

**ANL488 July 2023 Final Report**

|  |  |
| --- | --- |
| **Course Code** | ANL488 |
| **Project Title** | Prediction of potential CO2 storage locations |
| **SUSS PI No** | N1981115 |
| **Name** | Low Kai Boon Calvin |
| **Submission Date** | 6 November 2022 |

**Table of Contents**

[1. Abstract 3](#_Toc149744544)

[2. Introduction 4](#_Toc149744545)

[3. Literature Review 4](#_Toc149744546)

[4. Data Mining Using CRISP-DM Methodology 6](#_Toc149744547)

[4.1. Phase 1: Business Understanding 7](#_Toc149744548)

[4.2. Phase 2: Data Understanding 8](#_Toc149744549)

[4.2.1. The CSS Dataset 8](#_Toc149744550)

[4.2.2. The CSRC Dataset 9](#_Toc149744551)

[4.3. Phase 3: Data Preparation 10](#_Toc149744552)

[4.3.1. Cleaning the CSS Database 10](#_Toc149744553)

[4.3.2. Cleaning the CSRC Dataset 13](#_Toc149744554)

[4.3.3. Combiming Datasets & Final Cleaning 14](#_Toc149744555)

[4.4. Phase 4: Modelling 17](#_Toc149744556)

[4.4.1. Pre-Modelling Preparation 21](#_Toc149744557)

[4.4.2. Building Predictive models 22](#_Toc149744558)

[4.5. Phase 5: Evaluation of Predictive Models 24](#_Toc149744559)

[4.5.1. Evaluation of Results 27](#_Toc149744560)

[4.6. Phase 6: Deployment of Predictive Model 32](#_Toc149744561)

[4.6.1. Recommendations 32](#_Toc149744562)

[4.6.2. Discussion 33](#_Toc149744563)

[4.6.3. Next Steps 33](#_Toc149744564)

[5. Conclusion 34](#_Toc149744565)

[6. References 35](#_Toc149744566)

# Abstract

Since the pre-industrial era, the earth has been experiencing the increase in temperatures, causing detrimental effects on the environment and population centres around the world. To combat global warming, there are several initiatives proposed to address the effects of global warming, one of which is Carbon Capture and Storage (CSS).

In this case study, I will be determining the feasibility of Carbon Capture and Storage (CSS) projects. To do this, I will be using machine learning techniques to analyse two public datasets, one which contains information regarding dedicated CSS facilities constructed around the world while the other contains information regarding various oil fields and saline aquifers found around the world that can be used to store Carbon Dioxide (CO2).

I first combine these two datasets together and retaining only necessary information detailing the facility type, its storage or capture capacity, the technologies used, the combustion and separation methods and the project cost, with the overall status of the project set as the target variable. Out of the predictive models available, I have decided to use Classification and Regression Trees (CART) and Artificial Neural Network (ANN) together with various ensemble models as my preferred predictive models to work on the dataset. Based on the evaluation of all the models’ performance, the ANN model with Bagging is chosen as the best model as it has the highest weighted F1 score among all the other models.

Since the ANN model is a black box, I decide to use the SHapley Additive exPlanations (SHAP) python library to analyse the features of the predictive model. Together with using the Classification and Regression Trees (CART) model for interpretability, I determine that the best option to store CO2 is to inject up to 8545 metric tons of pressurized CO2 into an oil field. However, dedicated CCS facilities will still need to be constructed to capture the CO2 and shipped to the oil fields for storage.

# Introduction

Every year, global temperatures are increasing at a rapid pace due to global warming, which is caused by the greenhouse and El Nino effects. One method that will be able to lessen the effects caused by global warming is using CSS, where the CO2 is captured and stored during the pre-combustion or post-combustion of industrial processes, preventing the CO2 from entering into the atmosphere. As CCS is a promising technology that aims to prevent further emissions and increase of greenhouse gases into the atmosphere, there is potential to commercialise the technology and to build more CCS facilities and storage sites to mitigate the effects of global warming.

Therefore, by using predictive modelling techniques, I will predict the feasibility of potential CCS projects based on a combination of various factors by utilising information obtained from public databases that contains data regarding existing CCS facilities and storage sites from around the world to construct different types of predictive models. Based on the insights gained from these predictive models, I will be able to determine the different number of situations and circumstances in which the CSS project can be feasible. This information can then be disseminated to companies undertaking CSS projects, thus helping them to make informed decisions and increasing the chances in which the project will be a success.

# Literature Review

According to the World Meteorological Organisation (2023), global temperatures from the year 2023 to 2027 is predicted to have an estimated increase of 1.1℃ to 1.8℃ higher than the average temperature increase recorded between the years 1850 to 1900. This drastic increase in temperatures is caused by the greenhouse and El Nino effects. The greenhouse effect is caused by the trapping of heat by greenhouse gases, including carbon dioxide, whereas the El Nino Effect is caused by warmer waters releasing heat into the atmosphere. Thus, these two effects contributing to the increase of temperatures around the globe.

With global temperatures increasing around the world at an increasing rate, there will be adverse effects felt in population centres around the world. According to Haines & Hatz (2004), the adverse effects of global warming include thermal stress, increased occurrence of floods and droughts, malnutrition, and an increased number of diseases contracted by humans. As the climate becomes warmer, heat waves will become more prominent, resulting in wildfires, causing air pollution and an increased number of heat-related injuries or deaths. A warmer climate will also mean that weather patterns will become more erratic, resulting in an increased number of floods and droughts, destroying crops, and causing starvation and famines in several countries. Lastly, a warmer climate can also cause pests, including mice, fleas, and mosquitoes to thrive in tropical areas, leading to an increased number of people contracting animal-borne illnesses.

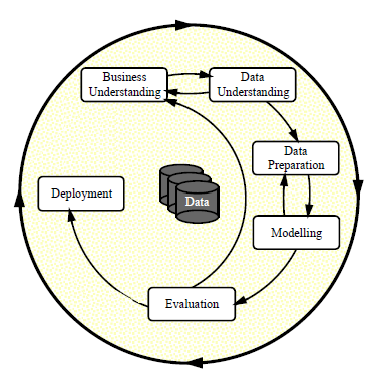
One way to mitigate or lessen the effects of global warming is to reduce the amount of greenhouse gases in the atmosphere. Such technologies are known as CSS, where the CO2 is captured during industrial processes which are notorious for greenhouse gas emissions, including the burning of fossil fuels. Using various CCS technologies, the captured CO2 can then be stored in a storage facility or be pressurised and injected into a stable geological feature (Boot-Handford et al, 2014). Both of these methods are ensured that the CO2 emissions will be reduced by storing them in a secure facility and preventing them from entering into the atmosphere.

According to Handford et al (2014), there are several methods in which CCS can be implemented. The most common methods include solvent scrubbing and oxyfuel combustion (Handford et al, 2014). Solvent scrubbing is the process of letting the CO2 react in a chemical reaction with a solvent that creates a by-product that contains carbon and oxygen particles, which then can be stored. On the other hand, oxyfuel combustion involves the use oxygen-rich gas that generate emissions that contains only carbon dioxide and water, making it easier to capture the carbon dioxide from the flue gas before it reaches into the atmosphere. In recent years, several companies have developed and introduced their own CCS technologies.

There are several advantages of CCS (Bui et al, 2018). One advantage that CCS provides is the easy integration with existing energy systems. This is due to the fact that CCS technologies can be easily retrofitted without causing much changes to existing systems. Another advantage that CCS provides is the decarbonisation of emission-intensive industries as such industries are notorious for producing carbon dioxide emissions. Thus, using CCS to capture these emissions will lower the industries’ carbon footprint. Lastly, CCS can be used in conjunction with carbon-neutral bioenergy (BECCS) to produce negative emissions as CCS will be able to remove as much carbon dioxide from the atmosphere as the amount of carbon dioxide produced by BECCS.

