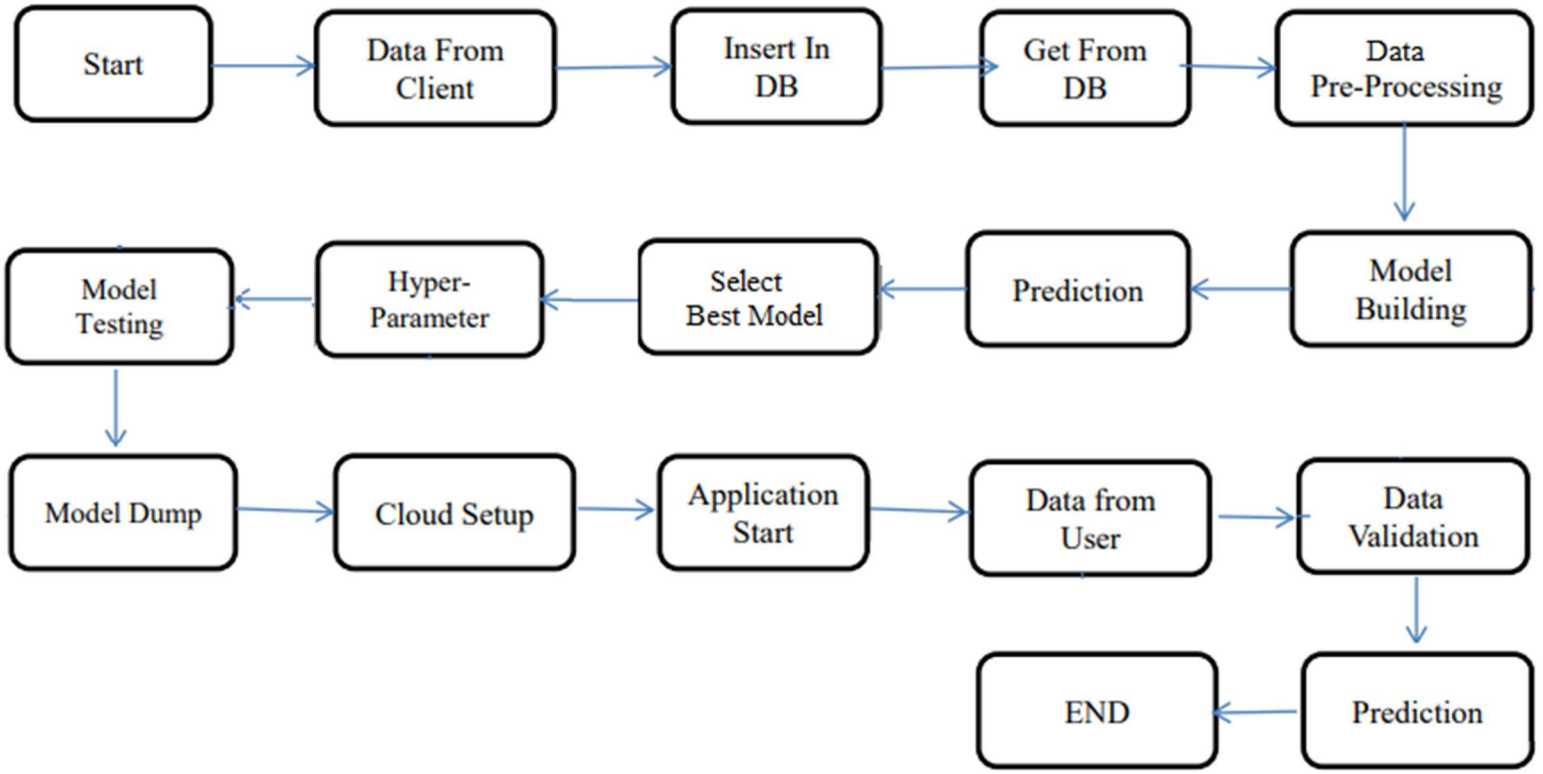
PREDICT CREDIT RISK
USING
SOUTH GERMAN BANK DATA

- OBJECTIVE

Credit analysis draws conclusions by evaluating the available quantitative and qualitative data regarding the creditworthiness of a client and making recommendations on whether or not to approve the loan application. The objective of credit analysis is to determine the risk of default that a client presents and assign a risk rating to each client. The risk rating will determine if the company will approve (or reject) the loan application, and if approved, the amount of credit to be granted.

