Detailed Project Report(DPR)

Insurance Premium Prediction

Version 1.0

25/08/2021

Version Control

Version	Description	Responsible Party	Date
1.0	Initial version	Motheeshkumar Jay	25-08-2021

URL Details:

IPP application URL: https://insurance-premium--prediction.herokuapp.com/

Objective

Development of a predictive model is to give people an insurance premium estimate of how much they need based on their individual health situation. After that, customers can work with any health insurance carrier and its plans and perks while keeping the projected cost from our study in mind.

Benefits

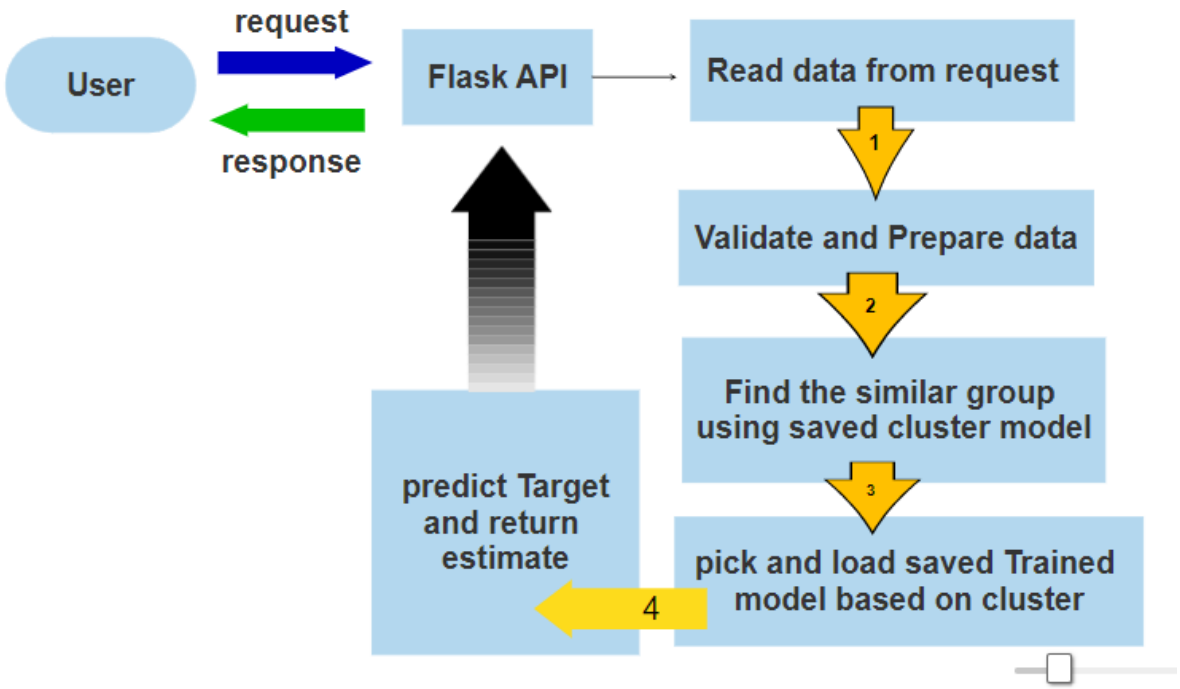
1. Prediction of estimate for insurance premium.
2. Gives better insight to the customer about insurance premium.

