Spacy Take Home Assignment

Processing and Exploratory Analysis:

First and foremost, this assignment contained an imbalanced dataset with 7950 positive cases, and 2089152 negative cases. This means that for every positive case, there are approximately 260 negative cases …

The following steps were performed:

- Load in the three excel files provided as data frames. The three data frames were merged based on their common categories.
- We inspect the datatypes of the different columns. Object and string type fields were converted to numerical representations to be easier to work with.

```
ID_O             int64
Object_Type      int64
Purchase_Date    int64
Object_Age       int64
BB_X1_O          int64
BB_Y1_O          int64
BB_X2_O          int64
BB_Y2_O          int64
ID_A             int64
Animal_Type      int64
Feeding_Time     int64
BB_Y2_A          int64
No_Animals       int64
No_Objects       int64
No_Farmers       int64
is_Raining        bool
Is_UFO            bool
dtype: object
```

Figure 1: Ensuring no string or object data types after processing

- Data frame columns were inspected for missing values. Columns for which missing values were found were attended to e.g. Object_Age and Purchase_Date.

```
ID_O             0.0
Object_Type      0.0
Purchase_Date    0.0
Object_Age       0.0
BB_X1_O          0.0
BB_Y1_O          0.0
BB_X2_O          0.0
BB_Y2_O          0.0
ID_A             0.0
Animal_Type      0.0
Feeding_Time     0.0
BB_Y2_A          0.0
No_Animals       0.0
No_Objects       0.0
No_Farmers       0.0
is_Raining       0.0
Is_UFO           0.0
```

Figure 2: Noting that after processing, all categories do not have empty values

- In order to analyse which features might be most beneficial for detecting UFOs, a correlation analysis was performed to inspect which features corresponded. Unfortunately there were no obvious features, however, features with very low correlation to the UFO were removed … e.g. > 0.001
- The data was split in train, validation and testing sets.
- In order to address the imbalance issue when training, we consider two options: under-sampling versus SMOTE. Scikit random under-sampler: Random under sampling involves randomly selecting examples from the majority class and deleting them from the training dataset. In the random under-sampling, the majority class instances are discarded at random until a more balanced distribution is reached. Ultimately under-sampling was selected as operating on the resultant smaller dataset was significantly less computationally expensive.