- ARCHITECTURE

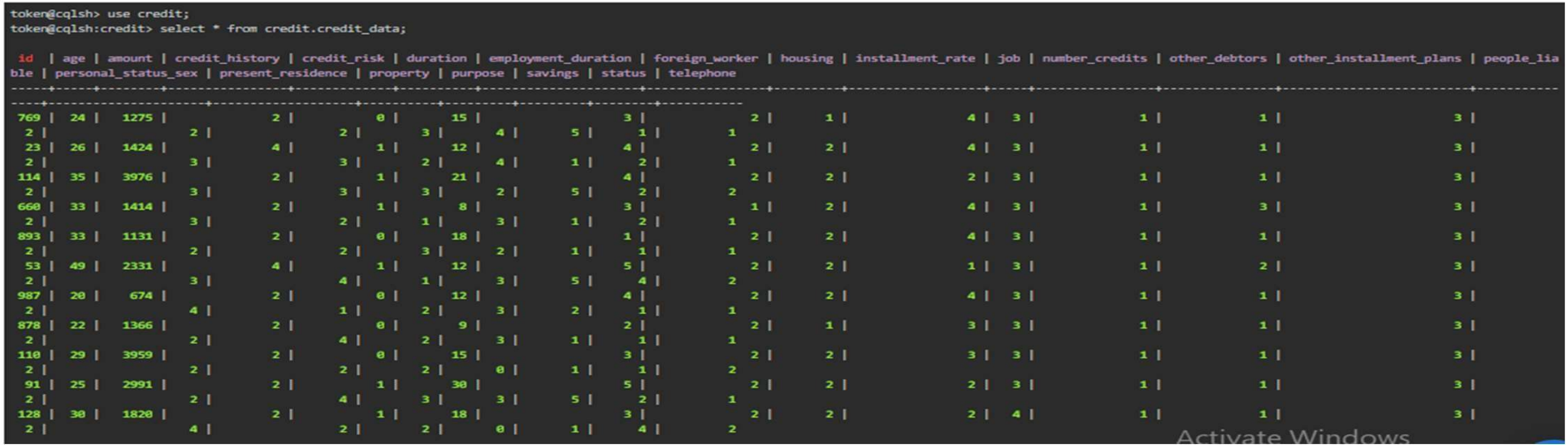


- Data From Client:-

We collect the data from client in .asc format in folder. In that we have get different file format, out of that that the .txt format file contain the feature name in German and English language.

