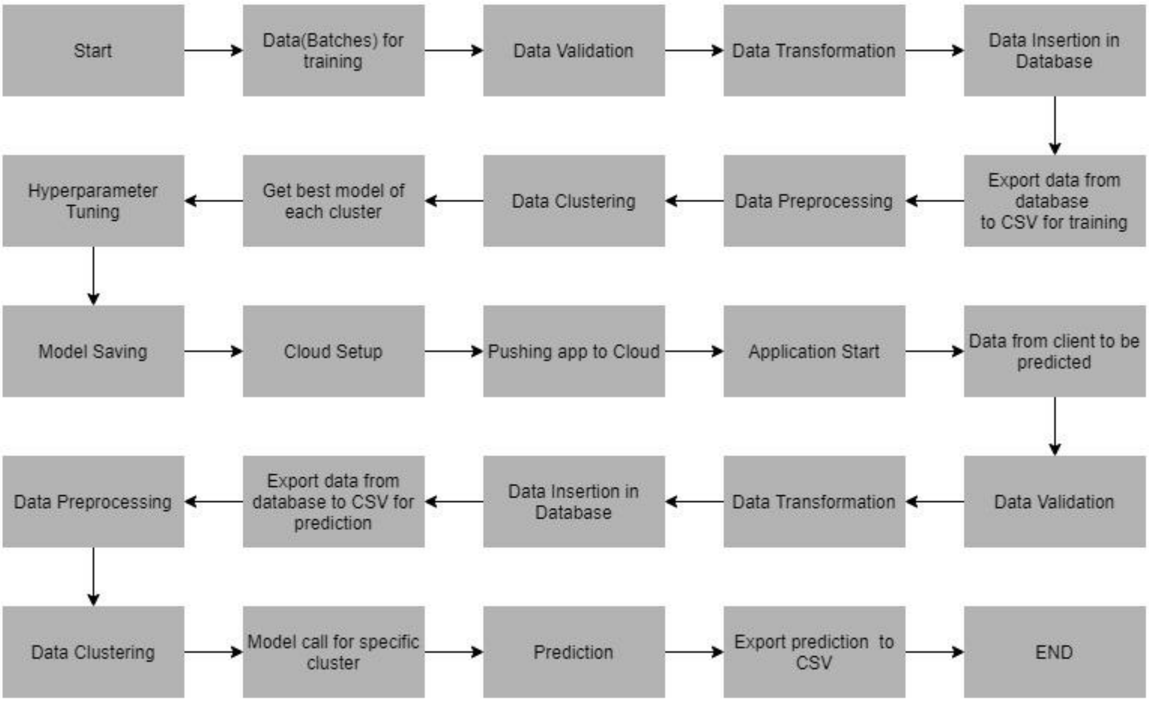
Problem Statement

To build a classification methodology to determine whether a person defaults the credit card payment for the next month.

Architecture


Start
Data(Batches) for training
Data Validation
Data Transformation
Data Insertion in Database
Export data from database to CSV for training
Data Preprocessing
Data Clustering
Get best model of each cluster
Hyperparameter Tuning
Model Saving
Cloud Setup
Pushing app to Cloud
Application Start
Data from client to be predicted
Data Validation
Data Transformation
Data Insertion in Database
Export data from database to CSV for prediction
Data Preprocessing
Data Clustering
Model call for specific cluster
Prediction
Export prediction to CSV
END

Data Description

The client will send data in multiple sets of files in batches at a given location. The data has been extracted from the census bureau.

The data contains 32561 instances with the following attributes:
Features:

1. **LIMIT_BAL:** continuous.Credit Limit of the person.
2. **SEX:** Categorical: 1 = male; 2 = female
3. **EDUCATION:** Categorical: 1 = graduate school; 2 = university; 3 = high school; 4 = others
4. **MARRIAGE:** 1 = married; 2 = single; 3 = others
5. **AGE**-num: continuous.
6. **PAY_0 to PAY_6:** History of past payment. We tracked the past monthly payment records (from April to September, 2005)

7. **BILL_AMT1 to BILL_AMT6:** Amount of bill statements.
8. **PAY_AMT1 to PAY_AMT6:** Amount of previous payments.

Target Label:

Whether a person shall default in the credit card payment or not.

9. default payment next month: Yes = 1, No = 0.

Apart from training files, we also require a "schema" file from the client, which contains all the relevant information about the training files such as:

Name of the files, Length of Date value in FileName, Length of Time value in FileName, Number of Columns, Name of the Columns, and their datatype.

Data Validation

In this step, we perform different sets of validation on the given set of training files.