However, according to Haszeldine (2009), the main disadvantage is that CCS has is regarding its operational costs as CCS will be expensive to commercialise at the initial stages. This is due to CCS requiring 25% to 40% of a power plant’s energy needs. However, despite the expensive costs need to commercialise CCS, it is useful to note that towards the path of commercialisation, CCS requires larger equipment to be constructed for prototyping or demonstration purposes. Hence, with the lessons learned from the CCS prototypes or demonstrations, it can be used to improve the reliability of these CCS technologies and to bring the operational costs down as adoption of the technology increases. Therefore, this will make CCS a viable option for reducing carbon dioxide emissions and towards the path of decarbonisation.

# Data Mining Using CRISP-DM Methodology



*Figure 4.1. The CRISP-DM Diagram*

According to Wirth and Hipp (2000), the CRISP-DM framework consists of 6 phases, namely the business understanding, data understanding, data preparation, modelling, evaluation and deployment phases. As shown in figure 4.1 above, the CRISP-DM framework is a continual improvement cycle in which the experiences gained from previous data mining projects can be used to improve future data mining projects.

During the business understanding phase, the company will need to identify the business problem it faces and come up with objectives to solve the problem. To ensure the project is on track, the company should assess the situation by identifying the resources needed for the project, and understand the benefits, risks and pitfalls of the project. The final step the company needs to undertake during the business understanding phase is to find out the project's goal and develop a project plan accordingly.

During the data understanding phase, the company will need to collect the data they need for the project and to undertake data understanding and exploration to look for any inconsistencies and data quality issues. During this phase, the data is cleaned to remove any inconsistencies and data quality issues identified and to make it suitable for data mining.

During the modelling phase, the company should select the appropriate data mining technique and suitable methods to evaluate the data mining model’s performance before building the data mining model. During the validation phase, the company will need to evaluate the performance of the model and determine the steps to take before the deployment of the model. Lastly, the deployment phase involves the usage, maintenance, and monitoring of the model to ensure it is kept up-to-date and relevant in solving the business problem.

As shown in Figure 4.1, the transition between the different phases is not strictly unidirectional as the transitioning between the different phases can go in both directions, when there is an improvement to be made in the previous phase when working in the next phase.

## Phase 1: Business Understanding

Due to the threat of global warming, there are several mitigation measures proposed to lessen the effects of global warming. However, out of the mitigation measures proposed, one viable mitigation measure that can be undertaken is CCS as the process of capturing and storing CO2 has been done in a few facilities around the world, with several more such facilities being planned or developed. Thus, there are initiatives to propose and develop more CSS facilities to address the issues of global warming.

However, the development of the CSS facilities can be a challenge as these projects often large in scope and require complex planning and implementation. In addition, there are various factors to consider when undertaking the project, including how much the project should cost, the amount of carbon dioxide to capture and store and the types of CSS technologies to use and the types of facilities to build.

Thus, there are public databases available that records information from past, current and future CSS projects to encourage companies to make informed decisions and risks when undertaking CSS projects. By using information obtained from these public databases, the objective is to develop a predictive model that will be able to predict and explain the feasibility of potential CSS facilities based on the factors discussed.

## Phase 2: Data Understanding

For this case study, I am given two datasets to work with, namely the CCS and CSRC datasets. The CCS dataset contains information about CSS facilities constructed with the purpose of capturing or storing of CO2 or both. On the other hand, the CSRC dataset is created by the CO2 Storage and Resource Catalogue (CSRC) to record natural-occurring geological features that can be used to store CO2 for millennia.

### The CSS Dataset

An overview of what the dataset contains shown in Table 4.2.1.

|  |  |  |
| --- | --- | --- |
| **S/N** | **Header Name** | **Description** |
| 1 | Project Name | The name of the project |
| 2 | Company | The company that oversees the project |
| 3 | Plant Name | The name of the CCS plant |
| 4 | Storge and/or Capture | Whether the CCS plant in involved with Capture, Storage or Capture and Storage of CO2 |
| 5 | Overall Status | Whether the overall project is active, completed, hold, potential or terminated |
| 6 | Plant Status | Whether the CSS plant is cancelled, decommissioned, existing, in development or planned |
| 7 | Country Location | The country in which the CCS plant is located. |
| 8 | State Location | The country’s state in which the CCS plant is located. |
| 9 | Specific Site Location | The location in which the CCS plant is located. |
| 10 | Plant Sie or Capture Amount | The amount of CO2 intended for the CCS plant to capture/store. |
| 11 | Combustion/Separation | The types of combustion/separation technologies used by the CCS plant |
| 12 | Capture Technology | The types of CCS technologies used by the plant |
| 13 | Amount of C02 Captured/Stored | The amount of carbon dioxide captured/stored by the plant |
| 14 | Captured/Stored Unit | The measurements used to determine the volume of carbon dioxide captured/stored |
| 15 | Project Summary | A short description of the project |
| 16 | Project Start Date | The starting date of the project |
| 17 | Project Cost | The cost of the project |
| 18 | Currency | The currency which the company pays for the project |
| 19 | Project Information Webpage | The hyperlink that directs to the information regarding the project |

*Table 4.2.1. CCS Dataset Fields and Descriptions*

### The CSRC Dataset

The CO2 Storage and Resource Catalogue (CSRC) is a multi-year project held over six annual cycles that is commissioned by the Oil and Gas Climate Initiative (OGCI) and spearheaded by Global CCS Institute (GCCSI), starting from cycle 1 in 2020 and ending with cycle 6 in 2025. The OGCI aims to gather and obtain a global perspective of the commercial readiness in using CCS.

The CSRC database is thus set up to record information regarding natural-occurring geological sites that can be used for CO2 storage, which are then evaluated using the Storage Resource Management System (SRMS) to determine the site’s maturity level. The CSRC database is updated yearly, with newly discovered CO2 storage sites added in during each cycle. However, as there are numerous columns contained in the dataset, the overview of what the dataset contains shown in Table 4.2.2 will only show the information I require for analysis.

|  |  |  |
| --- | --- | --- |
| **S/N** | **Header Name** | **Description** |
| 1 | Type | Whether the geological feature is an oil field (petroleum) or a saline aquifer (saline). |
| 2 | Discovery Status | Whether there is enough information or regulations to justify the geological feature’s discovery. |
| 3 | Aggregated SR (Mt) - Mid | The total average amount of carbon dioxide the geological feature can store. |

*Table 4.2.2. CRSC Dataset Required Fields and Description*

## Phase 3: Data Preparation

Before I proceed with the cleaning of the dataset, I need to import both Pandas and Numpy libraries, as shown in Figure 4.3.1 below.

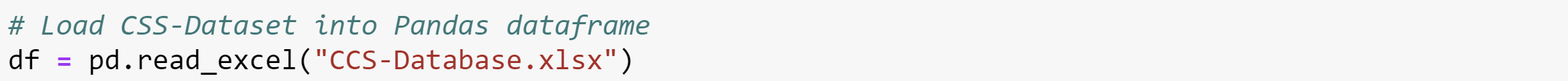


*Figure 4.3.1. Import the Pandas and Numpy libraries*

The Pandas library is used for data manipulation and analysis whereas Numpy library is used for manipulating arrays.