- Insert Into Database & Get From Database:-

After gathering the data from client we have to put that data in database for future purpose. The database we use is Cassandra database for that we use DataStax Astra website. Our database name is “South German Bank Data” then the keyspace name is “credit” and last table is “credit_data”. Then we extract the data from database with help of query “SELECT * FROM credit.credit_data” and save in the file.



```
token@cqlsh> use credit;
token@cqlsh:credit> select * from credit.credit_data;

 id  | age | amount | credit_history | credit_risk | duration | employment_duration | foreign_worker | housing | installment_rate | job | number_credits | other_debtors | other_installment_plans | people_lia
ble | personal_status_sex | present_residence | property | purpose | savings | status | telephone
-----+-----+--------+----------------+-------------+----------+---------------------+----------------+---------+------------------+-----+----------------+---------------+-------------------------+----------
---+---------------------+-------------------+----------+---------+---------+--------+-----------
 769 |  24 |   1275 |              2 |           0 |       15 |                   3 |              2 |       1 |                4 |   3 |              1 |             1 |                       3 |
   2 |                   2 |                 2 |        2 |       3 |        4 |      5 |          1 |       1
  23 |  26 |   1424 |              4 |           1 |       12 |                   4 |              2 |       2 |                4 |   3 |              1 |             1 |                       3 |
   2 |                   3 |                 3 |        2 |       4 |        1 |      2 |          1
 114 |  35 |   3976 |              2 |           1 |       21 |                   4 |              2 |       2 |                2 |   3 |              1 |             1 |                       3 |
   2 |                   3 |                 3 |        3 |       2 |        5 |      2 |          2
 660 |  33 |   1414 |              2 |           1 |        8 |                   3 |              1 |       2 |                4 |   3 |              1 |             3 |                       3 |
   2 |                   3 |                 2 |        1 |       3 |        1 |      2 |          1
 893 |  33 |   1131 |              2 |           0 |       18 |                   1 |              2 |       2 |                4 |   3 |              1 |             1 |                       3 |
   2 |                   2 |                 2 |        3 |       2 |        1 |      1 |          1
  53 |  49 |   2331 |              4 |           1 |       12 |                   5 |              2 |       2 |                1 |   3 |              1 |             2 |                       3 |
   2 |                   3 |                 4 |        1 |       3 |        5 |      4 |          2
 987 |  20 |    674 |              2 |           0 |       12 |                   4 |              2 |       2 |                4 |   3 |              1 |             1 |                       3 |
   2 |                   4 |                 1 |        2 |       3 |        2 |      1 |          1
 878 |  22 |   1366 |              2 |           0 |        9 |                   2 |              2 |       1 |                3 |   3 |              1 |             1 |                       3 |
   2 |                   2 |                 4 |        2 |       3 |        1 |      1 |          1
 110 |  29 |   3959 |              2 |           0 |       15 |                   3 |              2 |       2 |                3 |   3 |              1 |             1 |                       3 |
   2 |                   2 |                 2 |        2 |       0 |        1 |      1 |          2
  91 |  25 |   2991 |              2 |           1 |       30 |                   5 |              2 |       2 |                2 |   3 |              1 |             1 |                       3 |
   2 |                   2 |                 4 |        3 |       3 |        5 |      2 |          1
 128 |  30 |   1820 |              2 |           1 |       18 |                   3 |              2 |       2 |                2 |   4 |              1 |             1 |                       3 |
   2 |                   4 |                 2 |        2 |       0 |        1 |      4 |          2
```

- Pre-Processing:-

In pre-processing there is different method are involved such as, filling up nan value, encoding categorical column etc. But in our dataset there is no nan value.

In our dataset the feature name is in German language so we have to convert it into English language that are given in dataset file.

Convert Numerical values with Original values in all categorical columns of Dataset, that are given in dataset file.

Drop few non important coloumns in dataset.

Converting the columns having ordinal values to Label Encoding. Converting the columns having non-ordinal values to One Hot Encoding

- Model Building:-

In the given step we divide the data into train test split. So that we can apply train and test data to the model. We are use various ML model for our project, such as Logistic Regression, Support Vector Machine, Random Forest Classifier, K- Nearest Neighbor and Naïve Bayes Classifier.

But problem is that our target feature is not equally distributed, so it will affect on the model accuracy. Here we create model with this imbalanced target value, if accuracy is lower we can do over sampling techniques to make target value is balanced and retrain the model with new train & test value for getting better accuracy.

- Prediction :-

After build the model with train & test value. Then we fit thedata to the model, see the prediction of the test data and model accuracy, and F1 score.

- Select Best Model :-

Here we need a best classification model, so select the best model based on F1 Score not only accuracy of the model. Select best model as random forest model, it have higher F1 score and accuracy compared to other models here we trained.

- Hyper-parameter :-

In these section we tune the best model that we use at the time off model building. In hyper parameter tuning, we have implemented Randomized Search CV for model tuning. From that we have chosen best parameters for model according to hyper parameter tuning.

- Model Testing :-

After hyper parameter tuning we put all the best parameter in our ML model. We test our data in this model & check F1 score and accuracy of the model is improved or not, here we concluded that score has been almost same, Accuracy is 94.3% and F1 score 0.90 for 0 and 0.96 for 1.

- Model Dump:-

The higher accuracy model is dump in pickle file (Random Forest Regressor).

