Leeds University
Business School

UNIVERSITY OF LEEDS

Assessed Coursework Coversheet

For use with *individual* assessed work

Student ID Number:	2	0	1	4	8	4	4	8	4
Module Code:	LUBS5990M								
Module Title:	Machine learning in practice								
Module Leader:	Xingjie Wei								
Declared Word Count:	2950								

Please Note:

Your declared word count must be accurate, and should not mislead. Making a fraudulent statement concerning the work submitted for assessment could be considered academic malpractice and investigated as such. If the amount of work submitted is higher than that specified by the word limit or that declared on your word count, this may be reflected in the mark awarded and noted through individual feedback given to you.

It is not acceptable to present matters of substance, which should be included in the main body of the text, in the appendices ("appendix abuse"). It is not acceptable to attempt to hide words in graphs and diagrams; only text which is strictly necessary should be included in graphs and diagrams.

By submitting an assignment you confirm you have read and understood the University of Leeds **Declaration of Academic Integrity** (http://www.leeds.ac.uk/secretariat/documents/academic_integrity.pdf).

Predicting Success of ICO Project

1. Introduction:

Cryptocurrency is a digital payment mechanism that does not rely on banks for transaction verification. It is a digital or virtual currency that is encrypted, making counterfeiting and double-spending nearly impossible. Several cryptocurrencies use blockchain technology to construct decentralized networks. It's a peer-to-peer system that allows anybody to make and receive money from anywhere. A blockchain database is a form of database that is unique. Blockchains store data in blocks that are then connected together, unlike traditional databases. As new information is received, it is entered into a new block. Every block in the chain is connected to the previous block. So, it is impossible to hack, and it is a totally secure network. One of the most popular cryptocurrencies is Bitcoin, which is invented in 2009. (tusharpedia, n.d.)It is used worldwide.

More recently, a company has produced currencies based on several blockchains in order to gather cash from a big audience of people for the development of a project. This is called initial coin offering (ICO), which is a method of raising cash for a company aiming to establish a new currency, app, or service. The 'All-or-Nothing' approach is used by the majority of ICO crowdfunding campaigns, in which the ICO team sets a fundraising target (e.g., raise $1 million in 30 days). The ICO is deemed a success if they were able to reach its funding objective. Otherwise, the fundraiser will be a flop, and they will receive nothing. The main task in this problem is to predict whether the ICO project is a success or not. In this report, we are going to validate and interpret the performances of different models and choose the best model that fits this problem.