1. Name Validation- We validate the name of the files based on the given name in the schema file. We have created a regex pattern as per the name given in the schema file to use for validation. After validating the pattern in the name, we check for the length of date in the file name as well as the length of time in the file name. If all the values are as per requirement, we move such files to "Good_Data_Folder" else we move such files to "Bad_Data_Folder."

2. Number of Columns - We validate the number of columns present in the files, and if it doesn't match with the value given in the schema file, then the file is moved to "Bad_Data_Folder."

3. Name of Columns - The name of the columns is validated and should be the same as given in the schema file. If not, then the file is moved to "Bad_Data_Folder".

4. The datatype of columns - The datatype of columns is given in the schema file. It is validated when we insert the files into Database. If the datatype is wrong, then the file is moved to "Bad_Data_Folder".

5. Null values in columns - If any of the columns in a file have all the values as NULL or missing, we discard such a file and move it to "Bad_Data_Folder".

Data Insertion in Database

1) Database Creation and connection - Create a database with the given name passed. If the database has already been created, open a connection to the database.
2) Table creation in the database - Table with name - "Good_Data", is created in the database for inserting the files in the "Good_Data_Folder" based on given column names and datatype in the schema file. If the table is already present, then the new table is not created, and new files are inserted in the already present table as we want training to be done on new as well as old training files.
3) Insertion of files in the table - All the files in the "Good_Data_Folder" are inserted in the above-created table. If any file has invalid data type in any of the columns, the file is not loaded in the table and is moved to "Bad_Data_Folder".

Model Training

1) **Data Export from Db** - The data in a stored database is exported as a CSV file to be used for model training.

2) **Data Preprocessing**

a) Check for null values in the columns. If present, impute the null values using the categorical imputer.
b) Scale the numeric values using the standard scaler.
c) Check for correlation.

3) **Clustering -** KMeans algorithm is used to create clusters in the preprocessed data. The optimum number of clusters is selected by plotting the elbow plot, and for the dynamic selection of the number of clusters, we are using "KneeLocator" function. The idea behind clustering is to implement different algorithms

The Kmeans model is trained over preprocessed data, and the model is saved for further use in prediction.

4) **Model Selection –** After the clusters have been created, we find the best model for each cluster. We are using two algorithms, “Naïve Bayes” and "XGBoost". For each cluster, both the algorithms are passed with the best parameters derived from GridSearch. We calculate the AUC scores for both models and select the model with the best score. Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction.

Prediction Data Description

The Client will send the data in multiple sets of files in batches at a given location. Data will contain the credit and payment record of customers.

Apart from prediction files, we also require a "schema" file from the client, which contains all the relevant information about the training files such as:

Name of the files, Length of Date value in FileName, Length of Time value in FileName, Number of Columns, Name of the Columns and their datatype.

Data Validation

In this step, we perform different sets of validation on the given set of training files.

1) Name Validation- We validate the name of the files based on given Name in the schema file. We have created a regex pattern as per the name given in the schema file, to use for validation. After validating the pattern in the name, we check for the length of date in the file name as well as the length of the timestamp in the file name. If all the values are as per requirement, we move such files to "Good_Data_Folder" else we move such files to "Bad_Data_Folder".

2) Number of Columns - We validate the number of columns present in the files, and if it doesn't match with the value given in the schema file, then the file is moved to "Bad_Data_Folder".

3) Name of Columns - The name of the columns is validated and should be same as given in the schema file. If not, then the file is moved to "Bad_Data_Folder".

4) Datatype of columns - The datatype of columns is given in the schema file. This is validated when we insert the files into Database. If the datatype is incorrect, then the file is moved to "Bad_Data_Folder".

5) Null values in columns - If any of the columns in a file has all the values as NULL or missing, we discard such file and move it to "Bad_Data_Folder".

Data Insertion in Database

1) Database Creation and connection - Create a database with the given name passed. If the database is already created, open the connection to the database.

2) Table creation in the database - Table with name - "Good_Data", is created in the database for inserting the files in the "Good_Data_Folder" based on given column names and datatype in the schema file. If the table is already present, then a new table is not created, and new files are inserted into the already present table as we want training to be done on new as well old training files.

3) Insertion of files in the table - All the files in the "Good_Data_Folder" are inserted in the above-created table. If any file has invalid data type in any of the columns, the file is not loaded in the table and is moved to "Bad_Data_Folder".

Prediction

1) Data Export from Db - The data in the stored database is exported as a CSV file to be used for prediction.

2) Data Preprocessing :

a) Check for null values in the columns. If present, impute the null values using the categorical imputer.
b) Scale the numeric values using the standard scaler.
c) Check for correlation.

3) Clustering - KMeans model created during training is loaded, and clusters for the preprocessed prediction data is predicted.

