1. The project's domain background

   This project is based on Porto Seguro's Safe Driver Prediction Competition on Kaggle[1].
   Porto Seguro is one of the biggest auto insurance company in Brazil.
   Porto Seguro has been working with machine learning over the last 20 years, but now they rely on Kaggle community to build a new fair algorithm.
   A common problem for an insurance company is to predict if a person will claim insurance in the coming years.
   Inaccurate evaluations for insurance cost a lot of money, so being able to predict if a new insurance is a good investment or not, it's important.
2. A problem statements

   This is a regression problem, which objective is to analyze characteristics from drivers, which includes personal information like age, sex, employment status, education background and many others. Besides personal information, there are other information related about the car (model for example) and neighborhood that are taken into consideration (HDI, slums existence, police station number).
   With this information, it's possible to check if a driver is a good driver or a bad driver, usually a good driver has a cheaper insurance and bad driver usually has a more expensive insurance.
   It can be defined that a good driver as the one that doesn't claim insurance at all and a bad driver is the one that claim insurance very often (once or two in a year).
   Inside the train file there is the target input (Second Column) which values can be 0 or 1, 1 means that that driver has claimed insurance in the last year and 0 means that no insurance has been claimed.
   The others 57 columns inside the train file (58 total) are the information mentioned above. Since it's a regression problem with some data available, it's possible to develop a machine learning model.
   Like the train file, the test file has 57 columns, all information related to driver without the target input.
   For each ID in the test set (First Column), it should be predicted a probability of an insurance claim.
3. The datasets and inputs

   There are 2 files for this competition:
   - Test File[2]
   - Train File[3]

   Test file has 595.213 records with 58 inputs.
   Train file has 892.817 records with 57 inputs.
   Most of inputs are anonymized since information are related to real auto insurance.
   Let's look inside train file:
   - First column: customer ID,

[1] https://www.kaggle.com/c/porto-seguro-safe-driver-prediction#description
[2] https://www.kaggle.com/c/porto-seguro-safe-driver-prediction/download/test.7z
[3] https://www.kaggle.com/c/porto-seguro-safe-driver-prediction/download/train.7z

- Second column: target (1 – has claimed insurance, 0 – hasn't claimed insurance),
- Third column to twentieth column (18 inputs): personal information,
- Twenty-first column to twenty-third column (3 inputs): region information,
- Twenty-fourth column to thirtieth-octave column (15 inputs): auto information,
- Thirtieth-ninth column to fiftieth-octave column (20 inputs): calculated feature.

And the test file:

- First column: customer ID,
- Second column to nineteenth column (18 inputs): personal information,
- Twenty column to twenty second column (3 inputs): region information,
- Twenty third column to thirtieth seventh column (15 inputs): auto information,
- Thirtieth octave column to fiftieth seventh column (20 inputs): calculated feature.

Porto Seguro hasn't said what criteria was used to divide the train file and the test file.

- Analyzing the target feature in train file[4]

The target features are unbalanced.
There are 573.518 records which target is 0 and 21.694 records which target is 1.
The ratio for target 1 is 1:26,4 (approximately).

4. A solution statements

For this project, it will be evaluated 5 tests:

- the first one the model will only be trained considering the personal information from the driver
- the second will be based on region information only
- the third test is the auto information only
- fourth just the calculated feature
- the last one will be considered only the features that are more relevant to the test. (more information below)

The main point to test so many times, it's to check if only personal information or region information or auto information or calculated features is enough to make a reasonable prediction model. (Most Likely not)
The last one is the most important test, since it takes just the features that have the higher contribution for a generalization.
To check which features are more relevant, it will be made prior analysis.
The Kaggle platform have a discussion forum, that many contributors make this prior analysis, so for proposal submit it will be considered their analysis. However, for the capstone project it will be made a personal analysis checking if the information is reported correctly.[5]

Comentado [RH1]: Some ideas to address
•What "preprocessing" will be done? How will you process your features to be inputted in your models?
•What features will you use? All of them? Will you reduce the size of the feature set by choosing the most predictive features? If so what techniques will be explored? Maybe check out the feature_selection module in sklearn?

[4] https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/
[5] https://www.kaggle.com/arthurtok/interactive-porto-insights-a-plot-ly-tutorial

Since the data is unbalanced, it can be used some filters on the last test (relevant features) like Variance Threshold, to remove features with low variance, this way only the features that contribute the most to target feature will be trained.
It'll be listed below techniques that will be applied in this project:

a. Naïve Bayes (GaussianNB)[6]
b. Stochastic Gradient Descent[7]
c. Random Forest[8]
d. SVM[9]

Besides the variance threshold, it can be applied some feature selection from sklearn:

e. Select K Best features[10]
f. Select Percentile[11]

At first, it's not planned to apply any optimizing technique in the dataset.

5. A benchmark models
   Since it will be evaluated 3 techniques with 5 datasets for training, it will be made 15 models.
   I believe the Naïve Bayes it will be a good benchmark model. (1 technique with 5 datasets = 5 models)
   However, the real evaluation it will be performed at Kaggle website, since the main point is to participate my first competition.
   It's known that the scoring metric is based on Gini coefficient.
6. A set of evaluation metrics
   For the EDA (Explorer Data Analysis), it will be performed the following evaluation metrics:
   a. Confusion Matrix – Evaluate FP (False Positive) and TP (True Positive)
   b. ROC[12] curve – (Check the Area under ROC curve)
   c. F1 measure
   d. Correlation Plot

   For the regression models, the R^2 score is a good evaluation metric.
7. An outline of the project design
   a. Create IPython file
   b. Import Data and library from sklearn
   c. Format the data so it fits test and train models (format data for each feature (car, calculated features, personal information and region information)
   d. Apply Techniques (Naïve Bayes, SGD, Random Forest)
   e. Evaluate the results (Gini Coefficient, F1 and Area under ROC curve)

Comentado [RH2]: •Since, I believe that your dataset is imbalanced, maybe think about using a naive benchmark model and only predict the most favorable class.
•Maybe use a simple machine learning model to get a very simple baseline to get an idea of how a simple model could perform.

Comentado [RH3]: https://www.kaggle.com/batzner/gini-coefficient-an-intuitive-explanation

[6] from sklearn.naive_bayes import GaussianNB
[7] from sklearn import linear_model
[8] from sklearn.ensemble import RandomForestRegressor
[9] http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR
[10] http://scikit learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#sklearn.feature_selection.Selec KBest
[11] http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectPercentile.html#sklearn.feature_selection.SelectPercentile
[12] Receiver Operating Characteristic

f. Check the most value features and select them
g. Evaluate the results (Gini Coefficient, F1 and Area under ROC curve)
h. Send data to Kaggle Platform and evaluate
i. Generate report.html
j. Upload report.html

Observations:
Is there is some techniques that might be a good evaluation for this dataset, I kingly ask to tell me. Since it's my first competition on Kaggle I'm kind of lost.
I'm not sure if Decision Tree is a for this dataset, because usually decision tree overfit (maybe pruning might help, not sure).

Maybe Neural Network for regression might be good (SGD).

There is a workflow to decide which preprocessing should be used since we know the data that will be evaluated?

How can I decide that's the best technique to use it?