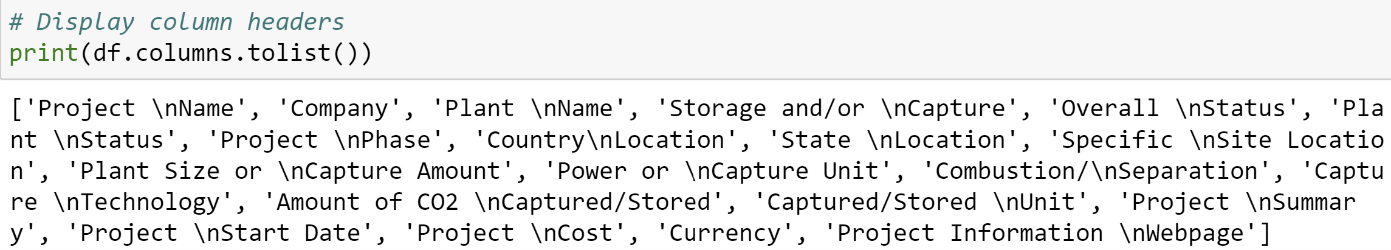
### Cleaning the CSS Database

Before I start the data preparation process, I will first read the CSS dataset into a Pandas dataframe, as shown in figure 4.3.1.1 below.



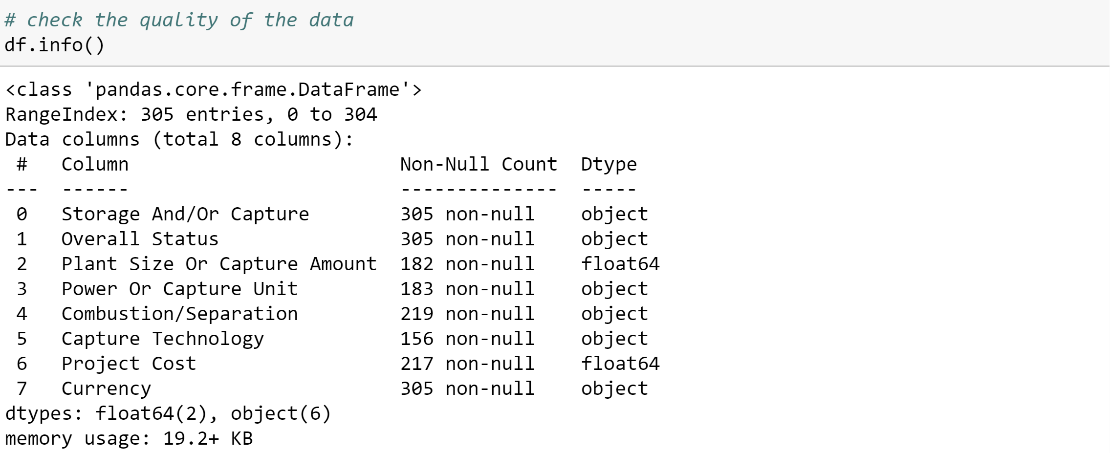
*Figure 4.3.1.1. Reading CCS Dataset into Pandas Dataframe*

There are a few data quality issues I have encountered after reading the data into the dataframe. The first issue I face is the naming of the column headers as they included line breaks, as evident in the “\n” newline characters, which is shown in figure 4.3.1.2 below.



*Figure 4.3.1.2. Pandas Dataframe Columns Headers*

The second issue the I face is the dataset containing missing values, which is evident in the difference in the non-null vales in the columns as shown in figure 4.3.1.3 below. The missing values will need to be removed as these null values will hinder my analysis when I build the predictive models.

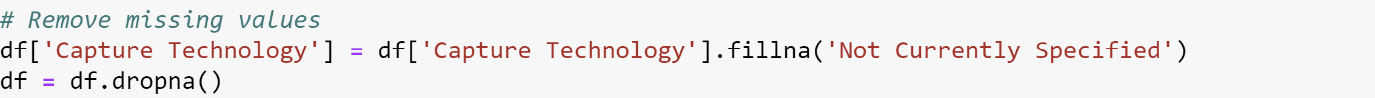


*Figure 4.3.1.3. Missing values in Pandas Database*



*Figure 4.3.1.4. Codes to clean the “CCS-Database.xlsx”*

According to Figure 4.3.1.4 above, I decide to rename each of the column headers in a for loop by capitalising each word and replacing the line break character “\n” with an empty string. As the dataset contains additional project information that is redundant in my analysis, I use the drop function to remove these columns.



*Figure 4.3.1.5. Codes to clean the “CCS-Database.xlsx”*

As shown in Figure 4.3.1.5 above, prior to removing the missing values from the dataset, I decide to fill in the missing values in the ‘Capture Technology’ column with ‘Not Currently Specified’ as this value is already present in the column itself. After that, I can proceed to remove the remaining missing values by the using dropna() function.

Furthermore, according to the measurements shown in the CCS dataframe, there are different units of measurement used in the collection of the data. Thus, I will need to convert all amounts to a common unit. These common units are metric ton for the capture amount and United States Dollar (USD) for currency, with the conversion rates as shown in Table 4.3.1 and Table 4.3.2 respectively.

|  |  |
| --- | --- |
| **Original Capture Unit** | **Converted Capture Unit In Metric Ton** |
| Barrels Per Day | 0.138 Metric Ton |
| Cubic Foot | 0.0278635 Metric Ton |
| Megawatt | 284.345136094 Metric Ton |

*Table 4.3.1. Conversion to Metric Ton*

|  |  |
| --- | --- |
| **Original Currency** | **US Currency** |
| Australian Dollar | 0.66 USD |
| British Pound | 1.28 USD |
| Canadian Dollar | 0.75 USD |
| Chinese Yuan | 0.14 USD |
| Danish Krone | 0.15 USD |
| Euro | 1.10 USD |
| Japanese Yen | 0.0070 USD |
| Norwegian Kroner | 0.0098 USD |

*Table 4.3.2. Conversion to United States Dollar (USD)*

A screenshot of a computer

Description automatically generated

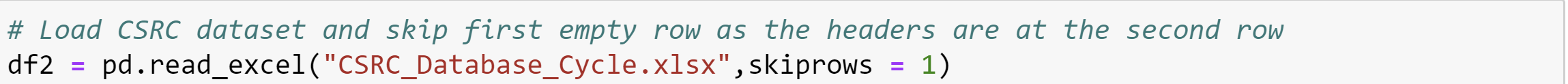
*Figure 4.3.1.6. Codes to convert measurements to a common unit*

According to Figure 4.3.1.6 above, I first store the capture unit and currency unit in their respective lists, along with their conversion units in another list. After that, I use the np.where function to aid in the conversion of the measurements by iterating through both lists at the same time. The np.where function works similarly to an if-else statement, where the condition is specified in the first parameter, the value I want to change if the condition is met is specified in the second parameter and the value I want to change if the condition is not met is specified in the third parameter. In this case, I want to keep the original value when the condition is not met.

After we are done with the conversion, I can proceed to drop the columns that contain the capture or currency units from the dataframe.

### Cleaning the CSRC Dataset

To read the “CSRC\_Database\_Cycle.xlsx” dataset into a Pandas dataframe, I will need to include the skiprows parameter and set it to one so that Pandas will skip the first row and read the data from the second row onwards, as shown in figure 4.3.2.1. This is because the column headers start at the second row in the excel workbook.



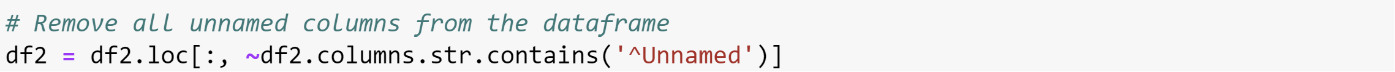
*Figure 4.3.2.1. Reading CSRC Data into Pandas Dataframe*

In addition, there are unnamed columns included in the dataset, as shown in figure 4.3.2.2 below. Thus, I will remove these unnamed columns that contains the string “Unnamed”, as shown in figure 4.3.2.3 below.

