# Machine Learning Engineer Nanodegree

## Capstone Proposal

Ramakanth Vanga,
May 27th, 2018

## Proposal

### Domain Background
Charge Account Prediction for Invoice lines is the project I would like to work on. Typical medium and large business companies have an ERP system and an Accounts Payable team who receives Invoices from Vendors, create Invoices in the ERP system and pays Vendors against the Invoices created. Based on the services/goods provided by Vendor Accounts Payable team will specify appropriate GL Account (Charge account) on each line on the Invoice. This is a repetitive process every day and specifying the appropriate GL Account comes with experience and memory.

### Problem Statement
Specifying Charge Account on Invoice line is a repetitive task where several parameters define the final charge account. These final values don't change and users keep on doing the same task over and over. When AP receives an Invoice from Vendor, it makes sense to look at the history of Invoices from the same Vendor and predict the charge account. This will save a lot of time in processing Invoices and will increase the overall productivity.

Charge account is dependent on several factors like Vendor Country, Payment Method, and Payment Terms etc. These factors are independent from each other so Naïve Bayes algorithm might be a good solution for this problem.

### Datasets and Inputs
When a new Invoice is received from a vendor, all the old Invoices lines for the same vendor can be considered as training set. Current Invoice lines will be considered as prediction/test set.

I have 673,150 records available for training. All the features are characters so one hot encoding is necessary for all the features. Will use only last one-year data, using more or less than that might result in over fitting or under fitting.

## Benchmark Model

For every new Invoice when user manually specifies a charge account, there is an entry in audit table, which has a record for old and new value. With this algorithm predicted results we can compare the value predicted with the charge account value for recent Invoice in audit table. The count of entries in field audit table will be less for a new Invoice when charge accounts prediction is successful.

For the features of test set, count of distinct charge accounts can be fetched and compared to predicted charge account. In most of the cases, there should be only 1 or 2 different charge accounts, if the results are not correct it simply means we didn't account for a feature or a mistake in data preprocessing.

## Solution Statement

Naïve Bayes algorithm could be used to predict charge account, as occurrence of each feature in the input data set is independent of other features. New Invoices from Vendors keeps coming in and model gets better with more available training data.

Users spend a lot of time in specifying a charge account on each Invoice line, which is time consuming. Auto predicting charge accounts will decrease the overall processing time and improves the efficiency.

## Evaluation Metrics

We can measure the model by comparing the predicted charge accounts and charge accounts of the old Invoices. We only look at previous Invoices for the same Vendor as comparison metric if needed.

## Project Design

1. N features are needed to predict charge account, N value varies from customer to customer based on the business requirement. This will be crucial to come up with a design where these N features can be dynamically changed from customer to customer.
2. Python script should be able to access these features and charge account id values either by reading from Oracle database directly or CSV file or from JSON requests.

3. These features could be characters so it is needed to convert them to a numerical value. Converting this can be done using label encoder or hot encoding.
4. Y value which is charge account are characters too so converting them to numerical is important and have the capability to get the character value for every numerical value is needed.
5. Model score is important which will help users to understand how good or bad a model is.
6. Naïve Bayes algorithm seems to be the right choice for this. It would be interesting to see the results of Logistic Regression too.