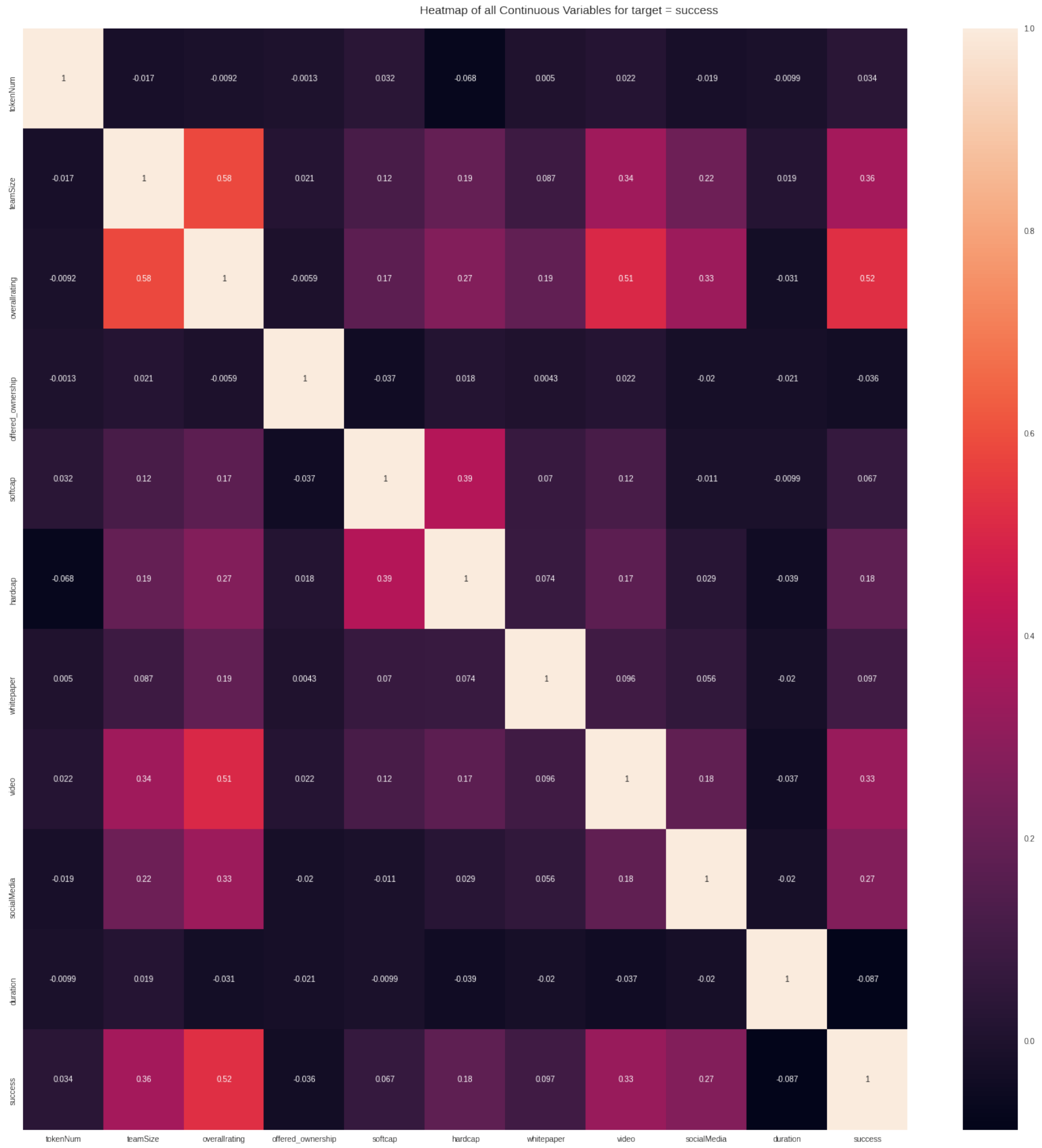
For this project, they have given a dataset with many columns that contain information about every ICO project. This dataset contains columns of ID, which is a unique ID of the ICO project, start date, end date, the token name is for the name of the token issued by ICO project, the country is where the ICO project is located, categories, ownership, token price, token name and etc. In total, the dataset contains 20 columns and 1606 rows. These columns are used to create different models that help to predict the success of the ICO project.

2. Data understanding and Data Preparation:

We are going to use Jupiter notebook to edit and run the python documents. It is a **server-client application.** To start with this project, the dataset is imported by using the commend read_csv and store in the "data" variable. This will load the CSV file in the notebook that will make the dataset ready to use. As there are a lot of columns in a CSV file, that will distract us from the prediction. So, unnecessary columns like ID, Token name, Token price, Token type are dropped. To prepare data for modeling, datatype and new columns are added. Duration of the ICO project will help to predict more efficiently, So, here to find the duration of the ICO project, both start date, and end date are used. Both the start and end date columns are converted to the DateTime data type. The end date column and start date column get subtracted so that we will get the duration of the ICO project. This duration column is the number of days that the ICO project takes place.

When choosing data, the NULL value might create issues. However, because the outcome is always unknown when comparing an unknown value to any other value, it is not included in the results. To check for the null value in the dataset IsNull() function is used. By using this function in the dataset, we came to know columns contain null values. This will affect our end result. There are many ways to handle the null values. The columns Token number, Soft cap, Hard cap, Whitepaper, video, Social media contains null values, and it is of numerical values. So null value is replaced by the mean of each column by the function fillna(). Now all the null values are filled, and the dataset is ready to go for the next process.

