**Automobile Insurance Fraud dataset: Some considerations regarding supervised machine learning models and their application**

**Fernando Rosales-andresrosales7@gmail.com**

**Problem Definition**

Fraud is a very common situation in the insurance industry, fraud analysis is getting traction lately, and several techniques such as linear regression, decision trees, and Gradient Boosting had proved effective. No doubt fraud detection has negative consequences and insurance companies have implemented a series of countermeasures to detect and prevent fraud.

**Exploratory Data Analysis Concluding remarks**

Exploratory Data Analysis (from now on EDA) is one of the first tasks at hand when analyzing a dataset, it consists in understanding the data and identifying potential patterns that might be helpful to achieve the goal. Before getting into model selection it’s a good idea to understand the nature of the variables and eliminate those which don’t contribute to the model.

It’s a thorough examination meant to uncover the underlying structure of a data set and is important for a company because it exposes trends, patterns, and relationships that are not readily apparent.

Reliable conclusions cannot be drawn from a massive quantity of data by just gleaning over it.

Getting a “feel” for this critical information can help to detect mistakes, debunk assumptions, and understand the relationships between different key variables. Such insights may eventually lead to the selection of an appropriate predictive model.

The data set has 1000 records, where each row corresponds to an accident, the sample size has been considered to be sufficient by the author of the dataset to get an idea of the behavior of this phenomenon. Clearly, some kind of data cleaning has been done by the author, since there are no missing values.

At first sight, 3 classes stand out, variables related to the characteristics of the policy, the insured, the incident, and the vehicle. Among categorical and numerical variables, 40 are present in the dataset.

**Data cleaning and arbitrary decision**

This case, and especially this being a Time Series problem, added some extra layers of complexity when analyzing the features. In order to handle this situation, it was considered necessary to leave out as many Time Series features as possible, and stick with strictly numerical values. The variable *month\_as\_customer*, seemed ideal, since it was a numerical, and left *incident\_date*, as the index of the dataset. All these decisions were taken arbitrarily, which seemed like the most straightforward path to handle the dataset.

**Categorical Encoding**

It’s a set of techniques to encode categorical values, which cannot be processed by the majority of machine learning algorithms. Any model needs to be adjusted and evaluated, and all the input and output variables need to be encoded. There are 2 kinds of categorical values, nominal and ordinal. Nominal values categorize and tag attributes, like color or pet. Ordinal values have an intrinsic value, a hierarchy, such as education level or agreeableness.

In this solution, it was necessary to manually input the suffixes ‘ord’ and ‘nom’, for ordinal and nominal values respectively, and then a function that analyzes the categories. For ordinal variables, another function was implemented and the values were manually imputed as part of it. Finally, a categorical mapping function alongside the label encoding technique was used to map each of the nominal values.

For more information on this, check this Kaggle notebook: MOBIUS. January 2022. An overview of Categorical Encoding Methods. Retrieved  July 2022 from <https://www.kaggle.com/code/arashnic/an-overview-of-categorical-encoding-methods/notebook>

**Data Pre-processing with MinMax Scaler**

Data scaling is a preprocessing step for numerical data. Many machine learning models like Gradient Boosting, KNN, linear regression or logistic regression need scaled data to produce good results. MinMax Scaler reduces the data within a given range, usually between 0 and 1. It transforms the data scaling the features in a given range without changing the shape of the original distribution.

**Feature Selection with Select From Model and Random Forest Classifier**

Not all the variables are needed to get a good model. Usually, using a certain number of variables inside a subset of the data set generates better outcomes. A good feature selection will allow the reduction of the complexity of the models to use, will allow to better understand them and increase the precision of the predictions made by them. Nevertheless, no feature selection algorithm guarantees an optimal model.

Overall Random Forests are one of the most popular machine learning models. In general, they provide a general good predictive performance, low overfitting, and easy interpretability. Feature selection using Random Forests come under the category of embedded methods. Every tree of the forest is built under a random extraction of the observation from the dataset and a random extraction of the features. Not every tree sees all the features or all the observations, and this guarantees that the trees are de-correlated and less prone to over-fitting.

After applying this feature selection method, the features that were revealed were very intuitive, considering the nature of the problem at hand, among them: *months\_as\_customer*, *age*, *policy\_anual\_premium*, *insured\_zip*.

**Splitting data into test and train subsets**

Building the models, training them with known data, and made them able to apply them with unseen data. That’s where the *train\_test\_split* model validation procedure comes into place, in this case 25% of the dataset was used for testing. In this section, it was made sure that the data builds a model with enough data to be able to get insights that allow the model not only to interpret the training data, but to be able to predict.

