The Lightweight IBM Cloud Garage Method for Data Science

Architectural Decisions Document

1.1 Data Source

All the data has been collected from "**Kaggle.com/"**

1.1.1 Technology Choice

The technology I had used here is Machine learning approach.

1.1.2 Justification

I used it because it helps in getting more access to the data that we want to work with.

Problem Statement:
There are many people around the globe that are complaining that they are being charged for the transactions that they never processed.
Therefore, Our model predicts that whether a our transaction is fraud or genuine based on the dataset.

Justification of Problem Statement:
Why you anybody pay if they did nothing i.e. if you did not purchased anything will you pay?

ETL

Data Quality Assessment:

1. The analysis on the independent variables are conducted.

2. Plots were visualized
3. Drop duplicate entries
4. Mean, variance etc were computed

Data Exploration:

- Correlation using inbuilt .corr() function

Data Visualization:

- Library used: Seaborn, Matplotlib
- Plots used: BAR, Heatmap, and Histogram

Feature Engineering:

1. Imputing missing values using forward fill

Justification: Since the data is almost similar to a time series, the best imputation technique would be to take the forward values(previous values) and impute it in the place of missing value.

Model Performance Indicator:

1. Recall_score
2. Accuracy_score
3. F1-Score

Justification: A good example is mentioned below, highlighting the importance of a good Recall score:

Recall helps when the cost of false negatives is high. What if we need to detect incoming nuclear missiles? A false negative has devastating consequences. Get it wrong and we all die. When false negatives are frequent, you get hit by the thing you want to avoid. A false negative is when you decide to ignore the sound of a twig breaking in a dark forest, and you get eaten by a bear.

In our case, it is required that we must be absolutely certain that a hack is about to happen so that the cyber security team can be ready when it occurs. Else there will be wastage of resources and man power, and the

effect can lead to loss of privacy. Thus it is used as the metric to determine the quality of our model.

Traditional ML algorithm:

1. Random Forest Classifier
2. Logistic Regression

Model Performance without Feature Selection:

Recall = 99.89%

Model Performance with Feature Selection:

Recall = 100%

Questions:

1) Why have I chosen a specific method for data quality assessment?

   The data quality is a very important aspect. The first step is to identify the statistics such as mean, variance etc. It will give a good idea and also check for skewness and kurtosis etc. The next step is to drop duplicates, clean and visualize the data. Various visualizations are plotted to get a clear idea of the distribution.

2) Why have I chosen a specific method for feature engineering?

   From the analysis conducted earlier it has been observed that there are missing values present in the data and they have to be imputed.

   There are several ways to impute the missing values, they are;

   Mean or median or other summary statistic substitution

   Maximum likelihood imputation

   frequency imputation

simple imputation

Multiple Imputation

Since data is almost similar to a time-series problem, we can use the technique of filling the data with forward fill and backward fill.

3) Why have I chosen a specific algorithm?

There are numerous ML and DL algorithms, we have chosen Random forest, since the algorithm uses decision tree and groups the data together, they would be useful to classify based on the independent variables. Logistic regression was also used, however random forest outperformed the former. The n_neighbors is the hyperparameter tuned and the best value was found to be 1000. With the inclusion of Deep Learning Algorithms, an improvement of the Random forest is the XGBoost, which works even better since it uses gradient boosting too.

4) Why have I chosen a specific framework?

Since the data available is small in size, it does not require much computation resources. Thus Scikit learn provided me with easy of use and easy implementation. Since this model does not need to be deployed, rather a simple prediction would be sufficient, scikit learn with a tensorflow background has a greater advantage.

5) Why have I chosen a specific model performance indicator?

Recall is the metric chosen.

A good example is mentioned below, highlighting the importance of a good Recall score:

Recall helps when the cost of false negatives is high. What if we need to detect incoming nuclear missiles? A false negative has devastating consequences. Get it wrong and we all die. When false negatives are frequent, you get hit by the thing you want to avoid. A false negative is when you decide to ignore the sound of a twig breaking in a dark forest, and you get eaten by a bear.

In our case, it is required that we must be absolutely certain that a hack is about to happen so that the cyber security team can be ready when it occurs. Else ther will be wastage of resources and man power, and the effect can lead to loss of privacy. Thus it is used as the metric to determine the quality of our model.