Machine Learning Engineer Nanodegree

CAPSTONE PROPOSAL

Kennedy Sousa
December 11th, 2017

Proposal

Domain Background

Bank ACME's[1] Office of Compliance is a department responsible for monitoring the activities and conduct of employees: whenever an irregularity is detected, the bank must analyze whether the irregularity stems from misconduct or weaknesses in the process, in order to mitigate the operational risk and apply the penalty to those involved, if applicable, including possible compensation for financial losses.

The procedure starts with a process called *preliminary analysis* that consists in an investigation and aims to gather information about the issue, like authorship, which rule was broken, description of the facts, value involved, etc. After all the relevant information is gathered, the final report and the chain of evidence are sent to decision-making authority for deliberation. If the case is admitted, the indictee becomes defendant and is subject to penalties like written reprimand, suspension and discharge.

This project addresses the real world problem of identifying whether the case will be admitted or not, based in some multiple-choice fields filled in the report.

Problem Statement

The goal is to create a basic API that receives an input (the observations) and returns the predicted class (process admitted or not); the tasks involved are the following:

1. Download and preprocess the labeled data
2. Create a naïve classifier
3. Train an estimator
4. Test the estimator against the naïve classifier
5. Improve the estimator
6. Create a web service to deliver the functionalities

The final application is expected to give a clue about decision trend.

Datasets and Inputs

The dataset used in this project is a response to a data query sent to SQL Server and stored into a Comma Separated Values (.csv) file. Due to sensitivity of data all fields that could identify any

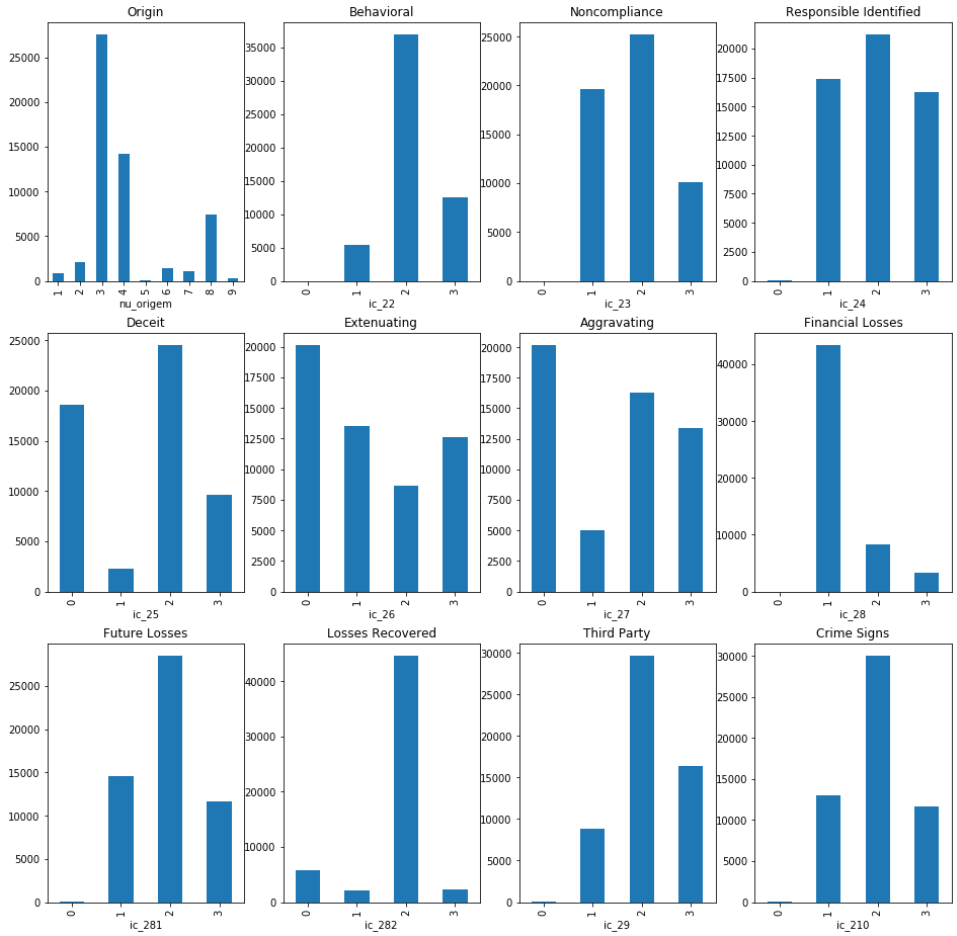
[1] Some names and identifying details have been changed.

person or entity were intentionally removed. The dataset has a 61,016x39 shape and the following fields:

Feature	Description
nu_analise	Sequential number, by department and year, from 1 to 9999.
nu_unidade	Department responsible for analysis.
dt_inicio_analise	Begin date.
nu_origem	Origin of register: 1. Audit 2. Denunciation 3. Manager 4. Credit Operation 5. Suspected money laundering 6. Guarantee Fund for Time Service 7. Robbery 8. Customer account refund 9. Office of Compliance
nu_unidade_ocorrencia	Department where the event occurred.
cd_situacao*	Analysis status: 1. Awaits filling 2. Awaits decision 3. Granted 4. Declined
ic_22	Are the supposedly irregular facts of behavioral order? 0. Not filled 1. Yes 2. No 3. Not identified
ic_23	Is there evidence of noncompliance with rules, laws or standards per employee, former employee or retiree? 0. Not filled 1. Yes 2. No 3. Not identified
ic_231	Has noncompliance determined the occurrence? 0. Not filled 1. Yes 2. No 3. Not identified
ic_24	Was the employee, former employee or retiree responsible for regulatory noncompliance identified? 1. Yes 2. No 3. Not identified

Feature	Description
ic_25	Are there evidences of fraud or deceit in the conduct of the indictee? 0. Not filled 1. Yes 2. No 3. Not identified
ic_26	Are there extenuating circumstances in the fact investigated? 0. Not filled 1. Yes 2. No 3. Not identified
ic_27	Are there aggravating circumstances in the fact investigated? 0. Not filled 1. Yes 2. No 3. Not identified
ic_28	Are there financial losses? 0. Not filled 1. Yes 2. No 3. Not identified
ic_281	Is there any possibility of future financial losses? 0. Not filled 1. Yes 2. No 3. Not identified
ic_282	Were the financial losses recovered or reimbursed? 0. Not filled 1. Yes 2. No 3. Not identified
ic_283_total	Amount lost (money).
nu_283_res	Amount recovered or reimbursed (money).
ic_284	Have the amounts been accounted? 0. Not filled 1. Yes 2. No
ic_29	Is there a third party involved? 0. Not filled 1. Yes 2. No 3. Not identified
ic_210	Are there signs of crime? 0. Not filled 1. Yes 2. No 3. Not identified

