Cardiff University School of Mathematics

Predicting e-commerce visitors' purchasing intention.

CMT307 Applied Machine Learning.

Mourouzidou Elisavet
Student ID Number: C2027059

Question 1:

For calculating the four metrics for performance evaluation, the key step is to create the Confusion Matrix.

		PREDICTED CLASS	
		Class=True	Class= False
ACTUAL CLASS	Class =Yes (positive)	TP = 7	FN = 2
	Class=No (negative)	FP = 4	TN = 7

TP = all predictions that are correctly predicted as True = 7

FN = all predictions that are incorrectly predicted as False = 2

FP = all predictions that are incorrectly predicted as True = 4

TN = all predictions that are correctly predicted as False = 7

For calculating the accuracy, precision, recall, and F-measure we used the above formulas:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{7+7}{20} = \frac{7}{10} = 0.7$$

$$\text{Precision}(p) = \frac{TP}{TP + FP} = \frac{7}{11} = 0.6364$$

$$\text{Recall}(r) = \frac{TP}{TP + FN} = \frac{7}{9} = 0.777$$

$$\text{F-measure}(F) = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2*\frac{7*7}{11*9}}{\frac{7}{11} + \frac{7}{9}} = \frac{2*7*7*9*11}{9*11*(7*9+7*11)} = \frac{2*7*7}{7*20} = \frac{7}{10} = 0.7$$

Question 2:

Data exploration

In the dataset, the last column 'Revenue' is the output(as can be seen in figure1 there is imbalance with ratio 5:1), while the other columns are the predictors(inputs).Ten of them are continuous positive variables, while the remaining eight are Categorical since their numerical values do not have any quantitative significance.Concerning numerical variables,we plotted "box-plots" and histograms for every variable separately.It is worth mentioning that, the dataset did not contain any missing values and we did not drop or replace outliers as we consider them crucial information.Regarding categorical features,we created bar charts to present their frequencies(figure2).Finally, we identified linear(or not) dependence between numerical attributes using a correlation matrix.

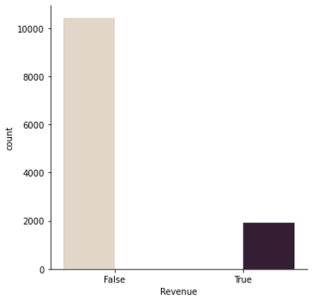

Figure 1: The imbalance in Revenue

Data pre-processing

Regarding pre-processing we observed that some "states" had very low frequencies concerning the others, so we introduced the binning method[i] to some categorical attributes. Because these frequencies may affect negatively the model's robustness or cause overfitting, we assigned a general category to these values, regarding the ratio of the True/False(Revenue).

About continuous features, we detected multicollinearity(a phenomenon in which a linear relationship can be detected in a multiple correlation model among unbiased explanatory features) with Variance Inflation Factor(VIF)[ii]. When VIF indicates values that are greater than 5 we dropped some of those features to reduce multicollinearity[iii].

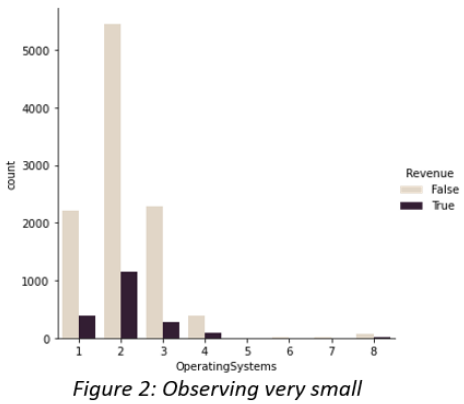

Figure 2: Observing very small frequencies at feature Operating Systems in classes 5,6,7,8