Autoviz is a python library that helps us to create a powerful chart and visualization for our dataset. By using this function, we got a helpful plot that helps us to understand the column dependency. Heatmap of all continuous variables for the target, and this will show how columns are related.



The above heatmap compares each column with other columns in the dataset and gives the comparative value. This value tells how much the columns are related. As the value is high, the column is more related. As the value is low, the column is less related. The Colour of the map is related to its value. When the value gets increases colour gets lighter. When the value gets decreases color gets darker.

The success column is our output column, so we are comparing this column with each other columns. In the heat map, we can see duration and ownership columns have negative values in success. This means these two columns are not related to the success column, and so it will affect the end result. To avoid such kinds of problems, both the columns are dropped from the dataset. Also, the start date and end date are not needed anymore in the dataset because we got the duration from this. So, we are going to drop these columns. Now dataset is straightforward and has only the columns which are needed.

For machine learning, it is unable to handle string values directly. We must first translate your nominal characteristics into integers if they are strings. In our dataset, there are three columns that contain string values. To convert these string values to integer values, get_dummies() function is used. This function is used to manipulate the data. It makes dummy or indicator variables out of categorical data.

For all the columns containing a string, we get the dummies and join the dummies columns to the dataset and remove the column we used to get dummies. By this, the categorical variable is converted to a dummy column, and the data preparation process is done. After all these steps dataset contains 205 columns and 1606 rows.

3. Modelling:

Modelling entails teaching a machine learning system to predict labels based on features. A machine learning model (ML model) is a mathematical model that makes predictions by identifying patterns in data. You train a model on a collection of data and give it an algorithm to use to reason about and learn from that data. (machinelearningmastery, n.d.) The learning algorithm searches the training data for patterns that relate the input data characteristics to the goal and then generates an ML model that captures these patterns. The ML model may be used to make predictions on new data for which the target is unknown. In machine learning, the algorithm is a method that is executed on data to generate a model. Pattern recognition is performed by machine learning algorithms, which learn from data or fit into a dataset. There are many algorithms. For instance, we have regression techniques, such as linear regression, and clustering methods, such as k-means.

To start with our modelling, we are dividing the dataset into two datasets. One dataset acts as an input dataset, and another dataset will act. As a result dataset. For a result dataset, we are keeping only the success column which contains zeros and ones, which means zero is failed, and one is successes of the ICO project. And for the input dataset, we are dropping the success column and keeping all the rest columns. Let X be the input dataset and Y be the output dataset.

Now, we need training and testing datasets. So, we need to split the primary dataset into a training dataset and a test dataset. So that we can use the training dataset to train the model and the testing dataset to cross-validate the model. To split the dataset, we are importing train_test_split from sklearn.model_selection library. By applying train_test_split function on both X and Y datasets with a test size of 30%. This function will split the 30% data of X and Y into the test dataset. At the ending of the function, we will get four datasets X_train, X_test, y_train, y_test. These four datasets are going to be used in modelling.

Values in the dataset are too large. These values may take more time to process the modelling. Without affecting the end result, we need to scale the value of this column. For that, StandardScaler is imported from sklearn.preprocessing. By using this, the mean will be removed, and each feature/variable will be scaled to unit variance. So, data is scaled and can be used with better efficiency.

In modelling, the accuracy score tells how the model works efficiently. The percentage of correct predictions for the test data is known as accuracy. It's simple to figure out simply by dividing the number of correct guesses by the total number of forecasts. Using the accuracy score, we will find the best model for this problem.