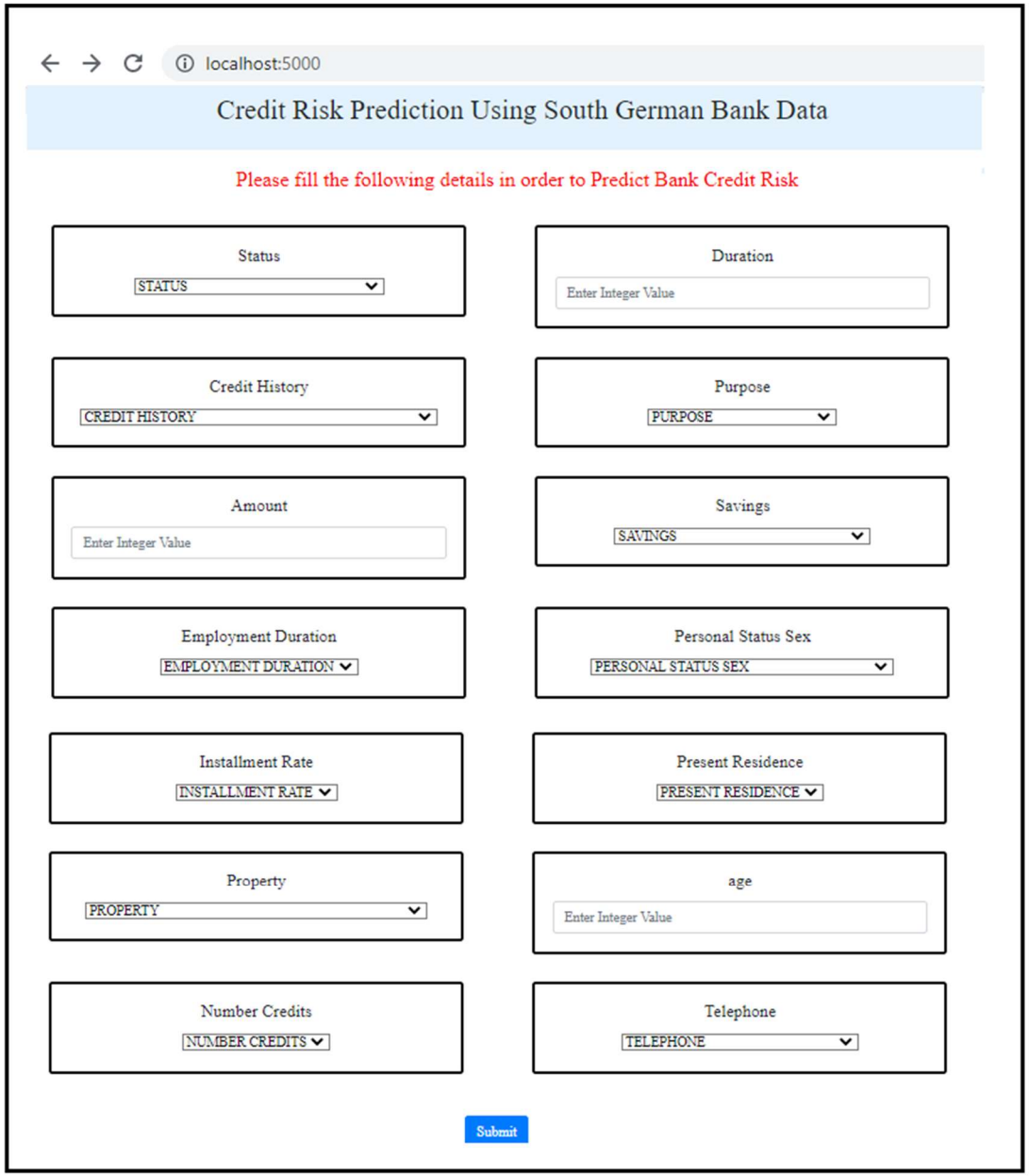
- Cloud Setup:-

In that we upload code on GitHub for deployment purpose. The HTML content to create the front page& backend we will be using the flask python framework.

