returned a value of 0.0525, this value is deemed relatively acceptable, the closer the mean squared error to zero the better the model is, as all the predicted values match the expected values.

The final model to be used is a neural network, neural networks are an expansive model that have have the ability to complete extraordinary tasks, such as spotting if a driver is falling asleep at the wheel. The process works through the neurons taking an imput value and producing an output value that is fixed, the weights are what change during the learning process and their values determine the behaviour of the network as a whole. The neurons are organised into layers and connected to those in the next layer. The model works by passing the incoming data through an acivation function, this funcion can vary and can be in logistic, linear or ReLU form. The model is refereed to as a multi layer perceptron (MLP), the process of building an MLP is much more complex than previous models.

The classification acuracy returned from this model was 0.794, this being the lowest of the three models. This number is difficult to account for however we know that not every model fits the dataset and task at hand. There are a range of hyperparameters that can be used to alter the model, the ones I applied were the hidden layer size and the max iterations. Having tested the different hyperparameters for this stage the results were quite similar, I expected a greater level of accuracy however it largely floated around the same numbers. The hidden layer size returning 0.781 and the max iterations returning 0.804. This processes was repeated using higher and lower values but the variation in results was almost null.

Having built three models it is now essential that I compare them and use the validation set to arrive at a final outcome. I am also going to go over all the models using grid search to really find the best parameters, this comparison will therefore ensure that all models are to be at the peak of their power when being compared.

Having completed a systematic evaluation of the model options there is a suitable model over the others. To reach this conclusion I used my initial findings and the average performance of the five fold cross validation to give me an initial indication, this process cannot be underestimated as it takes the average of five separate splits in the data. I also altered the hyperparameters for each model testing low, midlle and high levels to spot any differentiation, as we know this is a manual task and not one which gaurentees the best optimisation of the parameters. The resulting factors gave the same results as my initial test with the variation of averages changing to a miniminal extent. I finally tested each model on their validation set, this set is fresh and gives the best opportunity to test the models against unseen data. The decision tree again performed best in this and actually improved its mean average from 0.827 to 0.842. To conclude the decision tree has shown to be the best model in terms of mean accuracy and most likely to return the best measure of whether a shark is likey to appear or not.

Evaluation

The decision tree model performed marginally better than the other models in this report and is a useful model in terms of predicting the likelihood of an event happening.

Chart, treemap chart

Description automatically generated

This confusion matrix is one of the key visualisations of this report as the element of false negatives is significantly minimised compared to the others, it is important that we get this right as we don’t want to give the all clear at the beach when there is a shark present. This model is far from being perfect however, the closer the accuracy level is to one the greater it is, for a task with such consequences I believe that the mean average rate is on the edge in terms of being accurate enough to be trusted by the SCC and the wider public. This confusion matrix however has a high number of false positives, this meaning that it may be overly cautious in certain areas, this may concern the SCC as the beach may be getting closed more often than it should be.

Other calculations taken in this report in regards to the chosen model are relatively positive, these include the hyperparameter adjustments. The SCC may wish to use this model, however they may get a greater model if they prioritised one variable over another, that being public interst over economic uncertainty, this would be represented in a more concise confusion matrix diagram.

To conclude the decision tree model performed the best on the whole, this is further supported with the final test on the test data receiving the best result yet of 0.874, this is an ideal model to implement but could be improved depending on factors such as the business decisions and data collection methods. The SCC can be confident that they have a better idea on the situation now they have invested in machine learning.

References

IBM (2021) CRISP-DM Help Overview – Available at: <https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview>