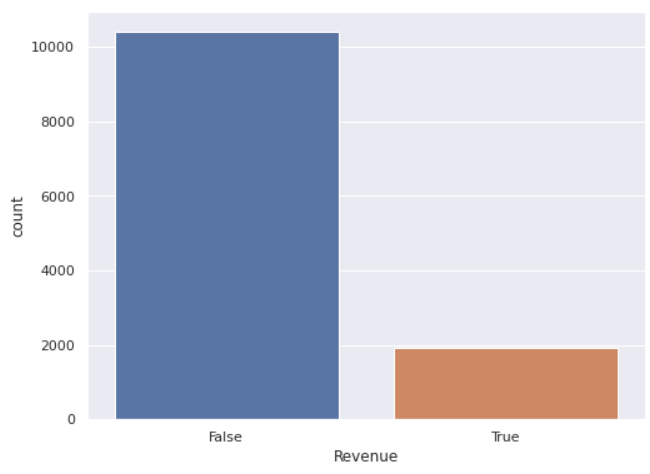
**Applied Machine Learning (CMT307)**

**Coursework 1**

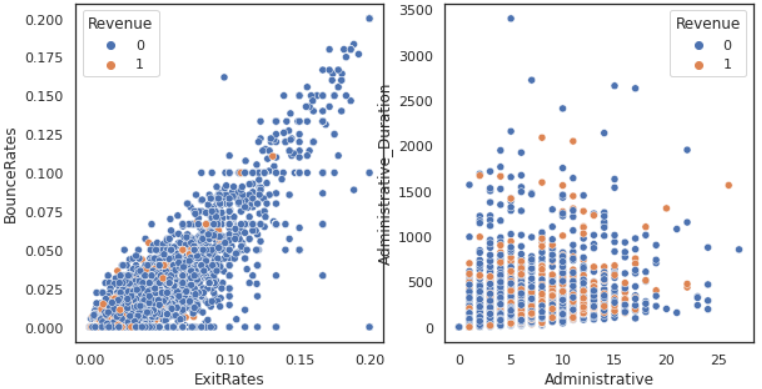
**Question 2**

**Data exploration**

The dataset consists of 12,330 rows of online shopping visits to a website, and eighteen features describe each row, divided into numeric and categorical features. Feature **“Revenue”** is the **label** of our binary classification problem. Our goal is to predict the label “Revenue”, to see whether a visit session will end up with a transaction or not. After plotting the**revenue we can see that**a small number of the visitors make a purchase, we suspect that the data is **imbalanced** as shown below.

****

In this section, we used multiple plotting types such as histogram, scatter, and box plot to be more familiar with the features. The distributions of the values in the individual features indicated which features correlate with each other. Furthermore, outliers could be clearly seen on the box plot. These are some correlated features with the target, as examples of what we have done on the attached python file.

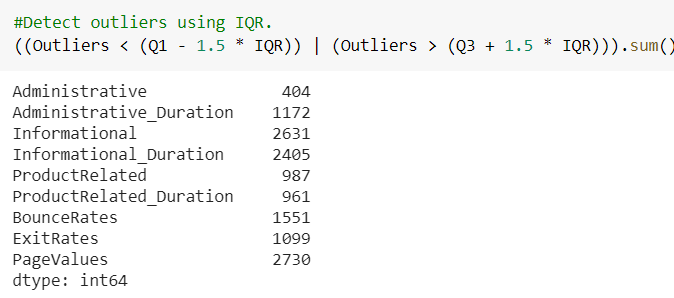


**Data pre-processing**

Since the dataset does not have any missing values, we started splitting the data into different train and test data sets. The training dataset (x\_train) will use 80% of the data. The test dataset (y\_test) will use the remaining 20% of the data. The datasets y\_train and y\_test contain the respective prediction labels. After that, we encoded categorical features using OneHotEncoder, OrdinalEncoder from sklearn for Visiter Type and Month respectively.