For this problem, we will test the prepared dataset with different machine learning algorithms and find the best one that suits and works efficiently.

a) Logistic Regression:

First, The Logistic Regression model is used in statistics to represent the likelihood of a specific class or event, such as pass/fail, win/lose, alive/dead, or healthy/sick, existing. This may be used to simulate a variety of occurrences, such as identifying whether a picture contains a cat, dog, lion, or other animals. (scikit-learn, n.d.) To apply this algorithm, we are importing LogisticRegression from sklearn.linear_model library. Then, fit the X_train_std and y_train set in Logistic Regression by using fit() function. Now model got trained. To check how the model is working, we will use the predict function on X_test_std. This will predict and gives the output dataset. By comparing the predicted output and the y_test dataset that we have, we will get the accuracy score. For Logistic Regression, the accuracy score is 0.724. That is, this model predicated 72.4% correctly.

b) K Neighbors Classifier:

The k nearest neighbours are represented by the K in the classifier’s name, where k is an integer number supplied by the user. As the name implies, this classifier uses learning based on the k closest neighbours. (tutorialspoint, n.d.) To apply this algorithm, we are importing KNeighborsClassifier from sklearn.neighbors library. Then, fit the X_train_std and y_train set in K Neighbors Classifier by using fit() function. Now model got trained. To check how the model is working, we will use the predict function on X_test_std. For K Neighbors Classifier accuracy score is 0.63. That is, this model predicated 63% correctly.

c) Support vector machines:

The SVM, or Support Vector Machine, is a linear model that may be used to solve classification and regression issues. It can handle both linear and nonlinear problems and is useful for a wide range of applications. SVM is a basic concept: The method divides the data into classes by drawing a line or hyperplane. (scikit-learn, scikit-learn, n.d.) To apply this algorithm, we are importing svm from the sklearn library. Then, fit the X_train_std and y_train set in Support vector machines by using fit() function. Now model got trained. To check how the model is working, we will use the predict function on X_test_std. For Support vector machines accuracy score is 0.734. That is, this model predicated 73.4% correctly.

d) Decision Tree Classifier:

A decision tree is all possible solutions to a problem based on a set of criteria. We aim to establish a condition on the features at each step or node of a decision tree used for classification to segregate all the labels or classes present in the dataset to the fullest purity. (scikit-learn, scikit-learn, n.d.) To apply this algorithm, we are importing tree from sklearn library. Then, fit the X_train_std and y_train set in Decision Tree Classifier by using fit() function. Now model got trained. To check how the model is working, we will use the predict function on X_test_std. This will predict and gives the output dataset. By comparing the predicted output and the y_test dataset that we have, we will get the accuracy score. For Decision Tree Classifier accuracy score is 0.718. That is, this model predicated 71.8% correctly.

e) Random Forest Classifier:

When developing each individual tree, random forest employs bagging and feature randomization in order to produce an uncorrelated forest of trees whose committee prediction is more accurate than that of any particular tree. (builtin, n.d.) To apply this algorithm, we are importing RandomForestClassifier from sklearn.ensemble library. Then, fit the X_train_std and y_train set in Random Forest Classifier by using fit() function. Now model got trained. To check how the model is working, we will use the predict function on X_test_std. For Random Forest Classifier accuracy score is 0.757. That is, this model predicated 75.7% correctly.

f) AdaBoost Classifier:

An AdaBoost classifier is a meta-estimator that fits a classifier on the original dataset before fitting further copies of the classifier on the same dataset but adjusts the weights of erroneously classified instances such that future classifiers focus more on difficult situations. (scikit-learn, scikit-learn, n.d.) To apply this algorithm, we are importing AdaBoostClassifier from sklearn.ensemble library. Then, fit the X_train_std and y_train set in AdaBoost Classifier by using fit() function. Now model got trained. To check how the model is working, we will use the predict function on X_test_std. For AdaBoost Classifier accuracy score is 0.759. That is, this model predicated 75.9% correctly.

g) Gradient Boosting Classifier:

Gradient boosting classifiers are a set of machine learning algorithms that integrate a number of weak learning models to build a powerful prediction model. When conducting gradient boosting, decision trees are commonly utilized. (scikit-learn, scikit-learn, n.d.) To apply this algorithm, we are importing GradientBoostingClassifier from sklearn.ensemble library. Then, fit the X_train_std and y_train set in Gradient Boosting Classifier by using fit() function. Now model got trained. To check how the model is working, we will use the predict function on X_test_std. For Gradient Boosting Classifier accuracy score is 0.759. That is, this model predicated 75.9% correctly.