A screenshot of a computer

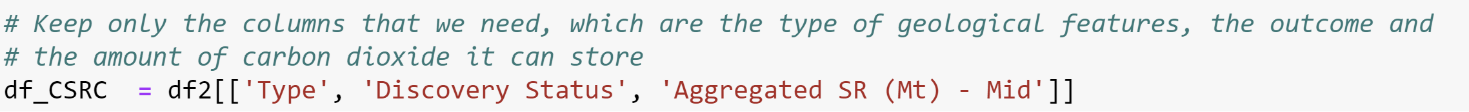
Description automatically generated

*Figure 4.3.2.2. Unnamed Columns in Dataframe*



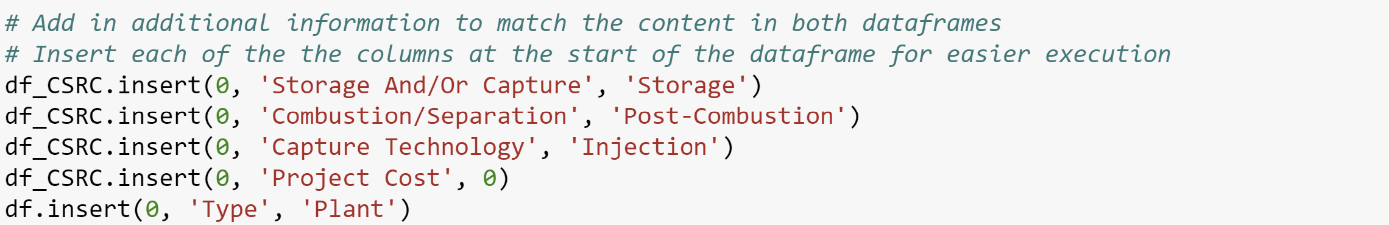
*Figure 4.3.2.3. Removing Unnamed Columns from Dataframe*

After I remove the unnamed columns, I can choose what columns to retain by including only the columns I needed in a new dataframe, as shown in figure 4.3.2.4.



*Figure 4.3.2.4. Subletting CSRC Dataframe*

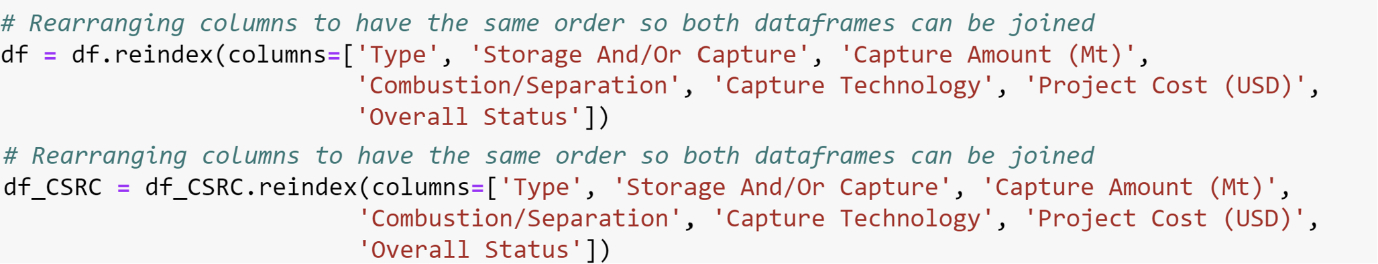
### Combiming Datasets & Final Cleaning



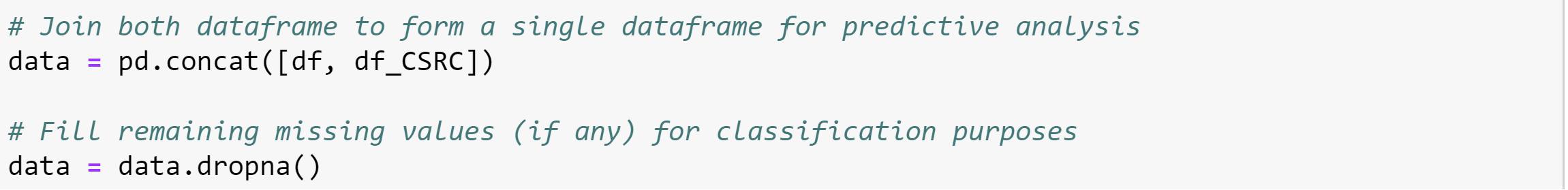
*Figure 4.3.3.1. Subletting CSRC Dataframe*

According to Figure 4.3.3.1, before I combine both the CSS and CSRC dataframes together, I will need to add additional information into both dataframes to match what is in both the CSS and CSRC dataframes. First, I add a 'Combustion/Separation' column to the CSRC dataframe and set the value to 'Post-Combustion' as the carbon dioxide is captured after the industrial process releases CO2 emmissions. Secondly, I add a 'Capture Technology' column to the CSRC dataframe and set the value to 'Injection' as the captured carbon dioxide will be injected into the geological feature to be stored indefinitely. I will also add a 'Project Cost' column to the CSRC dataframe and set the value to 0 as the geological features are naturally occurring, there is no need to construct new CCS storage sites. Lastly, I add a 'Type' column in the CSS dataframe and set the value to 'Plant' as the CSS facilities are constructed from the ground up.

After ensuring that I have the same number of columns in both dataframes, I will rename the dataframe columns in both dataframes to match each other, as shown in figure 4.3.3.2 below. After I have the same column headers in both dataframes, I can join the CSRC dataframe and the CSS dataframe together to form a new “data” dataframe using the pd.concat function, as shown in figure 4.3.3.3 below.



*Figure 4.3.3.2. Rearranging the CSS and CSRC Dataframe’s Columns*



*Figure 4.3.3.3. Concatenating the CSS and CSRC Dataframes Into a Single Dataframe*

Lastly, as predictive modelling algorithms only allow numerical values, I will need to convert the non-numerical values to numerical categorical values by using one-hot encoding, as shown in Figure 4.3.3.4 below.



*Figure 4.3.3.4. Using One-Hot Encoding*

According to Figure 4.3.3.4, I will apply one-hot encoding on all columns except for the ‘Overall Status’ column by using pandas’s get\_dummies function. The function works by converting categorical variables into indicator variables with a value of either 0 or 1.

Since the target variable ‘Overall Status’ has many categorical values but needs to have numerical values of either 0 or 1. Thus, I have to decide which categorical values should be assigned to 0 and which categorical values should be assigned to 1, as shown in table 4.3.3.

|  |  |
| --- | --- |
| **Categorical Value** | **Numerical Value** |
| Active | 1 |
| Completed | 1 |
| Discovered | 1 |
| Discovered awaiting detailed evaluation | 1 |
| O&G Assumed discovered | 1 |
| Partly Discovered | 1 |
| Undiscovered | 0 |
| Potential | 0 |
| Hold | 0 |
| Terminated | 0 |

*Table 4.3.3. Categorical Values with Their Corresponding Numerical Values*

Therefore, I will manually encode the ‘Overall Status’ column by using the np.where function, as shown in figure 4.3.3.5. An example of the final dataset is shown in figure 4.3.3.6.

A screenshot of a computer code

Description automatically generated

*Figure 4.3.3.5. Manually Encoding of Categorical Variables*

A screenshot of a computer

Description automatically generated

*Figure 4.3.3.6. Dataset after final cleaning*

After finishing cleaning the dataset, I can proceed with the modelling phase of the CRISP-DM framework.

## Phase 4: Modelling

There are a few prominent prediction models available including logistic regression, decision trees, Artificial Neural Networks (ANN) and Support Vector Machines (SVM). However, among the numerous different prediction models, I decide to use the CART decision tree and ANN for my modelling and evaluation of the data.

Decision Trees are tree-like predictive modelling structures consisting of various types of nodes and branches. The types of nodes in the decision tree consist of a single root node, which is at the top of the tree, the decision nodes, which each represents a decision made resulting in the separation of all observations in that node into two or more mutually exclusive sub-observations and the leaf nodes, which each represents the all the possible outcomes based on a combination of decisions made at the decision nodes (Song & Ying, 2015).