Whisker plots had shown a lot of outliers, so to detect them, we used IQR. Some features had more than 2000 outliers, so we decided to keep them may they have essential values, this provides a right candidate for using a robust scaler transform to standardize the data in the presence of skewed distributions and outliers.

The number of outliers was detected by using IQR.

****

As mentioned before we are dealing with an imbalance dataset, which means one class is dominating. There are very few instances of the other type. to treat that we chose to increase the minority class instances by using oversampling as it helps retain the data's variability. SMOTE algorithm (Synthetic Minority Oversampling Technique) was implemented.

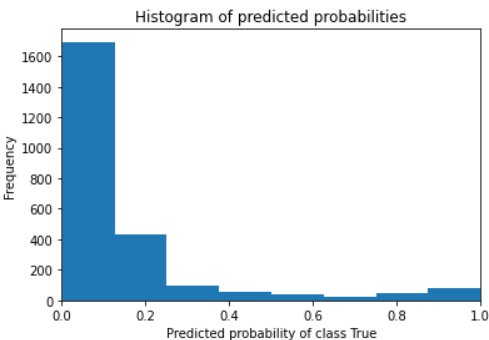
**Model implementation**

First, we started by test many classifiers used for a classification project include Logistic Regression, K-Nearest Neighbors (KNN), SVM, Kernel SVM, Decision Tree classification and RandomForest and most of them have similar performance so we will explain three of them, accordingly to what was asked in the question.

The XGBoost algorithm was suitable for a classification predictive modeling problem because it provides a set of hyperparameters that control the model training procedure. However, the algorithm works fine by default on our processed data set. We have tried it on imbalanced classification datasets because it provides a way to fine-tune the training algorithm to pay more attention to minority class misclassification of datasets with skewed class distributions. This modified version of XGBoost referred to as Class Weighted XGBoost could better perform binary classification problems with severe class disruption. Moreover, it is possible to achieve better performance by weighting a different class, but we were more interested in the positive class. graid search was used to determine the best parameters.

As we work with an imbalanced dataset, we implemented Logistic Regression because it might refer to probabilistic algorithms, as they often fit under a probabilistic framework. Good accuracy on train set and the test set was obtained by default, but the recall and precision were low for the positive label. This is achieved by using a threshold, such as 0.5, where all values equal to or greater than the threshold are mapped to one class, and all other values are mapped to another class. The cost of the True type of misclassification is more important than another type of misclassification.

As shown below, most of the true class probabilities are less than the default threshold(0.5), so we did decrease the threshold for predicting true to increase the classifier's sensitivity.

****

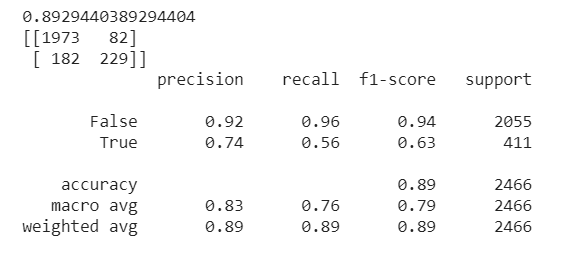
K-NN was implemented, but it did not perform as well as other algorithms with regard to recall or precision, although we searched for the optimal value for K using cross validation and grid search. So we prefer to use Kernel SVM because it can work well with a large number of features and is suitable for classification of extreme binary state with less effect of outliers.

**Performance evaluation**

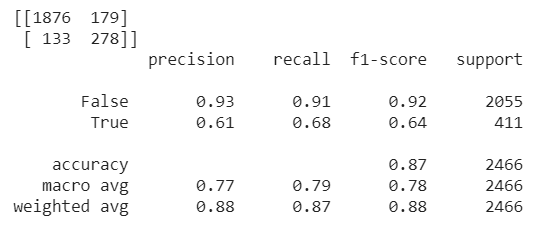
A comprehensive and clean way to illustrate the results of the classification model is the Confusion Matrix. It differentiates between expected and actual designations. Besides, to evaluate the models' performance, we used standard evaluation metric for classification problems such as recall, precision and f1 score for both labels, and the easiest way to track them with each model or when changing parameters is to print a classification report.