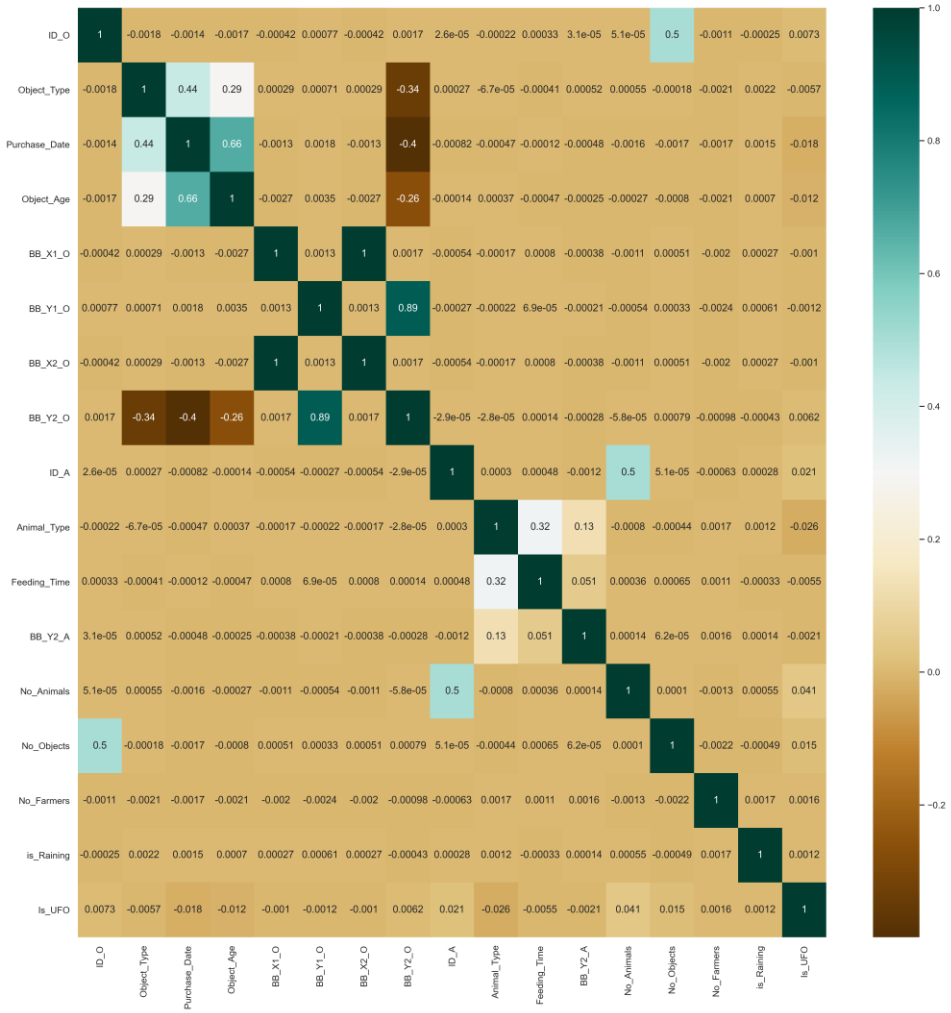

Figure 3: Correlation Matrix

Models:

Logistic Regression: Logistic regression is a linear model for classification.

- *Application*: Logistic Regression is used frequently for text classification.
- *Strengths*: Logistic regression can be updated after is has already been trained.
- *Weaknesses*: Logistic regression is not as fast as the naive bayes methods.
- *Selection Reasoning*: The ability to update itself after its initial training period may prove to be valuable.

Random Forest: A Random Forest is a ensemble algorithm that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. It counts a vote of each individual tree in order to produce the final class label.

- *Strengths*: The strength of the Random Forest classifier comes from the formation of its trees. Because it is formed from "random" subsets of the data, and the final result is compared to other trees that have also been formed "randomly", the algorithm guards well against "overfitting" from noisy data points that may have more influence on a single decision tree algorithm. The "random" formation of the trees ensures that there is little chance for a strong bias to be present in the data during tree construction.
- *Weaknesses*: A lot of trees are necessary to get stable estimates of variable importance and proximity. This can lead to a large amount of space in memory being needed to store the trees. Additionally, the trees need to be re-trained when new data is being introduced, unlike Naive Bayes.
- *Selection Reasoning*: Since random forest works well with high dimensional data, as is competitive with other algorithms such as SVM without the high training cost.

Adaboost: AdaBoost or "adaptive boosting" begins by fitting a "weak" classifier on the original dataset. It then fits additional copies of the classifier on the same dataset and adjusts the weights of incorrectly classified instances such that subsequent classifiers focus more on difficult cases. The adjustment is done using the SAMME-R algorithm (for classification)

- *Application*: Adaboost has been used in robust real time face detection.
- *Strengths of Model*: Because of its boosting property, Adaboost will not suffer from overfitting caused by too many training periods in the boosting algorithm. Theoretically, the algorithm should produce better and better results the more it is trained.
- *Weaknesses*: Adaboost is slow to train.
- *Selection Reasoning*: It will be interesting to measure the performance of the boosting method of Adaboost vs the vote method for the Random forest algorithm, as both use decision trees and differ only in the method in which the tree are ensembled together for the final classification.

Decision Tree: A decision tree is a flowchart-like structure. Each internal node represents a test on a feature (e.g. whether a pen is red or blue) , each leaf node represents a class label (decision taken after computing all features) and branches represent conjunctions of features that lead to those class labels. The paths from root to leaf represent classification rules. .

- *Strengths of Model*: Does not require data normalisation nor data scaling. Intuitive and easy to explain
- *Weaknesses*: Typically takes longer to train than most models. Relatively expensive to train. Not appropriate for applying regression and predicting continuous values.
- *Selection reasoning*: Its ability to using different feature subsets and decision rules at different stages of classification.

Evaluation Metrics:

ROC curve – receiver operating characteristic – plots true positive rate versus false positive rate at different thresholds.

AUC – area under curve. AUC measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1). AUC is the probability that a model ranks a random positive example more highly than a random negative example.

AUC has two main properties: scale-invariant and classification threshold invariant:

Scale-invariant: measures how well predictions are ranked, as opposed to absolute values.

Classification Threshold invariant – measure quality of the model predictions irrespective of classification threshold chosen.

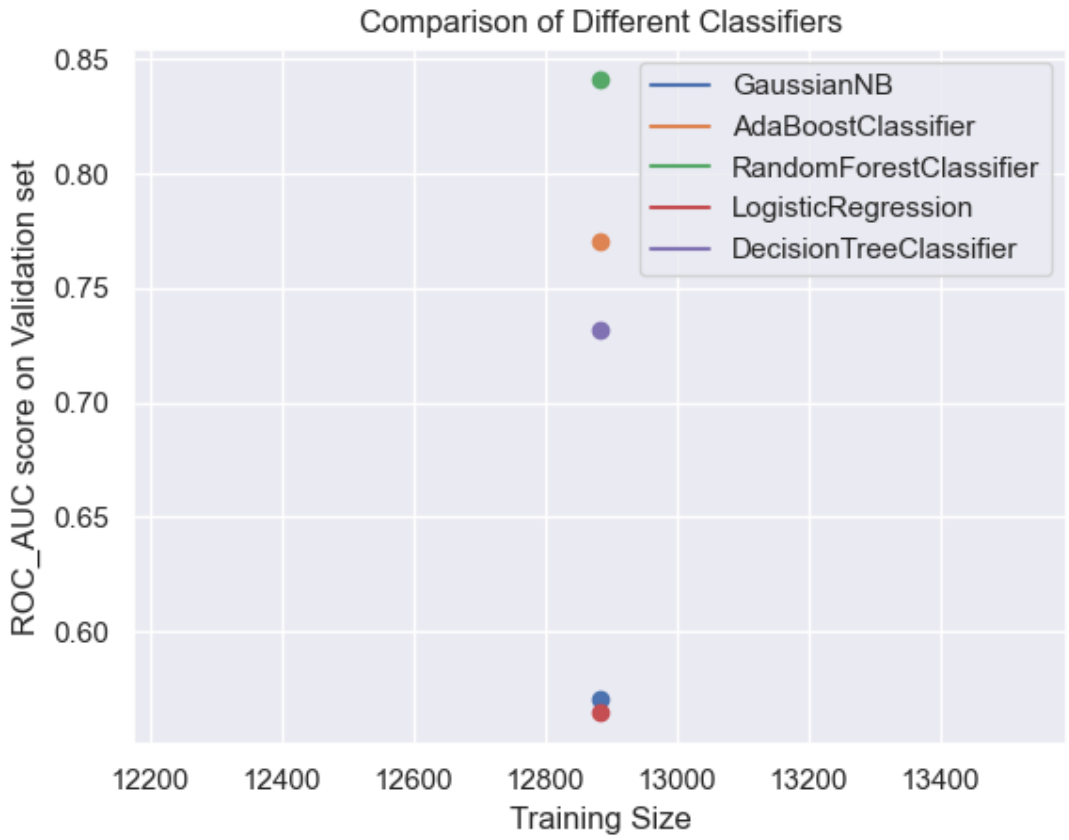

Figure 4: Classifier Performance

The best performing model was the Random Forest Classifier, such models are actually known to work well on imbalanced datasets – as an ensemble method, it used a variety of base classifiers. The independence is enforced by training each base classifier on a training set sampled with replacement from the original training set. This technique is known as bagging, or bootstrap aggregation. In Random Forest, further randomness is introduced by identifying the best split feature from a random subset of available features. The ensemble classifier then aggregates the individual predictions to combine into a final prediction, based on a majority voting on the individual predictions.

We also include the confusion matrix! The confusion matrix is an intuitive means to inspect True Positive, False Positive, True Negative, and False Negative rates. In such a task (i.e. with an imbalanced dataset) it is important to look beyond naïve accuracy measures and inspect further. E.g. We could obtain over 99% accuracy in this task just by always guessing false, but that would not really be of any worth. Finally we include the F1-score – the harmonic mean between precision and recall. Precision and recall are based on the confusion matrix categories, so the F1-score is a natural extension to the confusion matrix analysis.

	True Positive Rate	False Positive Rate	True Negative Rate	False Negative Rate	F1-Score
Adaboost	0.66	0.34	0.74	0.26	0.51
Decision Tree	0.7	0.3	0.75	0.25	0.45
Random Forest	0.66	0.34	0.85	0.15	0.5
Logistic Regression	0.67	0.33	0.47	0.53	0.5
Gaussian	0.67	0.33	0.47	0.53	0.5

We get some very interesting results! Despite the random forest outperforming the decision tree in the ROC_AUC curve, we actually note that the Decision Tree model actually sports better a better true positive rate, whilst the random forest has a superior true negative rate. Finally, the F1-score gives us the best overall picture, and we note that the Decision Tree actually performs poorest in this regard! The best is the Adaboost, which continuously trains on observations it previously predicted incorrectly.

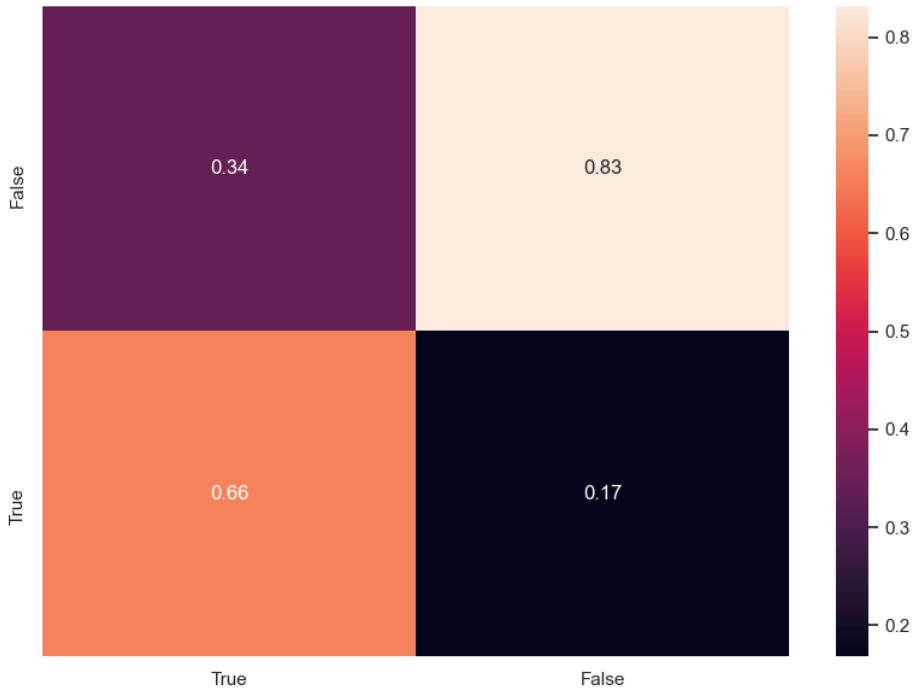

Figure 5: Random Forest Confusion Matrix

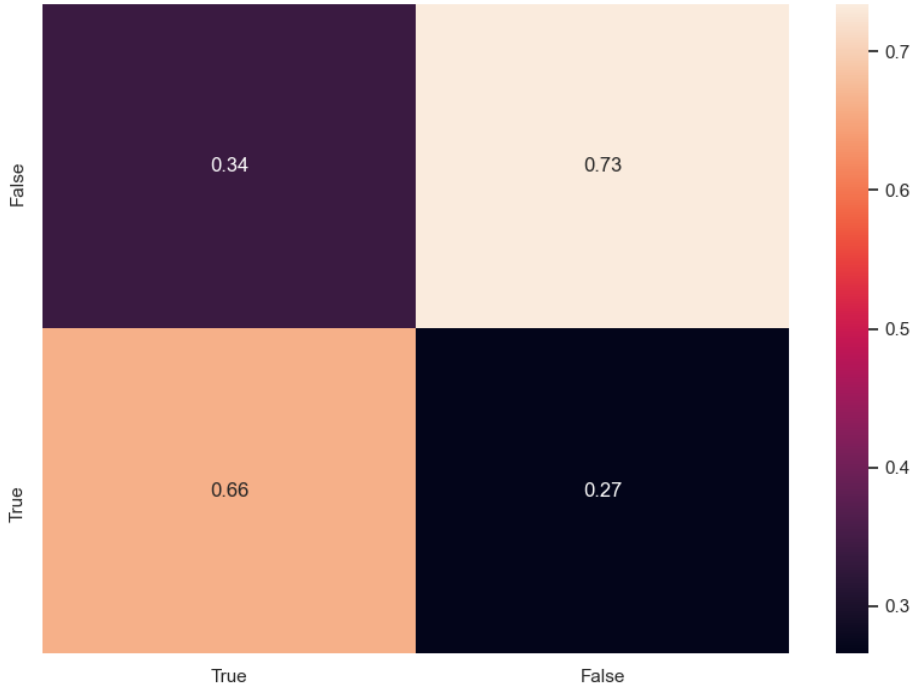

Figure 6: Adaboost Confusion Matrix