h) XGB Classifier:

XGBoost is a machine learning and Kaggle competition method for structured or tabular data that has lately dominated. XGBoost is a high-speed, high-performance implementation of gradient boosted decision trees. (xgboost, n.d.) To apply this algorithm, we are importing xgb from xgboost library. Then, fit the X_train_std and y_train set in XGB Classifier by using fit() function. Now model got trained. This will predict and gives the output dataset. By comparing the predicted output and the y_test dataset that we have, we will get the accuracy score. To check how the model is working, we will use the predict function on X_test_std. For XGB Classifier accuracy score is 0.793. That is, this model predicated 79.3% correctly.

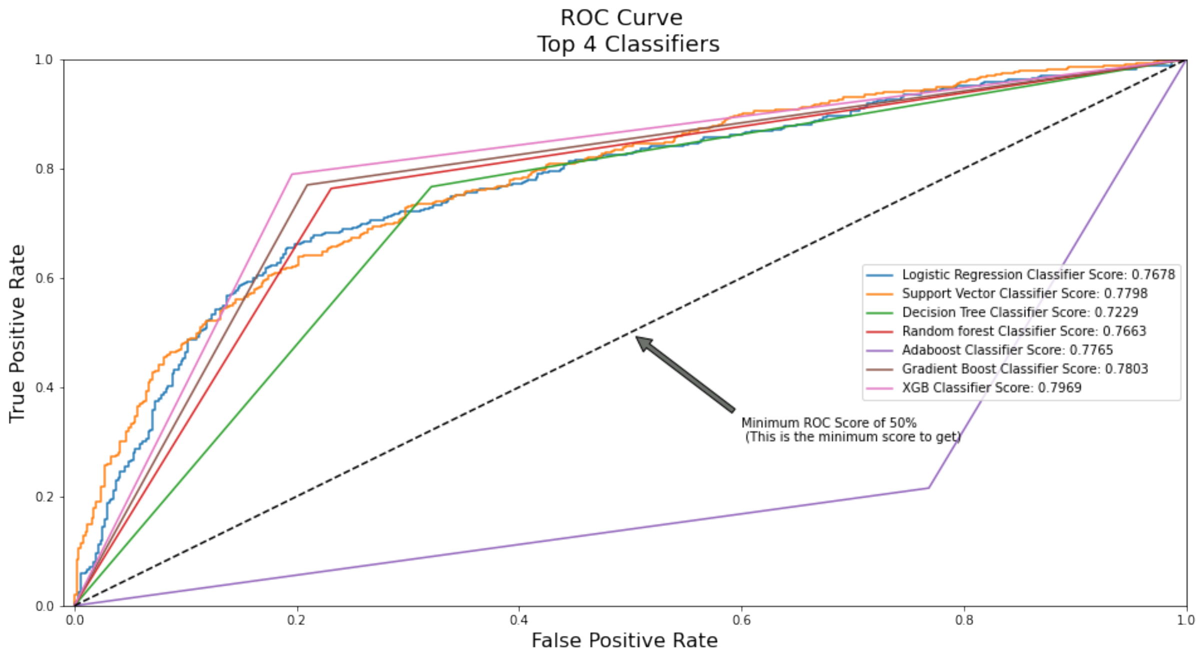
Accuracy Score for all the classifier

	Classifier	Accuracy Score
1	**Logistic Regression**	0.724
2	**K Neighbors Classifier**	0.63
3	**Support vector machines**	0.734
4	**Decision Tree Classifier**	0.718
5	**Random Forest Classifier**	0.757
6	**AdaBoost Classifier**	0.759
7	**Gradient Boosting Classifier**	0.759
8	XGB Classifier	0.793

The above table shows the accuracy score of all the classifiers for this dataset. In this, we can see XGB Classifier has the highest accuracy rate, which means this classifier is the best classifier of the eight classifiers.