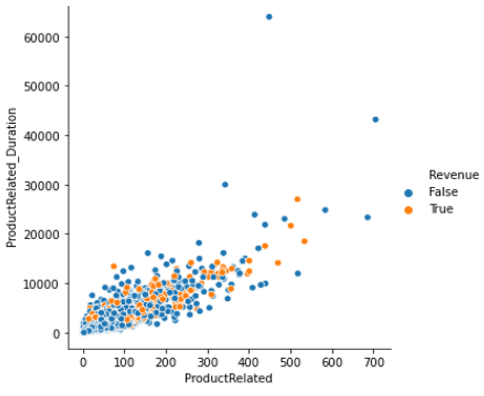

Figure 3: This scatter plot indicates linear relationship between two columns

We started with encoding categorical features and scaling numerical attributes(implemented in pipelines). Some categorical features seem that they were already label encoded without ordinal meaning.Hence, we used OneHotEncoder to create equal values of one and zero in each column.

Furthermore, the two major characteristics in our model were:1)The huge differences of variances among numerical features and 2)high positive skewness.We obtained the most suitable scaler, which will manage to stabilize variances and minimize skewness. For handling the outliers in the best possible way, we analyzed several different scalers. PowerTransformer[iv] handled the outliers well and improved the performance of specific algorithms such as logistic regression.

Model Implementation

1. *Logistic Regression(LR):*[v] We chose to start with a simple but also effective classifier.Logistic Regression is: highly interpreted, based on probabilities, can handle multiclass problems and is quick in terms of training. The first parameter we used for handling the imbalance data was **class_weight**.The **Solver** parameter is very useful as it can be adjusted depending on the type of dataset by using the appropriate algorithm. Logistic regression is a resistant technique in terms of overfitting since we can change appropriately the **parameter_C** for that purpose.Higher values give more freedom to our model.Finally, we used **penalty** parameter to consider penalization and control sparsity.

2. *Support Vector Machine(SVM):*[vi] We chose this classifier mainly because of the strength that takes from its parameters.Firstly, SVM has very strong **kernels** which can solve almost every complex(or not) problem since it can map data to a higher-dimensional space.It is very significant that SVM not only involves regularization by default but also by controlling **parameter_C>0**. So, the model can be robust even if the training sample has some bias.Moreover, in SVM the risk of overfitting is less.This classifier is quite sensitive regarding imbalanced data. However, SVM has its parameter **class-weight** to handle them.Lastly, by calculating the margin SVM reduces the risk of error.

3. *XGBoostClassifier:*[vii] XGBoost is a highly recommended ensemble learning method, since from weaker learners can create a strong learner, whereas provides a variety of hyperparameters. Moreover, it is more capable than other classifiers regarding the overfitting thread, as it uses a more regularized model.A parameter set to prevent overfitting is **learning_rate** and **Scale_pos_weight** set to deal with the issue of imbalanced data. The two parameters which are important to optimize together are **min_child_weight** and **max_depth** to find good balance between model bias and variance. The first one refers to the sum of instance weights needed in a child while the second is proportional to the complexity of our model. Furthermore, we used **n_estimators**[viii]since most of the gains and improvements are going to happen early, we avoid diminish them over time.The last two parameters are **Subsample** which corresponds to fraction of rows to subsample at each step and **Gamma** which requires the minimum loss reduction(when is large the algorithm becomes more conservative).

Performance Evaluation

[ix]*Performance metrics*:Our major problem is that our data are imbalanced and we calculated Precision, Recall, F-measure, Precision-Recall curve[x].While Accuracy and ROC are not suggested when we do not have roughly equal number of observations.

Evaluating Estimator Performance:We used the RepeatedStratified k-fold cross-validation(**CV**)[xi] technique, which involves making classes be equally presented in each k-fold. This method prevents overfitting and makes the model to be able to predict useful information on unseen data. We used Repeated stratified to keep the class distribution in train and test sets for each evaluation of a given classifier.