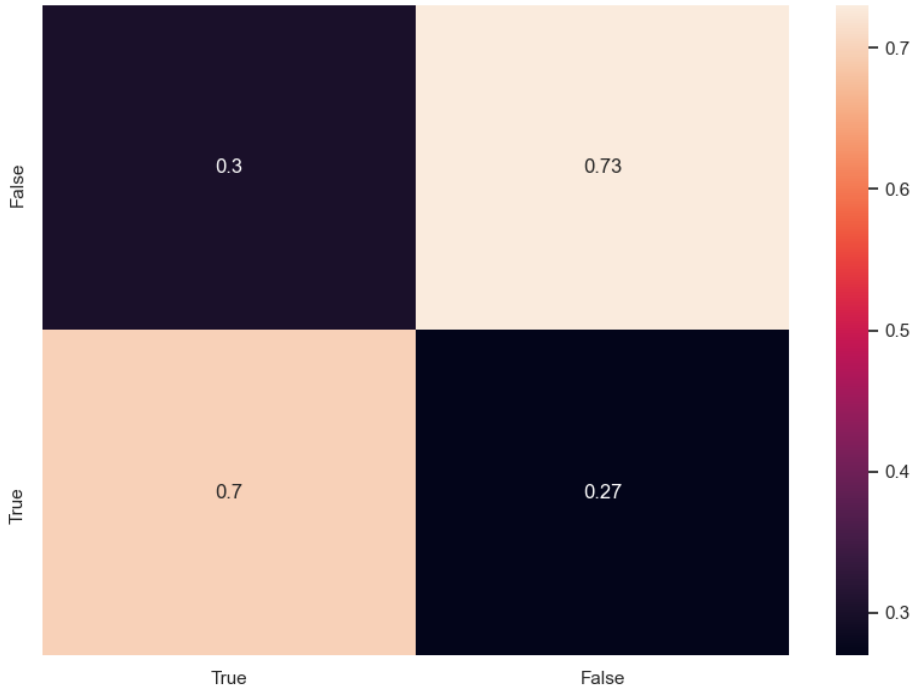

Figure 7: Decision Tree Confusion Matrix

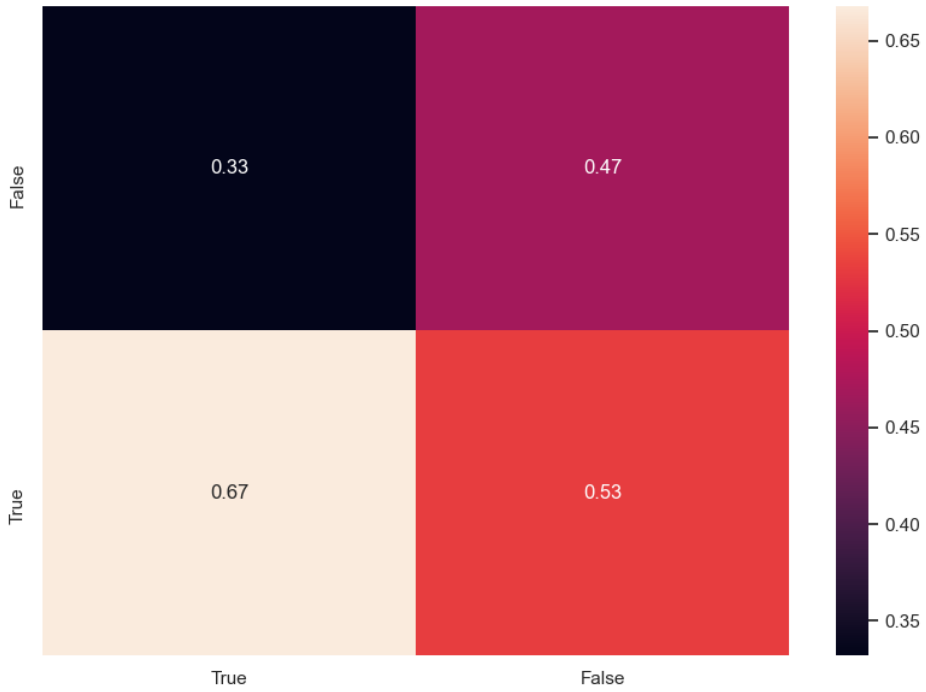

Figure 8: Gaussian Confusion Matrix