Fraud Detection Challenge

Lucas Ferreira da Rosa Moda

- **Reading data:** It is a 150k dataset with 30 features.
  - All features are continuous and there are *no null values*.
  - Features have different ranges and deviations.
  - It is an *unbalanced* problem, given that there are only 0.16% (243) records of the positive class.
- **Histograms:** Most of the features are nicely normally distributed, but some seem to have little variance (like PP7, PP8, PP20, PP21, PP23, PP27 and PP28).
  - Feature PP1 transformed (Cubic Root) in order to better separate its bimodal distribution.
- **Correlations**: Most of the features have insignificant correlations. Most notable are PP3/Ocorrencia and Sacado with other 4 varibles, the highest being -0.55 with PP2 (although this value does not imply very high correlation). Given that, I have decided to not exclude any variable based on correlation. PP17, PP14, PP12 and PP10 have the highest correlation with the target variable. Keep those features in mind.
- **WoE:** Variables were categorized by binning them into 3 equally-sized buckets (via Pandas' qcut). As an example, the first bin of feature PP2 has 10 times more frauds than the other 2 buckets. This kind of analysis is useful when you have knowledge about variables and assumptions about its relations with the target. Since data is anonymized, I did not go further in this direction.
- **Outliers:** Since most of the variables are normally distributed, a good way to detect outliers is to use the Z-Score and define an outlier as a record that has $|z| >= 3$.
  - The variable with the most outliers has 1.7% of records more than 3 standard deviations away from the mean. Many variables have less than 1% of such cases.
  - There are mixed results analyzing fraud frequency in each variable's outliers. For Sacado the distribution is basically the same as the whole dataframe, while PP3 and PP11 have much more outliers that are indeed fraudsters (up to 39% as opposed to 0.16%).
  - Three variables (PP24, PP1 and Ocorrencia) have not even a single Fraudster in their outliers, meaning it is just noise. In those cases, the outliers were *substituted by the mean*.
- **Features with Low Variance:** First, all features must be on the same scale - that's why Min Max Scaling was conducted (bringing everybody to the range [0,1]).
  - PP23, PP28 and PP27 have the least variance and (looking at the correlations heatmap) no correlation to the Target - those features were *dropped*. PP20 has correlation with Sacado, so I will leave it.
- **PCA:** First, features were scaled with Standardization (mean=0 and variance=1)
  - Looking at the Explained Variance, PCA was not able to retain much variance with few features (above 80% with 20 dimensions), so, it is not useful for this dataset and was not used.
- **Modeling:** A stratified 70/30 split was adopted, and the models used were Random Forest, Naive Bayes, Logistic Regression and SVM on the Standardized dataset (required for SVM and LR, not necessary for RF and NB). Since it is a Fraud detection problem, the focus was on **Recall**, since it is much worse to not detect a fraudster than to have a false alarm on non-fraudsters (*precision*).

- *Random Forest:* Did really well on the training set (0.94 Recall, 0.97 AUC), but severely overfits. Since it is an imbalanced problem, we can't look at accuracy, but rather KS and Recall. Those metrics are only about 0.68, although precision is good (rendering a respectable 0.78 F1-Score). If one wanted to maximize F1, this model is not too shabby, but we are looking for good Recall *regardless* of precision.
  - Feature Importance: 4 features are predominant (PP17, PP12, PP14 and PP11 - accounting for 60% of the splits.). Not coincidentally, those features are among the ones with the highest correlation to the target. The model makes sense.
  - Feature PP24 had the lowest importance, and it's the third variable with the highest variance, has 0 correlation with the target and 0 fraud frequency in its outliers. Dropping this column and retraining the model, however, just made one more fraudster being incorrectly classified as clean. As a result, it will be *not* be dropped.
- *Naive Bayes:* Didn't perform as well as RF on the training set, but it generalizes much better, and has a much higher Recall (0.85!). The "cost", however, is a huge decrease on Precision (0.07) and consequently on F-1 (0.13). Again, this depends on the objective - since we are focusing on Recall no matter what, NB did better than RF.
- *Logistic Regression:* Logistic Regression generalized better than RF, but had similar results (0.68 Recall, 0.84 AUC).
- *Linear SVM:* SVM also generalized better than the Random Forest and had basically the same F1-Score (0.77) - with a better Recall, AUC and KS, and worse Precision. If one wanted F-1 as metric, SVM would be the choice.

- **GridSearch:** Lastly, I tried a GridSearch using 5-fold Cross Validation and optimizing Recall - to see if Naive Bayes could be beaten. All three algorithms (and running SVM with both Linear and Gaussian kernels) did upgrade Recall by about 7 points, but still more than 10 points below NB (0.73 vs. 0.85).

Conclusion

We started with a dataset of 150k records and 30 features, containing an unbalanced (0.16%) amount of fraudsters and no null values. Most of the features were normally distributed and had very little correlation. Outliers ($|Z-score| > 3$) with no fraudster frequency were imputed with the mean, variables with low variance (PCA was also tried but deemed not useful, as too little variance was kept in low dimension spaces) were removed and then scaled.

For modeling, a stratified 70/30 split was adopted and Recall was the most important metric observed. Out of the models tested (Random Forest, Naive Bayes, Logistic Regression and SVM), NB had the best performance, achieving 0.85 Recall while not overfitting (although its precision and F1-score were much lower than the other algorithms - but we are focusing on Recall solely). While RF was almost perfect on the training set, it severely overfitted and had a Recall of 0.68 (same as LR, that generalized better). Linear SVM had about the same F-1 Score (0.77) as LR and RF, but it has a higher Recall (0.72) and Lower Precision (0.84) - hence, would be prefered if F-1 was the metric chosen. GridSearch did improve some points in Recall, but still about 10 points below NB.

If one wanted to go even further on this analysis, techniques such as Undersampling, Oversampling and Synthetic Sampling (like SMOTE) could be used to avert the imbalance between classes. Even more, the problem could be turned into an Anomaly Detection, using techniques such as Isolation Forest or Local Outlier Factor.