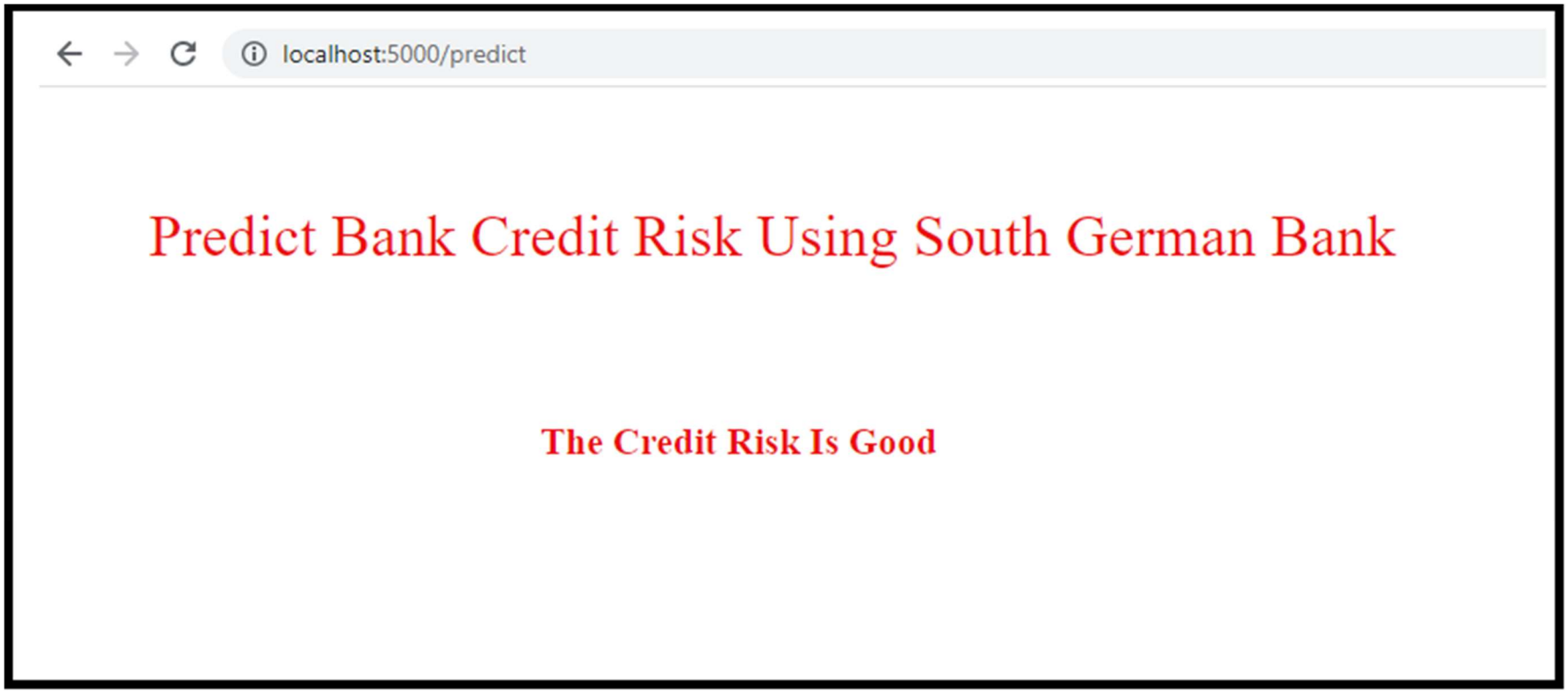
- User Interface:-

The user interface is divided into two pages, first one is homepage where user can enter the value forcalculating the credit risk & second one is result page where you can the result that whether the credit risk is good or bad

Homepage


localhost:5000
Credit Risk Prediction Using South German Bank Data
Please fill the following details in order to Predict Bank Credit Risk
Status
STATUS
Duration
Enter Integer Value
Credit History
CREDIT HISTORY
Purpose
PURPOSE
Amount
Enter Integer Value
Savings
SAVINGS
Employment Duration
EMPLOYMENT DURATION
Personal Status Sex
PERSONAL STATUS SEX
Installment Rate
INSTALLMENT RATE
Present Residence
PRESENT RESIDENCE
Property
PROPERTY
age
Enter Integer Value
Number Credits
NUMBER CREDITS
Telephone
TELEPHONE
Submit

Result Page


localhost:5000/predict
Predict Bank Credit Risk Using South German Bank
The Credit Risk Is Good

Q & A

Q1) What's the source of data?