Hyper parameter Optimisation: Searching for the right combination of hyperparameters(into spaces) to achieve the highest possible results, we used RandomizedSearchCV with cross-validation since it is very efficient in terms of exploration. Another advantage is its ability of very easy parallelization. The scoring metric of optimization was F-measure.

Result Analysis

- *First model* contains only the scaling and the encoding part of pre-processing.

1st MODEL (Basic Model, PowerTransformer)				
	Accuracy	Precision	Recall	F1-Measure
Logistic Regression	0.8939	0.70	0.58	0.63
Support Vector Machine	0.8955	0.71	0.59	0.64
XGBoostClassifier	0.8968	0.71	0.60	0.65

Table1

- *Second model* contains all four pre-processing techniques.

2nd MODEL (Extra Pre-processing)				
	Accuracy	Precision	Recall	F1-Measure
Logistic Regression	0.8913	0.69	0.58	0.63
Support Vector Machine	0.8962	0.70	0.60	0.65
XGBoostClassifier	0.8962	0.70	0.61	0.65

Table2

After extra pre-processing there is no improvement in each metric, but F1-measure increases approximately 1% in SVM.We chose to balance both models as we mainly focus on F1-Measure.

- *Third model* balancing and comparing the two models to see how balancing contributes to performances.

3rd MODELS (Parameter class_weight for balancing not perfectly the 2nd and the 1st Models)				
	Accuracy	Precision	Recall	F1-Measure
Logistic Regression (LR)	0.878 | 0.880	0.59 | 0.60 -0.01	0. 75 | 0.74 +0.01	0.66|0.66 -
Support Vector Machine (SVM)	0.878 | 0.880	0.58 | 059 -0.01	0.80 |0.80 -	0.68 | 0.68 -
XGBoostClassifier	0.882 | 0.879	0.61 | 0.61 -	0.74 | 0.69 +0.05	0.67 | 0.64 +0.03

Table 3

Comparing the two models after using parameter class_weight, we observe that there are almost no changes in LR and SVM,but regarding XGBoost there are some considerable increases. As a result we choose the 2ndmodel(with the extra pre-processing) to optimize.

2ndModel:

In general, accuracy is very high in every model almost 90%.As we mentioned,it is not reliable metric(cause still some imbalances). After balancing the data in 2ndmodel, there is an increase in recall but a decrease in precision(see Table2-Table3 the red columns).Considering them both important, we chose to optimize the F1-Measure metric (combination of them).Based on F1-Measure balancing our data has a positive impact on model at 2%-3%.

- *Final Model*

FINAL MODEL(After Optimizing)						
	Accuracy	Precision	Recall	F1-Measure	AUC Score	F1 (With different Threshold)
Logistic Regression	0.878	0.61	0. 71	0.667	0.662	0.670
Support Vector Machine	0.878	0.58	0.80	0.680	0.665	0.680
XGBoostClassifier	0.882	0.61	0.74	0.67	0.723	0.672

Table4

We noticed that our model did not improve after tunning.The AUC-scores indicate that our model has a medium performance.In conclusion,XGBoost classifier manages to improve our model,since its results are more balanced and have better AUC-score.

References:

[i] https://www.kaggle.com/kickitlikeshika/advanced-eda-feature-engineering

[ii] https://link.springer.com/article/10.1007/s11135-006-9018-6

[iii] https://www.statisticshowto.com/variance-inflation-factor/

[iv] https://machinelearningmastery.com/power-transforms-with-scikit-learn/

[v] https://medium.com/axum-labs/logistic-regression-vs-support-vector-machines-svm-c335610a3d16

[vi] https://statinfer.com/204-6-8-svm-advantages-disadvantages-applications/

[vii] https://www.kdnuggets.com/2020/12/xgboost-what-when.html

[viii] https://xgboost.readthedocs.io/en/latest/parameter.html

[ix] https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc#3

[x] https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

[xi] https://scikit-learn.org/stable/modules/cross_validation.html