The predictive performance its made up comparing the predicted facts mad by the model and in this case, the actual fraud cases present on the test sample

**Machine Learning Models**

Supervised Machine Learning classification models are definitely the way to go for the industry to manage this situation, they are reliable and accountable tools, since they can handle related variables. Supervised classification its one of the most common tasks carried out in business environments, and there are some tools that have proven their effectiveness. Among these tools, some stand out: Logistic Regression, Decision Trees, Random Forest, Gradient Boosting and Extreme Gradient Boosting.

Logistic Regression

This model is a statistical method for binary classification that is used to predict the relationship between one dependent binary variable and one or more independent variables. Logistic regression requires considerably big data samples. This model is very popular because is very efficient and does not used too many computational resources.

Decision Trees

Decision Trees is one of the most used methods in Machine Learning, it generates a graphic solution that considers all the possible outcomes to the problem, the first node its the *root* and then the algorithm arises. This algorithm its based on systematized conditions in an hierarchical structure that creates the branches until it reaches the final solution to the problem. The decision tree builds itself through decision rules, that’s why the final outcome is very visual.

Random Forest

Random Forest is the next step on the evolutionary path of the Decision Trees model, that has 2 major flaws: Bias and Variance error. Decision Trees with few branches or nodes usually have a high Bias error and reduced Variance error. Random Forest uses bagging where many trees are created and each of them with many branches, which translates in low Bias but high variance.

Extreme Gradient Boosting (XGBoost from now on)

This is one of the most relevant techniques that use ensemble boosting, and one of the key features its getting a “bias-variance” equilibrium. This model first generates a simple decision tree with very few branches, which translates in low  variance error.Trees are added one at a time to the ensemble and fit to correct the prediction errors made by prior models.

**Performance Metrics**

No doubt data cleaning and model selection and training it’s a key step in this process, but it is equally important to measure the perform. Having different the information of different metrics would allow to hone in the predictive power of the model before production, the moment when it has to deal with unseen data.

A proper evaluation has to be done in order to help the model to actually get insights from the data, to learn from it, and not memorize it. Performance metrics are a part of every machine learning pipeline. They tell you if you’re making progress, and put a number on it.

Every machine learning task can be broken down to either Regression or Classification, just like the performance metrics. There are dozens of metrics for both problems.

Classification models have discrete output, so metric that compares discrete classes in some forms are needed. Classification Metrics evaluate a model’s performance and tell how good or bad the classification is, but each of them evaluates it in a different way.

Performance measures in machine learning classification models are used to assess how well machine learning classification models perform in a given context. These performance metrics include among others accuracy, precision, recall, and F1-score.

Accuracy Score

It’s the total percentage of correctly classified elements, it’s the more direct and straightforward metric for classifiers. It has a value between 0 and 1, the higher the better.

ROC Curve and AUC

Once the models are applied its necessary to identify which of them have a better performance, and when a classification problem its at hand, ROC-AUC curve performance metric it’s one of the best ones out there. ROC stands for Receiver Operating Characteristic, and simply put, it tells how good the model its to tell apart from 2 or more things. This metric measures the capacity of the model to classify properly the positives in different thresholds. ‘

AUC is the area under the ROC curve, this score gives a good idea on how good the model performs, in an ideal scenario when 2 curves (in this case, *fraud\_reported* Y or N) do not overlap at all, the model has an ideal separation and its perfectly capable to discriminate between positive and negative.

Cohen Kappa Score

After a machine learning model is trained and tested, there are two primary factors that need attention. These are reliability and validity. Reliability is the level of trust we have on the model to produce consistent results in similar situations. It is the precision of the model. On the other hand, validity is the accuracy of the model on test data, that is how good is the result produced.

Cohen’s Kappa is a statistical measure that is used to measure the reliability of two raters who are rating the same quantity and identifies how frequently the raters are in agreement.

Time Taken

It is of utmost importance to make an accurate estimation of the time and cost required to train a machine learning model.

For instance, if the training period will be longer than anticipated, adjusting the parameters of an algorithm or choosing an alternative algorithm to implement in order to shorten the amount of time required to train the model might be necessary. This is very important when various machine learning experiments need to be run over a short period of time.

Confusion Matrix

It’s a matrix representation of the predictions of any binary classification model that is often used to describe its performance, and it is relatively easy to understand.

This tool helps visualize de performance the performance of a supervised model, it is called a confusion matrix because it's easy to identify where the model is mixing up 2 classes.

**Concluding Remarks**

After having a quick look of the features, and the selection of the most relevant ones, the next step forward was the application of the different models to detect fraud, and a model comparison scheme was plotted in order to compare the performance of each of them. The objective of this business case its being able to predict what might happen in the feature with certain degree of certainty.

Overall XGBoost showed better performance, and better accuracy and also took a considerable amount of time. Machine Learning models for fraud detection cannot substitute an experienced claims adjuster. Still, they will become handy to process vast quantities of information, get a better understanding of this situation and in ideal cases, would improve business strategies, minimize the negative impact of these behaviors and faster decision making.