Furthermore, a decision tree connects different types of nodes together through a hierarchy of branches with each path starting from the root node going through decision nodes and ending at a leaf node, which represents a decision rule, consisting of all the decisions made from the root node to each single leaf node (Song & Ying, 2015).

To split the nodes, CART trees have multiple criteria for splitting the nodes binarily. The first criterion used by the CART tree is finding the split that has the greatest reduction in the Gini Impurity Index. To do this, the criterion uses Gini Impurity Index and the Combined Gini Impurity Index, which both indicate the impurity of the nodes (Kingsford & Salzberg, 2008). The difference between the Gini Impurity Index and the Combined Gini Impurity Index is that the former is calculated prior to the splitting of nodes whereas the latter is calculated by using the split nodes after splitting is completed. The formula for the Gini Impurity Index is:

Where Pc is the percentage of data in node A belonging to category c of the nominal target

Whereas the formula for the Gini Impurity Index is:

Where is the number of data observations in nodes A, B and C respectively, where node B and C are split from node A.

Thus, the last step is to calculate the reduction in Gini Impurity Index by subtracting the Combined Gini Impurity Index from the Gini Impurity Index and the formula is:

Therefore, the split with the largest reduction in the Gini Impurity Index will be considered optimal and will be selected.

The second criterion used by CART is entropy, which is the measurement of the amount of uncertainty that the dataset has (Kingsford & Salzberg, 2008). The higher the entropy value, the larger the amount of uncertainty the dataset has. The concept of entropy works similarly to the concept of the Gini Impurity Index, in which there is also entropy and combined entropy. The formulas for the entropy and combined entropy are:

And

Hence, I can obtain the reduction in entropy by subtracting combined entropy from entropy . Therefore, the split with the largest reduction in entropy will be optimal and will be selected.

On the other hand, an Artificial Neural Network (ANN) is a computational model that seeks to emulate the function of the human brain. A typical ANN model consists of a web of interconnected layers of nodes (Kuhn & Johnson, 2013). A typical ANN has an input, hidden and output layer, as shown in figure 4.4.2.1.

A diagram of a layer

Description automatically generated

*Figure 4.4.2.1. A Single ANN Connection*

According to figure 4.4.2.1, represents the th input node,  represents the th hidden node,  represents the th output node, represents the weight set between the and nodes and the represents the weight set between the and nodes.

The ANN works by having a procedure of forward-feeding, where the input layers pass information to the hidden nodes and subsequently to the output nodes by using various mathematical formulas. After the forward-feeding procedure, the ANN then updates the weights via a backpropagation algorithm. An iteration is considered complete when the ANN does one forward-feeding and backpropagation algorithm procedure (Kuhn & Johnson, 2013). The iterations will continue until all observations are processed or when stopping rules are met.

For the forward-feeding procedure, the input node passes the information it received to the hidden node, where the hidden node combines the information received from other input nodes using the formula . The hidden node then processes the information using the formula . After processing the information, the information is passed onto the output node, where it further combines the information received from other hidden nodes using the formula . The information is then processed and outputted using the formula .

After the forward-feeding procedure has been completed, the ANN then backpropagates to update the weight. First, the prediction error is computed at the output layer using the formula , where represents the predicted output value and represents the actual output value. After that, the signal error is computed at the hidden node using the formula . The weights between the hidden and output layers are then updated using the formula , where the represent the learning rate ranging from 0 to 1. Finally, the weights between the input and hidden layers are updated using the formula .

For our case study, I will use the Python libraries from scikit-learn shown in figure 4.4.1 to construct predictive models. Furthermore, as scikit-learn library is large in size, I decide to import specific modules for easy maintainability of the program code.

A screenshot of a computer program

Description automatically generated

*Figure 4.4.1. importing Python libraries for predictive modelling*

The purpose of these modules is listed below:

* The pyplot module from the matpltlib library is used to plot graphs for visualisation purposes.
* The train\_test\_split function from the scikit-learn library’s module\_selection module is used to partition the dataset into a training dataset and a testing dataset.
* The GridSearchCV function from the scikit-learn library’s module\_selection module is used to search for the best parameter values for the predictive model that has the best score.
* The DecisionTreeClassifier and MLPClassifier function from the scikit-learn library’s tree and neural\_network modules are used to construct decision trees and ANN models respectively.
* The BaggingClassifier, AdaBoostClassifier and RandomForestClassifier functions from the scikit-learn library’s ensemble module are used to construct various ensemble models.

### Pre-Modelling Preparation

Before I start constructing the predictive models, I first have to partition our dataset, as shown in Figure 4.4.1.1 below.

A screenshot of a computer code

Description automatically generated

*Figure 4.4.1.1. Preparing the dataset for predictive modelling*

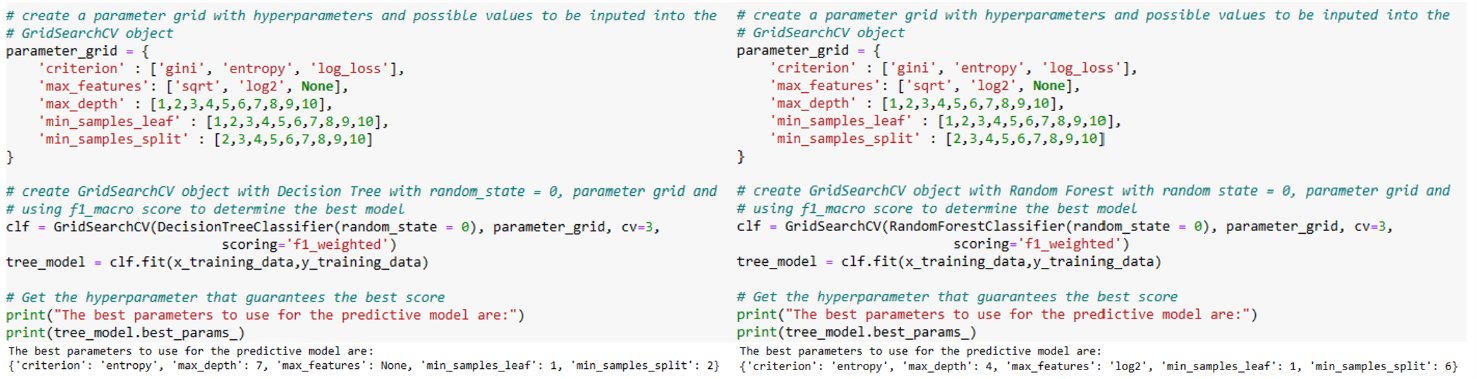
The purpose of partitioning the dataset is so that the predictive model can be evaluated for its true performance using unseen data. As the predictive model is constructed using a portion of data set aside for training, the model will not use the portion of data that has been set aside for testing. Therefore, by letting the model predict the outcomes based on the testing data, I can evaluate the model performance more accurately and ensure that the model is not overfitted.

According to Figure 4.4.1.1, I first partition the data using train\_test\_split function to set aside 80% of the data for training the predictive model and the rest of the data for testing the predictive model. After that, I further separated the data into a dataset containing the input or independent variables and a dataset containing just the target or dependent variable for both the training and testing datasets. This is so that I can fit the data into the predictive model and for it to learn and predict its features.

After I am done with the pre-modelling preparation, I can start constructing the predictive models.

### Building Predictive models

Before I build the predictive models, I will need to decide how to tune the parameters of predictive models so that I can get the best results. To do this, I have decided to accomplish this by using GridSearchCV, as shown in figure 4.2.1.1 below.