*** The field *cd_situacao* is the label or target values.**

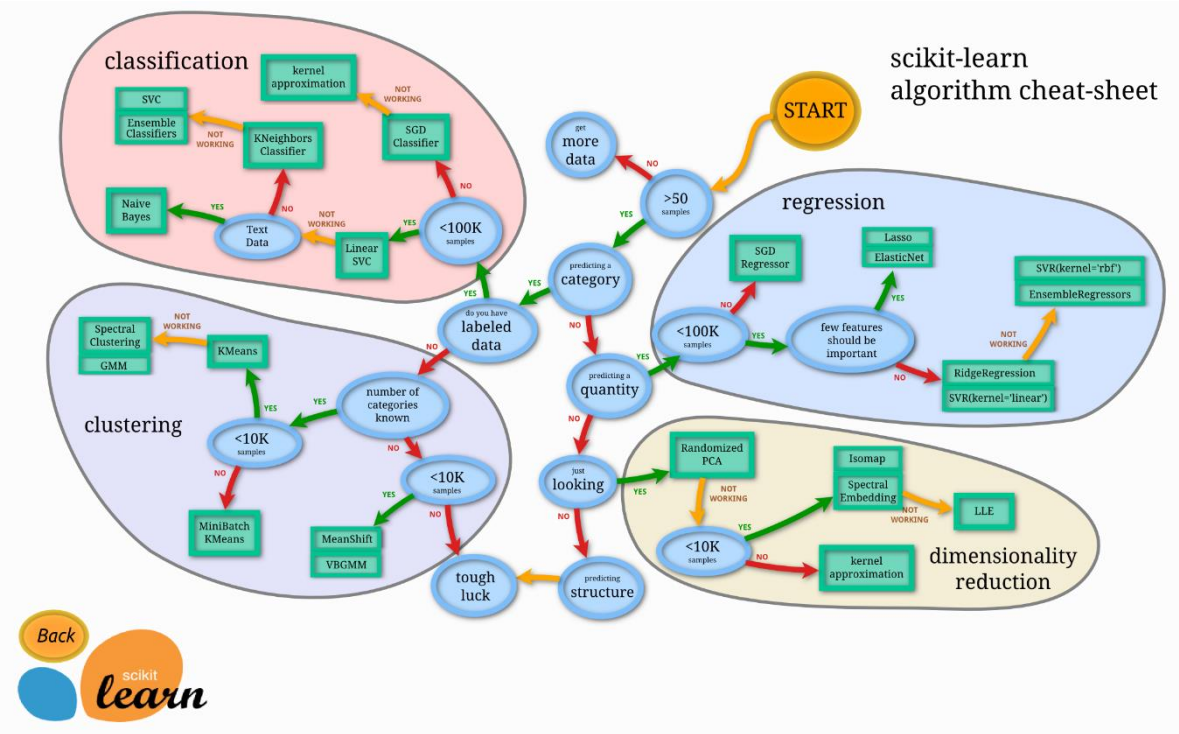


A quick visual analysis is very helpful to identify important insights, like the large number of features with answers marked as "not identified". On the other hand, the indication of deceit and aggravating circumstances can be very useful when predicting the target. Although the signs of crime appear to be a determinant feature, there is always a chance of the employee be just a victim of criminals.

Solution Statement

Classification is an example of pattern recognition and consists in one of the most commons problems in *Statistics* and *Machine Learning*. The goal is identifying to which of a set of categories a new observation belongs, based on a training set of data containing observations (or instances) whose category membership is known.

The following flowchart[i] is useful to describe the approach used to solve the problem: the solution handles more than 60k samples and tries to predict a category based on labeled data.



As deciding whether a preliminary analysis will become a disciplinary process or not is still discretionary and depends on a deeper observation of the facts, the main goal of the project is not predicting the right answer a 100% of the time. In fact, the objective is giving a direction on how decisions are made in order to help resource planning and related issues. Given these premises, if the model achieves an accuracy above 75% and f-score above 65% it succeeded in its objective.

Benchmark Model

The first benchmark created to test the model is a naïve classifier to make random guesses. It will establish a checkpoint in order to show how good can be use a ML model instead of just flip a coin and let it decide for us. As the naïve classifier doesn't require training, it will not set precedents related to training time.

Afterward we will compare three different algorithms and choose the best among them.

Finally, we will test different parameters for chosen model.

The expectation is keeping the training time bellow 10 seconds and predicting time bellow 500ms.

The two values above were determined by considering a reasonable response time for a web service in order to get a good user experience.

Evaluation Metrics

The goal of the ML model is to learn patterns that generalize well for unseen data instead of just memorizing known data as its target, i.e. overfitting to training data.

Typical metrics applied to binary classifiers are accuracy, precision, recall, and F1-measure. Each metric measures a different aspect of the predictive model. Accuracy measures the fraction of correct predictions. Precision measures the fraction of actual positives among those examples

predicted as positive. Recall measures how many actual positives were predicted as positive. F1-measure is the harmonic mean of precision and recall.

In order to check if the model is performing well on unseen examples it takes into account accuracy and F1-measure. Both metrics get a score in the range [0, 1].

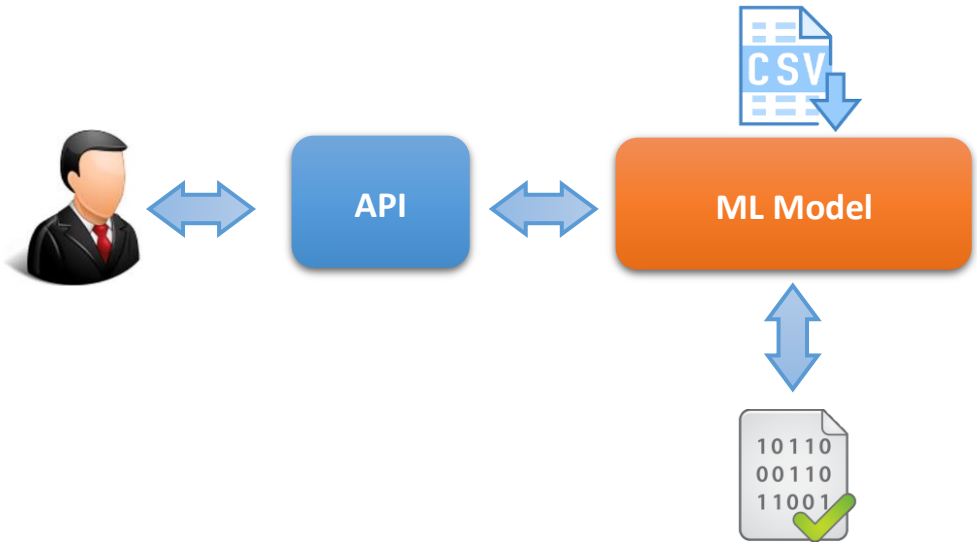
$$accuracy = \frac{TP + TN}{n} \qquad F = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

The module *sklearn.metrics* provides the function *accuracy_score* to compute the accuracy, either the fraction (default) or the count (normalize=False) of correct prediction. The function returns a value to maximize, the higher the better.

Considering the imbalance of the target labels (see Datasets and Inputs), for the purposes of the model, it was decided to adopt F-beta score weighted, which calculate metrics for each label, and find their average, weighted by support (the number of true instances for each label).

It is also important to keep track of the training and predicting time to ensure a good user experience.

Project Design



Basically, the project consists in a machine learning model that loads a comma separated values data file and stores the pre-trained model in a pickle file. If the pickle file is not present, the application runs the training and save the file. Then keep waiting the requests to make predictions and send the response.

Loading and pre-processing data

First of all, we must load all required libraries. After that, we need to load data from .csv file. By default, SQL Server 2014 uses semicolon (;) as separator when exporting query results. So, we should use the following code to get data loaded:

```
data = pd.read_csv(in_file, sep=';')
```

Then, discard all rows that do not have significant meaning for predicting outputs, like sequential and date, for instance.

```
data = data.drop(['nu_analise', ..., 'dt_inicio_analise'], axis = 1)
```

Also, we must clean all 'NaN' occurrences of data frame:

```
 data = data.dropna(axis=0)
```

With clear data we now can store the target features in a different variable a remove it from original data frame:

```
outcomes = data['cd_situacao']

data = data.drop('cd_situacao', axis = 1)
```

With these steps done, the data is prepared to serve as input for a machine learning model.

Implementation

The implementation process can be split into two main stages:

1. Choose the best classifier
2. Implement the best classifier
3. The application development stage

During the first stage, the classifier is trained on the preprocessed training data. This stage can be subdivided into the following steps:

1. Split training and testing data

   ```
   X_train, X_test, y_train, y_test = train_test_split(data, outcomes, test_size = 0.2,
   random_state = 0)
   ```

2. Create a naïve classifier to analyze the performance of random guessing
3. Create the evaluation metrics to test the models
4. Create three models using Gradient Boosting Classifier, Logistic Regression, and AdaBoost Classifier
5. Test the models and collect the results
6. Choose the best model based on results
7. Perform grid search on the classifier in order to get the optimized parameters

In the second stage we take the following steps:

1. Build the classifier using the optimized parameters
2. Train the classifier using training data

3. Serialize the model using Pickle

Finally, in the last stage we build an application to:

1. Load pickle file
2. Receive requests
3. Make predictions
4. Return answers

---

[i] http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html