Data Sharing Agreement

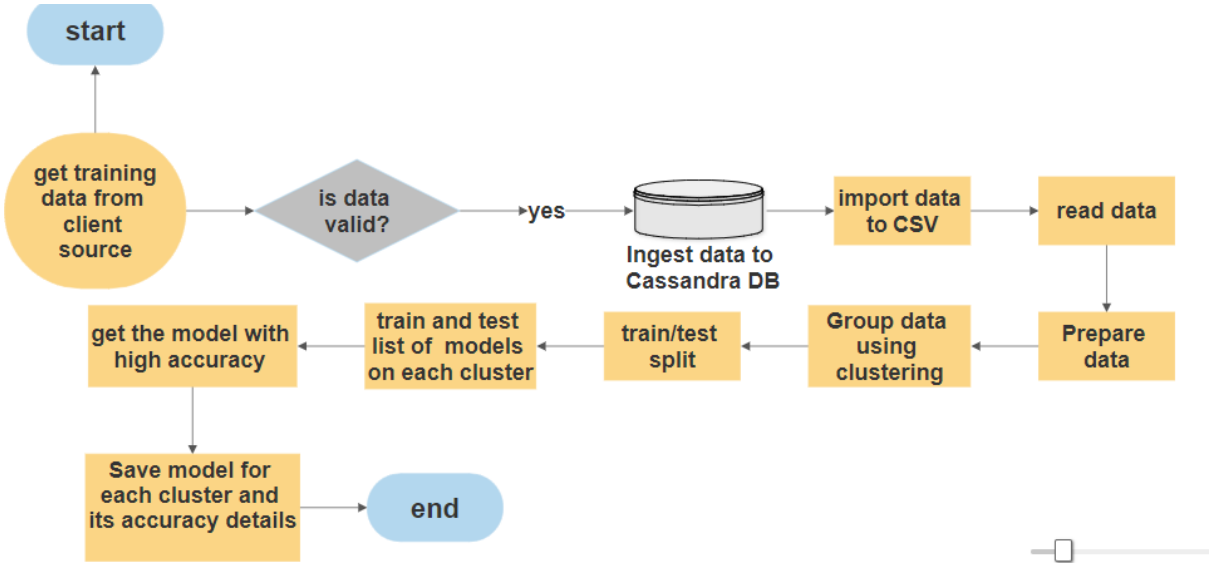
1. Sample file name (ex Insurance_20062021). we can change the format match by editing the data schema.
2. Length of date stamp(8 digits)
3. Number of Columns
4. Column names
5. Column data type

Architecture

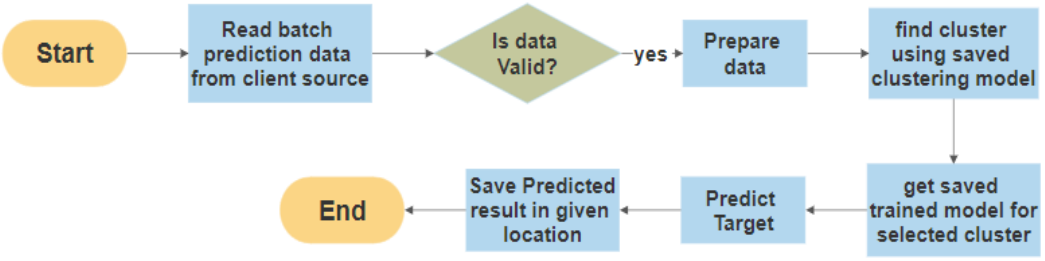
IPP Web application workflow:



Model Training workflow:



Batch Prediction workflow:



Data Validation

Data source and format

In this project, we are going to read a set of datasets from a client source where the client will be updating the training dataset whenever they want.

Data format:

CSV from client source

Data schema

Define structure in JSON format of data set which allows us to validate data structure and create table structure based on that. Training batch data should be read from client source.

Feature validation

Check whether all features are present or not. Check all the names and data types are matching with predefined data schema.Check if any of the features have missing values >95 percent. if yes reject dataset and move to bad dataset folder else move to good dataset folder.once data validation done prepare data for database by replacing missing values with NULL values.If Data validation is successful move data set to good data folder from client dataset source.If Data validation is failed move data set to bad data folder from client dataset source and specify the reason for rejection of data.

Data Ingestion

Database details

Database - Cassandra

Version - cqlsh 6.8.0

Read data

Read all the valid dataset from a good data folder and merge it in one data frame.

create table for dataset

Create a table based on data schema dynamically. drop if it already exists and create a new table with a data schema.

insert data into table

Once the table creation is successful, add all the valid data to the newly created table.

Read data and convert to CSV file.

Once the data is added, read the data from DB and convert it to a single CSV file for training the model.

Data preprocessing

This step involves data cleaning, data transformation and feature selection. We will be preparing data for machine learning models.

Data cleaning - imputing null values with KNN weighted average

Data Transformation - feature encoding and feature scaling process.