*Figure 4.2.1.1. Using GridSearchCV For Decision Tree and Random Forest*

According to Figure 4.2.1.1, GridSearchCV works by first taking a predictive model object and a parameter grid consisting of a dictionary containing the parameters of a predictive model as the key of the dictionary and the possible values of the parameters as the value of the dictionary. GridSearchCV then constructs the predictive model with every combination of the parameter values to find the combination of parameter values that guarantees the highest possible score. In this case, our preferred scoring method will be the f1\_weighted score.

A close-up of a computer screen

Description automatically generated

*Figure 4.2.1.2. Using GridSearchCV for Decision Tree with Bagging And with AdaBoost*

For building decision trees using Bagging and AdaBoost algorithms, I will also use a similar parameter grid with an additional step of adding the 'estimator\_\_' string to the parameters as shown in Figure 4.2.1.2 above. This allows us to access and tune the decision tree’s parameters when the decision tree is designated as the ensemble model’s estimator. Similarly, I will also repeat the process with ANN, as shown in Figure 4.2.1.3 below.

A screenshot of a computer code

Description automatically generated

*Figure 4.2.1.3. Using GridSearchCV For ANN And ANN With Bagging*

After using the GridSearchCV, I can then use the displayed optimal parameter values to construct the predictive models, as shown in Figure 4.2.1.4 below.



*Figure 4.2.1.4. Building Predictive Models with Tuned Parameters*

After I have finished building all our predictive models, I can proceed to evaluate the performance of all the models I have constructed.

## Phase 5: Evaluation of Predictive Models

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | True | False |
| Actual | True | TP | FN |
| False | FP | TN |

*Table 4.5.1. Example of Confusion Matrix*

For the evaluation of the performance of all the predictive models I have constructed, I choose to use the F1 score. As Lipton et al (2014) state, the F1 score is calculated using a confusion matrix, which according to Hong & Tae (2014), is a 2 × 2 table that measures the model’s predictive power. An example of a confusion matrix is shown in Table 4.5.1, where:

* True Positives (TP) is the number of observations the model predicted to be true is actually true.
* True Negatives (TN) is the number of observations the model predicted to be false is actually false.
* False Positives (FP) is the number of observations the model predicted to be true but is actually false.
* False Negatives (FP) is the number of observations the model predicted to be false but is actually true.

Using the information presented in the confusion matrix, I will be able to calculate two main classification metrics, which are Precision and Recall. Precision refers to the proportion of all predicted positives being true positives, whereas recall refers to the proportion that all actual positives are predicted positive (Lipton et al, 2014). The formulas are:

and

Using these two formulas, I can derive the F1 score by expressing it as the harmonic mean of

precision and recall:

Or by substitution in which the equation will be changed to:

Thus, the F1 score provides a more balanced evaluation of the model’s performance than just using precision or recall by considering both metrics and averaging them. An F1 score that is above 0.7 indicates that the overall performance of the model is good, whereas an F1 score that is below 0.5 indicates that the overall performance of the model is unsatisfactory. Therefore, the higher the value of the F1 score, the better the model is in its predictions.

However, as there are class imbalances present in the data, I will instead be using the weighted F1 score to evaluate the model’s performance as this variation of the F1 score takes account of class imbalances by averaging the class while considering the classes’ support, which refers to the number of actual occurrences of a class in the dataset.

Furthermore, due to class imbalances, the classification threshold will not be exactly at 0.5. To calculate the classification threshold to accurately evaluate the model’s performance, I will use the Receiver Operating Characteristic (ROC) curve, which is the plot of the True Positive Rate (TPR) against the False Positive Rate (FPR) and shows the model’s performance at various thresholds (Gonçalves et al, 2014). An example of a ROC curve is shown in figure 4.5.1 below.

A graph with a line

Description automatically generated with medium confidence

*Figure 4.5.1. An example of a ROC curve*

To find the optimal classification threshold, one way to do it is to find the minimum threshold that satisfies the equation of TPR = 1 – FPR (Hong and Tae, 2021).

A screenshot of a computer

Description automatically generated

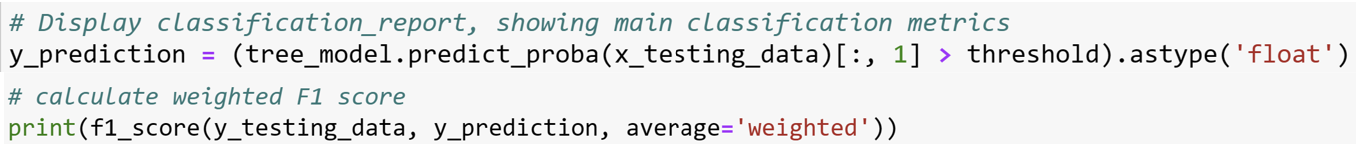
*Figure 4.5.2. Calculation of FPR, TPR and Classification Threshold Using Python*

According to Figure 4.5.2, I can calculate the FPR, TPR and threshold at the same time using the roc\_curve function. As the thresholds are contained with a Pandas Series and acts as an array, I can use the np.argmin function to find the minimum threshold value that satisfies the equation of FPR + TPR – 1. The optimal classification threshold for each model is shown in Table 4.5.2 below.

|  |  |
| --- | --- |
| **Model** | **Classification Threshold** |
| CART Tree | 0.5828220858895705 |
| CART Tree with Bagging | 0.5762366842128606 |
| CART Tree with AdaBoost | 0.5022421872302094 |
| Random Forest | 0.5187540405727532 |
| ANN | 0.4672184094689751 |
| ANN with Bagging | 0.6014611377556034 |

*Table 4.5.2.* *Individual Model’s Optimal Classification Threshold*

After obtaining each model’s optimal classification threshold, I can change the threshold manually by accessing the threshold attributes in the model’s prediction using the predict\_proba method, as shown in figure 4.5.3. After updating the threshold, I can proceed to calculate the weight F1 score.



*Figure 4.5.3. Updating Threshold and Calculating the Weighted F1 Score*

After calculating the weight F1 score, I can then proceed to evaluate the performance of each model. The weight F1 score of each model are shown in Table 4.5.3.

|  |  |
| --- | --- |
| **Model** | **Weighted F1 Score** |
| CART Tree | 0.7404867267407613 |
| CART Tree with Bagging | 0.7401931617925779 |
| CART Tree with AdaBoost | 0.736094674556213 |
| Random Forest | 0.7357190635451505 |
| ANN | 0.7250588381023163 |
| ANN with Bagging | 0.7623668639053256 |