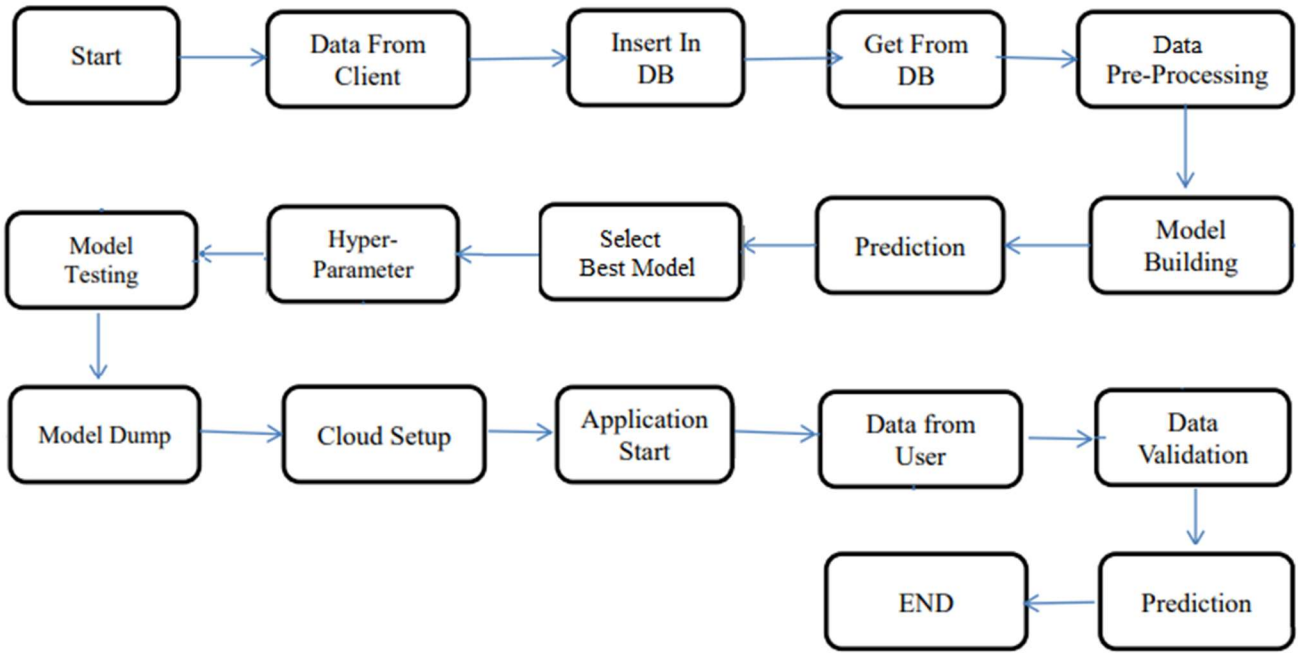
➡ The data for training is provided by the client .

Q 2) What was the type of data?

➡ The data type is numerical and catagorical.

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

➡



Q 4) How logs are managed?

⟹ We are using different logs as per the steps that we follow in, Data Insertion, Model Training log , prediction log etc.

Q 5) What techniques were you using for data pre-processing?

⟹ Converting the columns having ordinal values to Label Encoding.

Converting the columns having non-ordinal values to One Hot Encoding

Q 6) How training was done or what models were used?

⟹ Algorithms like Logistic Regression, Logistic Regression, Support Vector Machine, Random Forest Classifier, K- Nearest Neighbor and Naïve Bayes Classifier are used for training. Model selection based on higher F1 score.

Q 7) What are stage for deployment?

⟹
1. The 1st stage is to create your app.py file so that we can see our model.
2. The 2nd stage is to update your project on GitHub.
3. The 3rd stage in which you can deploy your model on any server such as GCP, AWS, Azure

THANK YOU