ROC Curve:

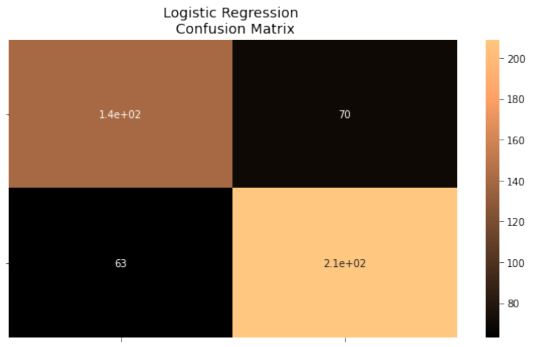
The True Positive Rate vs. False Positive Rate trade-off is depicted by the ROC curve. Classifiers with curves that are closer to the top-left corner perform better. The closer the curve gets to the ROC space's 45-degree diagonal, the less accurate the test becomes.

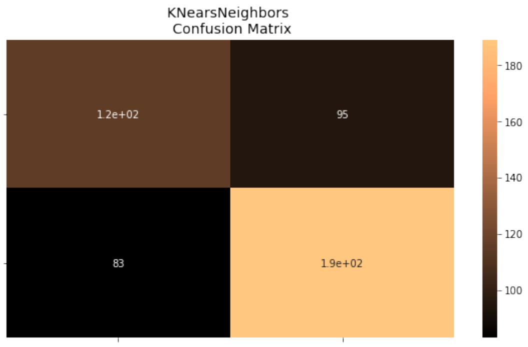


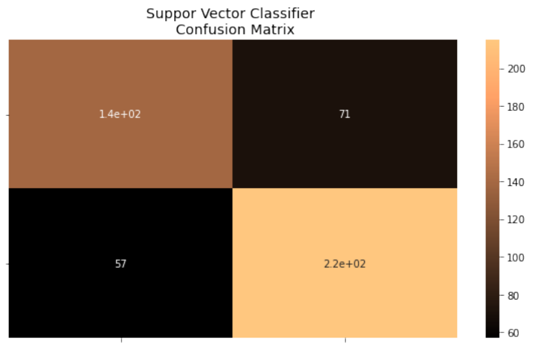
I have plotted the ROC curve with accuracy score for all the classifiers except K Neighbors Classifier because this accuracy score is very low compared to other classifiers. This char is divided into two equal half. The upper half is the True positive rate, and the lower half is the false positive rate. In all the seven lines, the XGB classifiers line is closer to the top left corner. So XGB classifier performance is the best. Followed by Gradient Boosting Classifier, which is the second best classifier.

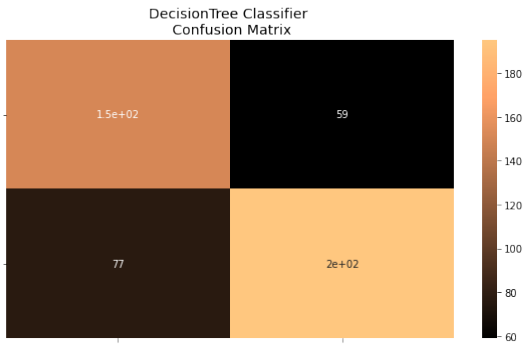
Confusion Matrix:

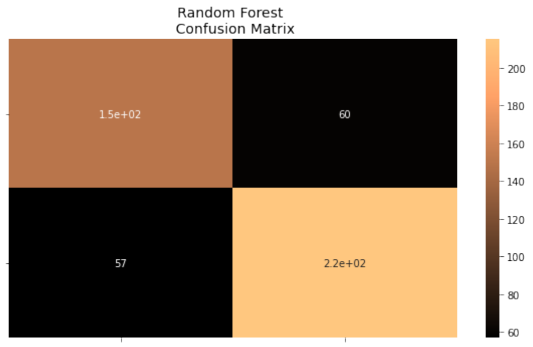
A table that describes how well a classification model or classifier performs on a set of test data for which the real values are known is called a confusion matrix.