*Table 4.5.3. Individual Model’s Weighted F1 Score*

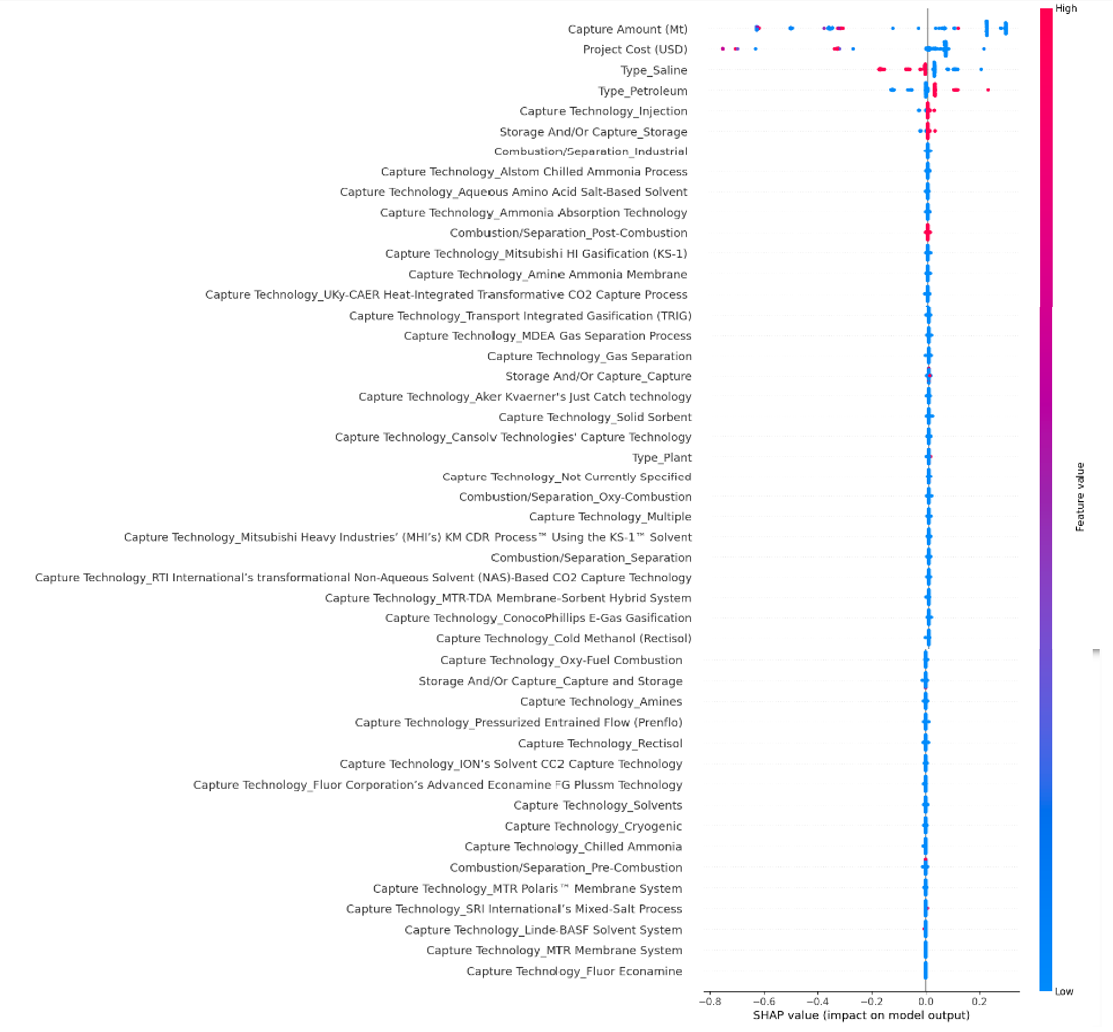
Based on the results shown in table 4.5.3, the model that performs the best is the ANN model with Bagging as it has the highest weighted F1 score among all the other predictive models.

### Evaluation of Results

As the ANN model with Bagging has the highest F1 score among all the other models, I am going to evaluate the results of the model. However, as the ANN model is a black box that does not reveal any information about its inner workings, it is often difficult to interpret the results of the model.

Despite this, there are a few ways to interpret the results of ANN. According to Yang et al (2019), there are two ways to interpret an ANN model, the first method is to figure out the workings of the ANN model as well as the rules the model came up with at a global level, whereas the second method focuses on a few observations and explain the reason why certain decisions are made by ANN model at a local level (Yang et al, 2019). Thus, the researchers developed a Python library known as XDeep to interpret the results of ANN using both methods. For this case study, I decide to use a similar Python library known as SHAP to interpret the ANN model.

SHAP is a Python library that is developed based on cooperative game theory, in which each variable contributes their part to the final model and the program calculates the contributions of each variable to the final model before displaying its value. Thus, the higher the values, the more the variable contributes to the final model and it will be placed near the top of plots that SHAP provides. According to figure 4.5.1.1, the summary plot shows how the value of the variables can affect the prediction of the final model. The variables that have most contributions are the capture amount of CO2, the project cost, the saline facility type, the petroleum facility type, and the amount of CO2 being captured, as shown in figure 4.5.1.2.



*Figure 4.5.1.1. Summary Plot of ANN Model*

A screen shot of a computer

Description automatically generated

*Figure 4.5.1.2. Significant Variables in Model*

According to figure 4.5.1.1, the summary plot shows that:

* If the facility captures a huge amount of CO2, this will contribute negatively to the final model. Whereas if the facility captures a moderate amount of CO2, this will contribute positively to the final model.
* If the facility costs a lot to build, this will contribute negatively to the final model. Whereas if the costs to build the facility are inexpensive, this will contribute positively to the final model.
* If the facility is a saline aquifer, this will contribute negatively to the final model. Whereas if the facility is not a saline aquifer, this will contribute positively to the final model.
* If the facility is not an oil field, this will contribute negatively to the final model. Whereas if the facility is an oil field, this will contribute positively to the final model.
* If the facility does not use Injection as a capture technology, this will contribute negatively to the final model. Whereas if the facility uses Injection as a capture technology, this will contribute positively to the final model.
* If the facility does not or not only capture CO2, this will contribute negatively to the final model. Whereas if the facility only captures CO2, this will contribute positively to the final model.

To further aid in the interpretation of the ANN model, I decide to use an interpretable predictive model offers a similar performance to determine the decision rules that the ANN makes. Thus, I decide to use the second-best performing model, which is the CART Tree as shown in figure 4.5.1.3 below:

A diagram of a computer flowchart

Description automatically generated

*Figure 4.5.1.3. Illustration of the CART Model*

According to Figure 3.5.1.3, the CSS project will be feasible if:

* The facility is an oil field and the amount of CO2 captured is less than or equal to 8545 metric tons.
* The facility is not an oil field, uses injection technology and the amount of CO2 captured is more than 83.405 metric tons but is less than or equal to 2830 metric tons.
* The facility is not an oil field, does not use injection technology or Chilled Ammonia Technology and the amount of CO2 captured is more than 0.057 metric tons but is less than or equal to 4905 metric tons.
* The facility is not an oil field, the amount of CO2 captured is more than 4905 metric tons, involves only in the capture of CO2 and the project costs less than or equals to US$1056249984.
* The facility is not an oil field, the amount of CO2 captured is more than 1.05 metric tons but is less than or equals to 9.9 metric tons and uses injection technology.
* The facility is not an oil field, the amount of CO2 captured is more than 0.057 metric tons but is less than or equals to 0.651 metric tons and uses injection technology.
* The facility is an oil field, the amount of CO2 captured is more than 9279 metric tons but is less than or equal to 11096.115 metric tons.
* The facility is not an oil field, the amount of CO2 captured is more than 17170 metric tons, does not involve only in the capture and separation of CO2 and the project costs more than US$17644000 but is less than or equal to US$67500000.
* The facility is an oil field, the amount of CO2 captured is more than 12587 metric tons but is less than or equal to 11096.115 metric tons.
* The facility is not an oil field, the amount of CO2 captured is more than 4905 metric tons, does not involve only in the capture but involves in the separation of CO2.
* The facility is an oil field, the amount of CO2 captured is more than 87442.5 metric tons.
* The facility is not an oil field, the amount of CO2 captured is more than 4905 metric tons, involves only in the capture of CO2 and the project costs more than US$347500000 but is less than or equal to US$1056249984.
* The facility is not an oil field, the amount of CO2 captured is more than 4905 metric tons, involves only in the capture of CO2 and the project costs more than US$2730000000.
* The facility is not an oil field, the amount of CO2 captured is less than or equals to 0.057 metric tons and does not involve post-combustion to capture the CO2.

