Auto Insurance Claim Prediction for Porto Seguro

TABLE OF CONTENTS

1. Discover

1.1 Understanding the problem

1.2 Why ML over other solutions

1.3 KPI and metric(s) to track

1.4 Data Collection

1. Play

2.1 Create Package

2.2 EDA

2.2 POC

2.3 Decide on Model

1. Develop
2. Deploy
3. **Discover**

**1.1 Understanding the problem**

Porto Seguro wants to improve customer satisfaction by tailoring auto insurance prices for drivers. We need to know which customers are likely to file a claim(bad drivers) and which are likely to not(good drivers). The final goal is to increase insurance prices for bad drivers and reduce the prices for good drivers.

This project aims to use machine learning to predict which customers are likely to file a claim.

**1.2 Why ML over other solutions**

* Adaptability
* Scalability
* Grasp Complex Patterns

**1.3 KPI and metric(s) to track**

1. KPI**:** Increase customer satisfaction by 60% over the next year.
2. Metric:

* F1: balance between reducing False Positives and False negatives as both are costly) and robust to class imbalance
* AUC: Robust to Class Imbalance

**1.4 Data Collection**

The data was collected from [Kaggle](https://www.kaggle.com/competitions/porto-seguro-safe-driver-prediction/data)

**1.4.1 About Dataset**

In the train and test data, features that belong to similar groupings are tagged as such in the feature names (e.g., ind, reg, car, calc). In addition, feature names include the postfix bin to indicate binary features and cat to indicate categorical features. Features without these designations are either continuous or ordinal. Values of -1 indicate that the feature was missing from the observation. The target columns signifies whether or not a claim was filed for that policyholder.

**1.4.1.1 File descriptions**

* Train.csv: contains the training data, where each row corresponds to a policy holder, and the target columns signifies that a claim was filed.
* Test.csv: contains the test data.

**2. Play**

2.1 Create Package

2.2 EDA

2.2 POC

2.3 Decide on Model