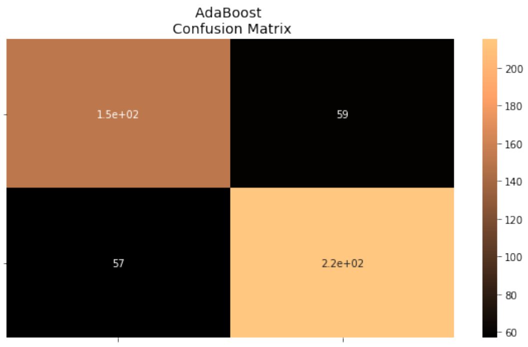


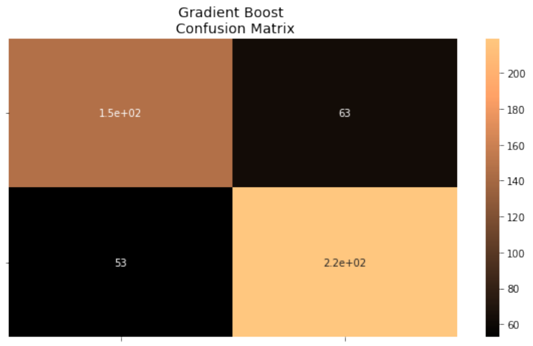


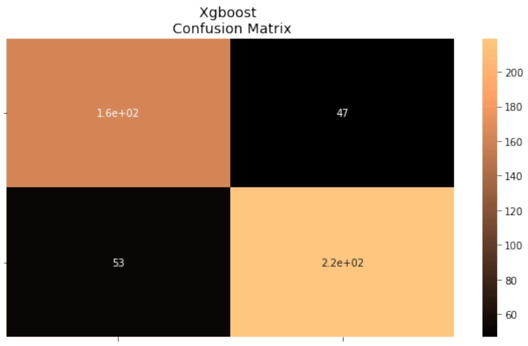












To plot the confusion matrix, I have import confusion_matrix from sklearn.metrics and then I used the function confusion_matrix() on the predicted result and y_test. This will give us the confusion matrix, and this confusion matrix is plotted on the heatmap. In the heatmap top left is for true positive, the top right is called false negative, the bottom left is called false positive, and finally, the bottom right is called a true negative. Heatmap shows values on each section. These values represent prediction results values.

4. Conclusion:

In this report, I have proposed many kinds of machine learning algorithms to find the best model to predict the success of the ICO project. From the models that I have done here, the XGB classifier is the best classifier that predicts the success of the ICO project with the highest accuracy rate and best performance. It is an accurate and scalable gradient-boosting machine implementation.

References:

builtin. (n.d.). *builtin*. Retrieved from builtin: https://builtin.com/data-science/random-forest-algorithm

machinelearningmastery. (n.d.). *machinelearningmastery*. Retrieved from machinelearningmastery: https://machinelearningmastery.com/difference-between-algorithm-and-model-in-machine-learning/

scikit-learn. (n.d.). *scikit-learn*. Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

scikit-learn. (n.d.). *scikit-learn*. Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/svm.html#:~:text=Support%20vector%20machines%20(SVMs)%20are,than%20the%20number%20of%20samples.

scikit-learn. (n.d.). *scikit-learn*. Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/tree.html#:~:text=Decision%20Trees%20(DTs)%20are%20a,as%20a%20piecewise%20constant%20approximation.

scikit-learn. (n.d.). *scikit-learn*. Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

scikit-learn. (n.d.). *scikit-learn*. Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

tusharpedia. (n.d.). *tusharpedia*. Retrieved from tusharpedia: https://tusharpedia.com/cryptocurrency/

tutorialspoint. (n.d.). *tutorialspoint*. Retrieved from tutorialspoint: https://www.tutorialspoint.com/scikit_learn/scikit_learn_kneighbors_classifier.htm

xgboost. (n.d.). *xgboost*. Retrieved from xgboost: https://xgboost.readthedocs.io/en/latest/python/python_api.html