Conversely, the CSS project will not be feasible if:

* The facility is not an oil field, the amount of CO2 captured is more than 41450 metric tons, does not involve only in the capture and separation of CO2 and the project costs less than or equal to US$17644000.
* The facility is not an oil field, the amount of CO2 captured is more than 17170 metric tons, does not involve only in the capture and separation of CO2 and the project costs less than or equals to US$67500000.
* The facility is not an oil field, the amount of CO2 captured is more than 9.9 metric tons but is less than or equal to 83.405 metric tons and uses injection technology.
* The facility is not an oil field, the amount of CO2 captured is more than 17170 metric tons but is less than or equals to 41450 metric tons, does not involve only in the capture and separation of CO2 and the project costs less than or equals to US$17644000.
* The facility is not an oil field, the amount of CO2 captured is more than 5650 metric tons but is less than or equals to 17170 metric tons, does not involve only in the capture and separation of CO2 and the project costs less than or equals to US$414850000.
* The facility is not an oil field, the amount of CO2 captured is less than or equals to 0.057 metric tons and uses a post-combustion process to capture of CO2.
* The facility is not an oil field, the amount of CO2 captured is more than 3229.5 metric tons but is less than or equal to 4905 metric tons and uses injection technology.
* The facility is not an oil field, the amount of CO2 captured is more than 4905 metric tons but is less than or equals to 5650 metric tons and does not involve only in the capture and separation of CO2.
* The facility is not an oil field, the amount of CO2 captured is more than 4905 metric tons, involves only in the capture of CO2 and the project costs more than US$1056249984 and less than or equal to US$2730000000.
* The facility is not an oil field, the amount of CO2 captured is more than 2830 metric tons but is less than or equal to 3229.5 metric tons and uses injection technology.
* The facility is an oil field and the amount of CO2 captured is more than 24588 metric tons but is less than or equal to 87442.5 metric tons.
* The facility is not an oil field, the amount of CO2 captured is more than 5650 metric tons but is less than or equal to 17170 metric tons, does not involve only in the capture and separation of CO2 and the project costs more than US$414850000.
* The facility is not an oil field, the amount of CO2 captured is more than 4905 metric tons, involves only in the capture of CO2 and the project costs more than US$1056249984 and less than or equal to US$347500000.
* The facility is not an oil field, the amount of CO2 captured is more than 0.651 metric tons but is less than or equal to 9.9 metric tons and uses injection technology.
* The facility is not an oil field, does not use injection technology but uses Chilled Ammonia Technology and the amount of CO2 captured is more than 0.057 metric tons but is less than or equal to 4905 metric tons.
* The facility is an oil field, the amount of CO2 captured is more than 8545 metric tons but is less than or equal to 9279 metric tons.
* The facility is an oil field, the amount of CO2 captured is more than 11096.115 metric tons but is less than or equal to 12587 metric tons.

## Phase 6: Deployment of Predictive Model

### Recommendations

Based on the insights we gained from the two predictive models, the best option is to use an oil field to store CO2 by injecting 8545 metric tons of pressurized CO2 into it. However, since it may not be the most practical option in some cases, including not having oil fields located in the country, I will also make recommendations for the rest of the facility types.

For saline aquifers, the best option is to store between 83.405 and 2830 metric tons of CO2 by injecting the pressurised CO2 into it, whereas for constructing CCS plants, the best option build dedicated facilities to capture the CO2, for the project to cost less than or equals to US$1056249984 and to capture the more than 4905 metric tons of CO2 but not storing them on-site.

### Discussion

Out of the recommendations for the three types of facilities, the usage of oil fields is the most practical as these oil fields are partially or completely devoid of fossil fuels. This means that these oil fields will have the capacity to store pressurized CO2 indefinitely up to its full capacity. Furthermore, the costs of using the oil fields as a storage site for CO2 will be inexpensive as compared to building dedicated facilities to capture and store the CO2 on-site.

However, using saline aquifers to store CO2 is a less economical option as there are as the storage capacity of saline aquifers is difficult to ascertain. According to CSRC, the storage capacity of saline aquifers is determined using hydrocarbon exploration. With hydrocarbon exploration, the specified area around wells within the saline aquifers that have been explored is considered to be usable, whereas the rest of the undrilled portions of the saline qualifiers will be considered unusable. This means that the storage capacity of saline qualifiers will be less than that of their oil field counterparts.

Lastly, building dedicated CCS facilities to store and capture CO2 is the least economical option as if companies want to build CCS facilities that can capture and store huge amounts of CO2, it will cost upwards of one billion USD, as analysed by the predictive model, in which more than half of the costs will go to constructing storage facilities to store the CO2. However, despite this, dedicated CCS facilities will be required to capture the CO2 from industrial processes. Therefore, the best option will be to construct dedicated CCS facilities to capture the CO2 and to ship the pressurized CO2 to the nearest oil field for storage.

### Next Steps

With the recommendations and the trained ANN model, companies can take into consideration the recommendations when designing and implementing the CSS project. Alternatively, one can develop a Graphical User Interface (GUI) program that will be embody the ANN model into its core programming. Thus, companies can use the GUI program by inputting the parameters into the program and have the program to predict the outcome. This will let the companies gauge whether their project will be feasible based on predictions made by the GUI program.

However, as the amount of information regarding existing and upcoming CCS projects will be increasing year on year, it is important to update the predictive model at least once a year by providing new information.

# Conclusion

By predicting the outcome of CCS projects obtained from public databases using machine learning models, one can obtain the decision rules and understand the rationale behind the reason why some CCS projects are feasible while other CCS projects are not feasible.

However, as there are different types of predictive models that are black boxes, it will be difficult to decipher its inner workings and obtain the decision rules. Therefore, one can use explainable machine learning techniques, including the SHAP python library, to uncover the main features of these models and to discover how each of the model’s features interact to produce the final outcome through the use of local and global analysis of the model. Alternatively, one can use interpretable predictive models that offers a similar performance to obtain the decision rules.

This information will prove valuable to companies as these companies can use the information to make informed decisions, thus saving time and project costs. Therefore, more of such CSS projects can be build and contribute to mitigating the effects of global warming.

# References

Global Annual to Decadal Climate Update. (2023, May 17). World Meteorological Organisation. Retrieved June 29, 2023, from <https://library.wmo.int/doc_num.php?explnum_id=11629>

Haines, A., & Patz, J. A. (2004). *Health effects of climate change*. Jama, 291(1), 99-103.

Wirth, R., & Hipp, J. (2000, April). *CRISP-DM: Towards a standard process model for data mining.* In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (Vol. 1, pp. 29-39).

Boot-Handford, M. E., Abanades, J. C., Anthony, E. J., Blunt, M. J., Brandani, S., Mac Dowe

ll, N., ... & Fennell, P. S. (2014). *Carbon capture and storage update*. Energy & Environmental Science, 7(1), 130-189.

Bui, M., Adjiman, C. S., Bardow, A., Anthony, E. J., Boston, A., Brown, S., ... & Mac Dowell, N. (2018). *Carbon capture and storage (CCS): the way forward*. Energy & Environmental Science, 11(5), 1062-1176.

Haszeldine, R. S. (2009). *Carbon capture and storage: how green can black be?.* Science, 325(5948), 1647-1652.

Gonçalves, L., Subtil, A., Oliveira, M. R., & de Zea Bermudez, P. (2014). *ROC curve estimation: An overview*. REVSTAT-Statistical journal, 12(1), 1-20.

Song, Y. Y., & Ying, L. U. (2015). *Decision tree methods: applications for classification and prediction*. Shanghai archives of psychiatry, 27(2), 130.

Kingsford, C., & Salzberg, S. L. (2008). *What are decision trees?*. Nature biotechnology, 26(9), 1011-1013.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26). New York: Springer.

Lipton, Z. C., Elkan, C., & Naryanaswamy, B. (2014). *Optimal thresholding of classifiers to maximize F1 measure*. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14 (pp. 225-239). Springer Berlin Heidelberg.

Hong, Chong Sun, and Tae Gyu Oh. *TPR-TNR plot for confusion matrix*. Communications for Statistical Applications and Methods 28.2 (2021): 161-169.

Yang, F., Zhang, Z., Wang, H., Li, Y., & Hu, X. (2019). *Xdeep: An interpretation tool for deep neural networks*. arXiv preprint arXiv:1911.01005.