Project 1: “Implementation of Churn Prediction Machine Learning Model”

Objective:

Developed batch predictive model for Customer churn for Telecomm Domain. Telecommunication industry has suffered from high churn rates and immense churning loss. Although the business loss is unavoidable, but still churn can be managed and kept in an acceptable level.

Benefits:

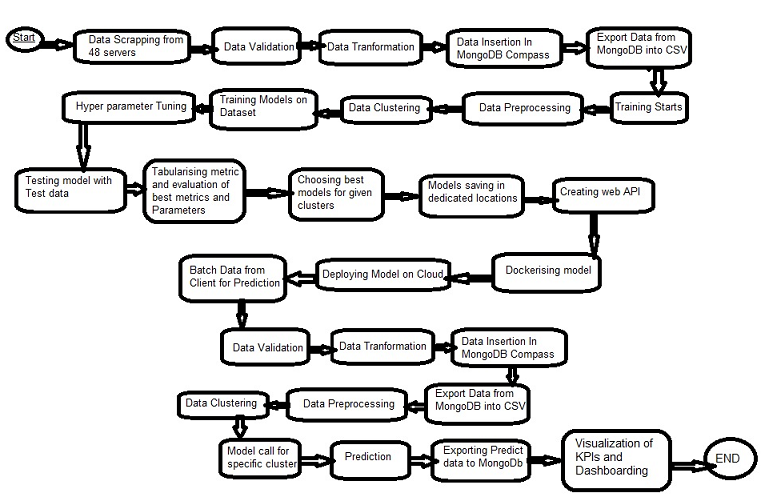
* Able to deliver tailored promotions and offers to positively influence their behavior of likely to churned customers.
* Able to segment the customers based on behavior and demographics to improve retention.
* Improved Service Standards.
* Increased Average Revenue per customer (ARPC) of company.
* it can cost five times more to attract a new customer than it does to retain an existing one, Increasing customer retention rates by 5% can increase profits by 25% to 95%.

Data Sharing Agreement:

Description of the Data:

* Standard file with schema definitions of data
* Defined number of columns and data types
* Source information and credentials sharing

Architecture:



Data Description:

Client has four data centers, data is distributed among 48 servers across pan India, North and south data centers are using SAP CRM and South and East are using Sible CRM.

Data scrapped from all databases by ETL pipelines which scheduled monthly basis. On first go client provided 30 months. data contains a total of nine million+ customers with 21 attributes, coming from personal characteristics, services signatures, and contract details. Last column in dataset is churn details with unique values 1 and 0. 1 means customer churned and 0 means not churned.

Data Validation:

In ETL pipeline, first a view is created in specific server database, data validation pipeline which checks following view and table level validations in one by one server for all servers according to master data management file (MDM) prepared upon DSA.

* View names, Table names, and schema names validation
* Number of records in views for specific month should be greater zero.
* Number of columns in views.
* Name of columns in views.
* Data types of columns in views.
* All Null values in records of column
* Range check of values

If any above validation fails, sending that log to responsible person through mail.

Data Insertion in MongoDB:

1. Database Creation and connection - Created collection with the given name passed in MongoDb. If the collection is already created, open the connection to the collection, it acts as central repository to have 30 months data.
2. First we delete oldest month data and add last month data to the collection.

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 KNN imputers b) Check if any column has zero standard deviation, remove such columns as they don't give any information during model training.
3. Handled categorical features with one hot encoding method.
4. 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 to train data in different clusters. The Kmeans model is trained over preprocessed data and the model is saved for further use in prediction.
5. Model Selection - After clusters are created, we find the best model for each cluster. We experimented most of ML algorithms. For each cluster, all the algorithms are passed with the best parameters derived from GridSearch. We calculated the AUC scores for both models and selected 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

1. Client provided data as a CSV file in remote location to be used for prediction.
2. Created a pipeline to check data validations in prediction data. If it fails it has on validation, it creates log which is going to mail to responsible person.
3. Performed Data Preprocessing on prediction data. a) Checked for null values in the columns. If present, imputed the null values using the KNN imputer. b) Checked if any column has zero standard deviation, removed such columns as we did in training. c) handled categorical features
4. 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 customer details and names are saved in a CSV file at a remote location and the location is returned to the client.

6) Using this Prediction CSV we dishoarded the KPIs using Power BI.

Project 2: “Forecasting Smart meter Energy usage using Machine Learning Model”

Objective:

Developed batch forecast model for Energy usage for Utility Domain. This model developed for smart meter which sends meter reading data for every 15 minutes of interval. Model developed on dataset is of 96 million (7GB) observation.

Benefits:

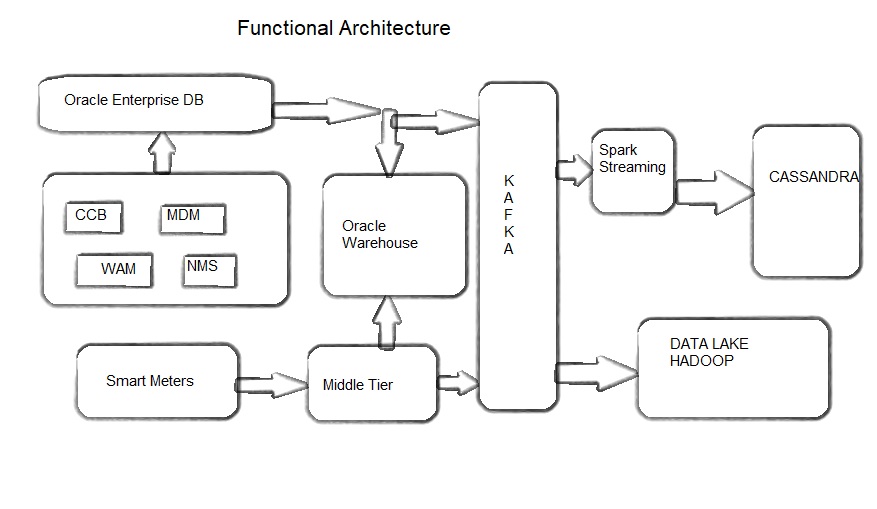
* Load forecasts helped suppliers to balance supply and demand
* Forecasting data helped Energy providers to balance their electricity purchase and sales.
* Load forecasting can performed in a wide range of time horizons aiming at different targets: short-term load forecast (a few minutes to 1 day ahead) to adjust supply and demand; medium-term load forecast (1 day to 1 year ahead) to plan outage and maintenance and long-term load forecast (more than 1 year ahead) to plan the development of power infrastructures
* Able to segment the customers into groups based on usage behavior and demographics.
* Improved Service Standards.
* Increased Average Revenue per customer (ARPC) of company.
* Suggested customers to use energy in Peak times during day.

Data Sharing Agreement:

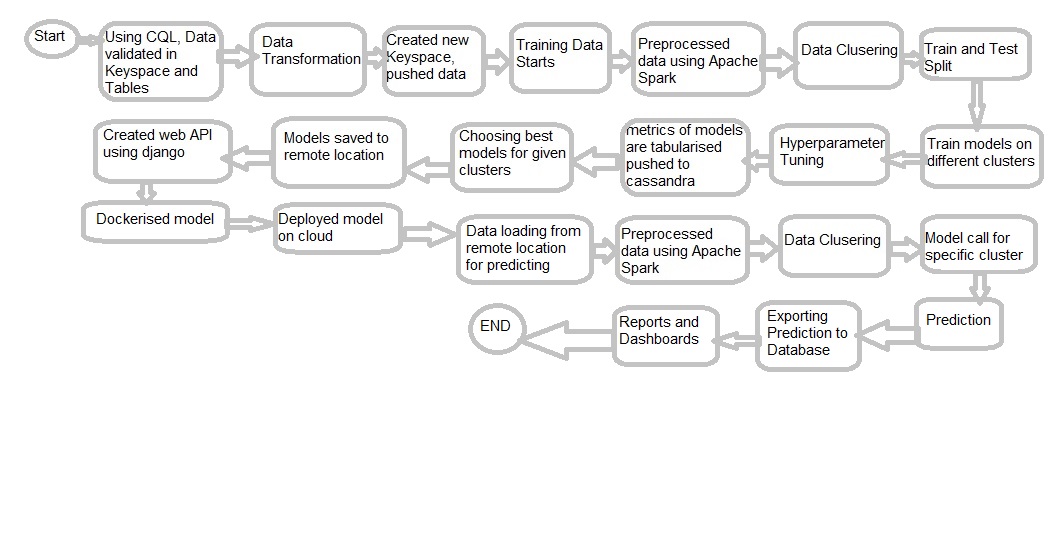
Description of the Data:

* Standard file with schema definitions of data
* Defined number of columns and data types
* Source information and credentials sharing

Functional Architecture:



Architecture:



Data Description:

Client has data on Cassandra DB, data pushed to Cassandra using Apache Spark on HDP. Data in the form of JASON from smart meters is sent KAFKA where data stored in the form distributed fashion. Apache spark consumes data from kakfa sent to Cassandra and Historical storage to Data lake on Hadoop as parquet format. .

Data Validation:

data validation pipeline is created which checks following table level validations using CQL, data level check is done according to master data management file (MDM) prepared upon DSA.

* Table names, and schema names validation
* Number of records in views for specific month should be greater zero.
* Number of columns in views.
* Name of columns in views.
* Data types of columns in views.
* All Null values in records of column

If any above validation fails, sending that log to responsible person through mail.

Data Insertion in New Train table:

Created a pipeline to created new table and push data to new table for training.

Model Training:

1. Data Preprocessing: a) Check for null values in the columns. If present, impute the null values using the KNN imputers b) Check if any column has zero standard deviation, remove such columns as they don't give any information during model training.
2. We then computed the mean energy consumption in that 15 minute period by averaging the energy consumption in that time period across the previous 10 days, generating a single feature. Therefore, there are 48 half-hour increments in a day, resulting in 96 additional features.
3. A final categorical feature was also included indicating household income, The three income classifications are Affluent, Comfortable, and Adversity. Handled these categorical features with one hot encoding method. Temperature max and Temp\_min also added to dataset.
4. 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 to train data in different clusters. The Kmeans model is trained over preprocessed data and the model is saved for further use in prediction.
5. Model Selection - After clusters are created, we find the best model for each cluster. We experimented most of ML algorithms. For each cluster, all the algorithms are passed with the best parameters derived from GridSearch. We calculated the AUC scores for both models and selected 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

1. Importing data for prediction in remote location to be used for prediction.
2. Created a pipeline to check data validations in prediction data. If it fails it has on validation, it creates log which is going to mail to responsible person.
3. Performed Data Preprocessing on prediction data. a) Checked for null values in the columns. If present, imputed the null values using the KNN imputer. b) Checked if any column has zero standard deviation, removed such columns as we did in training. c) handled categorical features
4. 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 customer details and names are saved in Cassandra database.

6) Using this Prediction data we developed dishoards with KPIs using Tableau.