**These are the reports of the best performance models.**

This is for XGBClassifier, the first line represents the accuracy in the test set and the matrix is FP TN then FN TP.

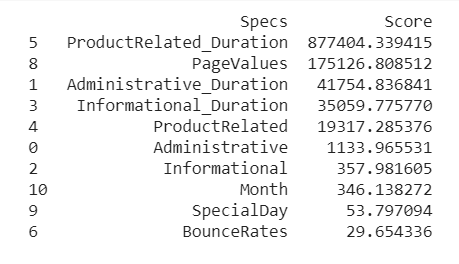
****

**Logistic Regression classification report.**

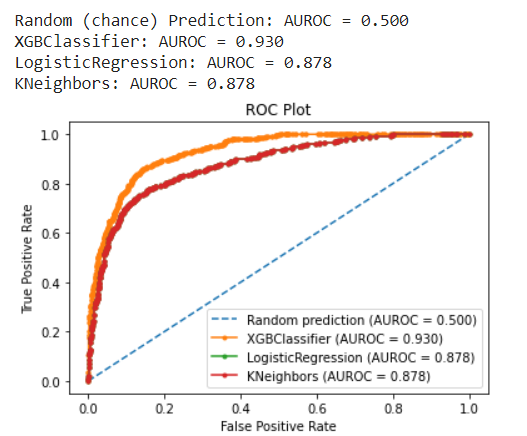
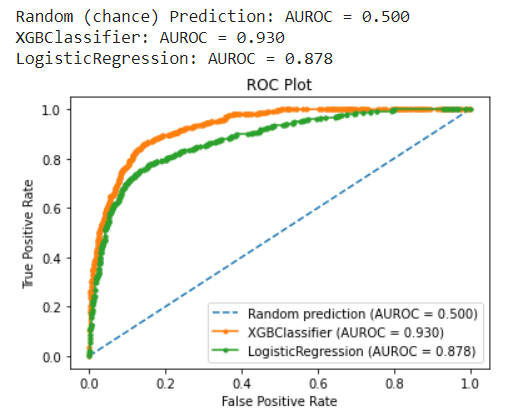
****

The K-NN and SVM kernel did not perform as expected at this stage of the evaluation but as we mentioned before cross-validation, gride search and random search were used during implementation. At this point, we defined features using the SelectKBest function and dropping the less critical features, but the models did not show a considerable difference from the first performance.

These were the most important features.

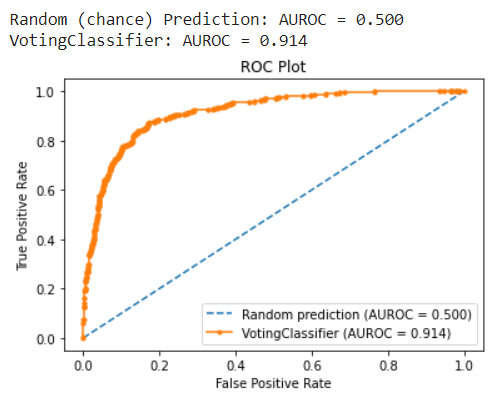


ROC\_curve for the three models, logistic regression and k-nn have the same performance.

****

**Result analysis and discussion**

We have developed a classification model for predicting the purchase intentions of online shoppers. Namely, we have trained and tested XGboost classifier, logistic regression, K-NN and SVM kernel algorithms, Therefore as described in ROC\_curve, it is one of the most common measures for evaluating machine learning algorithms' performance, especially when working with imbalanced data sets. All models function similarly, but the best one was the XGboost classifier. It clearly shows in the plots above, even when combined by VotingClassifier with all three models, XCboost is still the top performer.

ROC\_curve for VotingClassifier 

References

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition by Aurélien Géron.

Ian Goodfellow, Yoshua Bengio and Aaron Courville. Deep Learning. MIT Press,

2016.

Imbalanced Classification with Python, Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning by ason Brownlee.