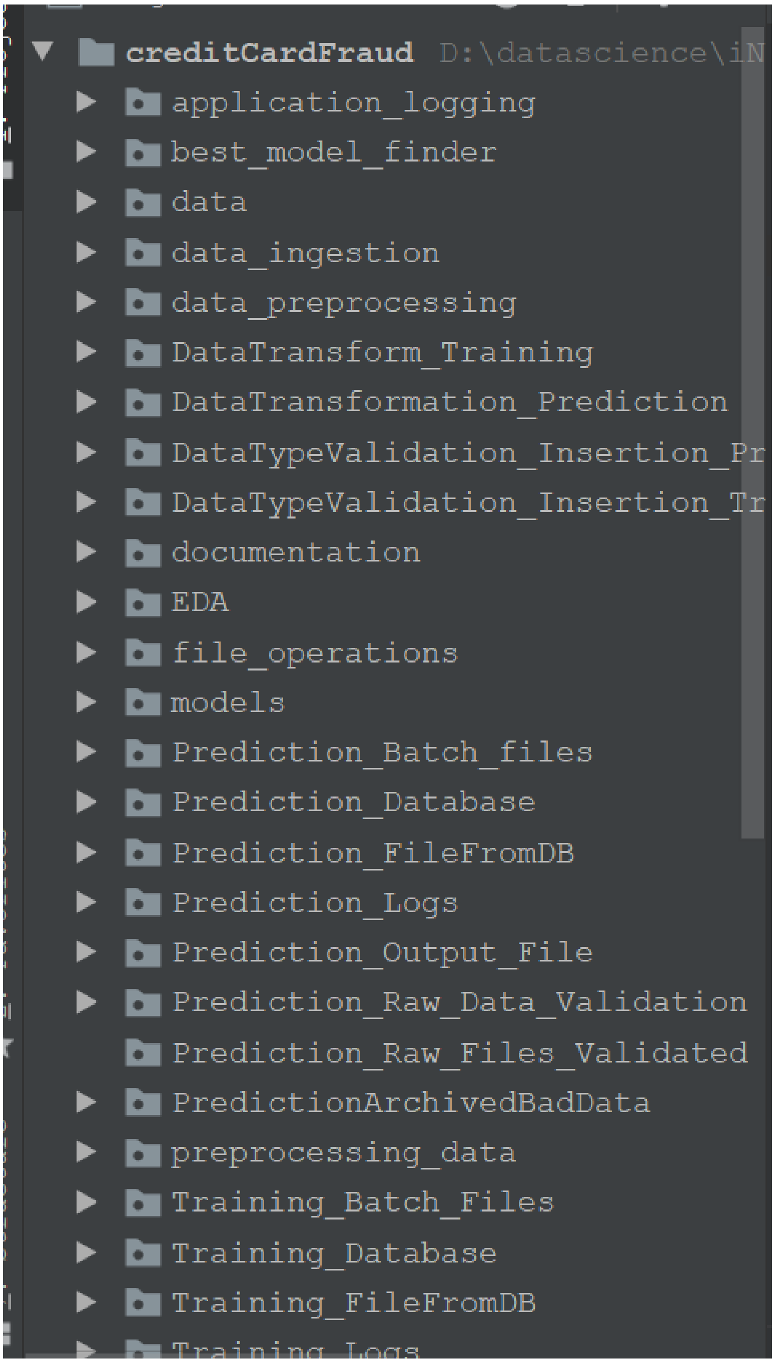
4) Prediction - Based on the cluster number, the respective model is loaded and is used to predict the data for that cluster.

5) Once the prediction is made for all the clusters, the predictions along with the Wafer names are saved in a CSV file at a given location, and the location is returned to the client.

Deployment

We will be deploying the model to the Heroku cloud platform.

Now let’s look at the project folder structure:



```
▼ creditCardFraud D:\datascience\iN
    ▶ application_logging
    ▶ best_model_finder
    ▶ data
    ▶ data_ingestion
    ▶ data_preprocessing
    ▶ DataTransform_Training
    ▶ DataTransformation_Prediction
    ▶ DataTypeValidation_Insertion_Pr
    ▶ DataTypeValidation_Insertion_Tr
    ▶ documentation
    ▶ EDA
    ▶ file_operations
    ▶ models
    ▶ Prediction_Batch_files
    ▶ Prediction_Database
    ▶ Prediction_FileFromDB
    ▶ Prediction_Logs
    ▶ Prediction_Output_File
    ▶ Prediction_Raw_Data_Validation
      Prediction_Raw_Files_Validated
    ▶ PredictionArchivedBadData
    ▶ preprocessing_data
    ▶ Training_Batch_Files
    ▶ Training_Database
    ▶ Training_FileFromDB
    ▶ Training_Logs
```