Feature selection - removing constant features, removing duplicate features, removing irrelevant features

Preserve Data Structure - Once we complete data preprocessing, feature column names will get changed after feature encoding. so we will save the feature names as pickle files. In future, we will use this data to check the column name match so that we won't get any column mismatch error while predicting using the saved model.

Data Clustering

In order to get the good accuray , we are using K-means clustering algorithm to cluster the similar data and then we will try to train a different ML algorithm on each cluster. we will choose the model with high accuracy and without overfitting.

clustering involves below steps:

1. Finding the K number of clusters using elbow method(KneeLocator)
2. Then we will try to cluster data using k-means clustering algorithm
3. Once we train we will save the clustering model for future prediction.

Model building

Once the data preparation is ready, model building involves the below step.

Test/ train split:

We will split each cluster into train and test data.

Training model:

We will try Random forest and Gradient boosting regressor algorithms on each cluster.

Testing:

Once we train the model, we will test each model with test data and pick the model with high accuracy without overfitting

Overfitting Threshold:

In this use case, we are using 5% as an overfitting threshold. We will reject models if test accuracy is less than 5% than training accuracy.

Saving Model:

Once we choose the model with high accuracy, we will save the model for prediction in pickle format and also save the model details.

Data Prediction

Batch data prediction:

Batch data prediction involves the steps below.

1. Reading data from batch prediction dataset location.
2. validating the data.
3. prepare data by imputing null values, encoding categorical values.
4. Once we are done with data preparation, we will use a saved clustering model to find which cluster it belongs to.
5. Then, we will load a saved model based on the cluster and we will predict the estimation.
6. Once we predict the data, we will save the result in the required destination.

UI based prediction from users:

In this process, we will use the same steps as batch data prediction but for a single record. We will follow the steps below.

1. Receive request from UI.
2. Read data from requests.
3. We will follow the same steps as batch prediction.
4. Respond back to users with estimated value and response time.

Q & A:

Q1) What's the source of data?

The data for training is provided by the client in multiple batches and each batch contains multiple files.

File format: csv

Q 2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q 3) What's the complete flow you followed in this Project?

Refer slide 3rd and 4th for better Understanding

Q 4) After the File validation what do you do with incompatible files or files which didn't pass the validation?

Files like these are moved to the bad data folder for future investigation why it failed and share the info with the client.

Q 5) How are logs managed?

We are using different logs as per the steps that we follow in validation and modeling like File validation log , Data Insertion ,Model Training log , prediction log etc. Each and every life cycle of model training, data prediction logs are maintained in different folders so that it will be easy to track and debug.

Q 6) What techniques were you using for data pre-processing?

1. Removing constant features.
2. Imputing null values using KNNImputer and Kneelocator to find K.
3. Removing outliers
4. Since we are using rule based ML algorithms we are not doing data scaling.
5. Encoding categorical data.

Q 7) How training was done or what models were used?

1. Before dividing the data in the training and validation set we performed clustering to divide the data into clusters using **K-means clustering**.
2. As per cluster the training and validation data were divided.
3. Algorithms like **Random forest** and **Gradient boosting** were used based on the recall. The final model was used for each cluster and we saved that model .
4. We used **gridsearchCV** to try different combinations of parameters and to choose the best estimator.
5. Once we trained the model we chose the model with high accuracy and without overfitting from above 2 models for each cluster.
6. For each cluster we saved the best estimator and its details.

Q 8) How Prediction was done?

Batch prediction

1. loading the data from client source
2. validating the data same as training
3. data preprocessing same as training
4. matching the features same as trained features
5. finding the cluster using a saved clustering model.
6. predicting the target using a saved trained model for a particular cluster.
7. save the results in the batch prediction result folder.

Prediction for UI users:

1. receive requests from users.
2. validate data based on data schema.
3. convert json to data frame.
4. Encoding categorical features.
5. Find the cluster using a saved clustering model.
6. load the model for the selected cluster.
7. predict the target
8. respond back to the user with an estimate.

Q 9) How is the IPP application deployed?

The IPP application is deployed in AWS EC2 instance.

Q 10) How to access this application?

url to access this application
https://insurance-premium--prediction.herokuapp.com/