THYROID DISEASE DETECTION

Detailed Project Report

Dheerendra Kushwaha
Data Science Intern at Ineuron.ai

INTRODUCTION

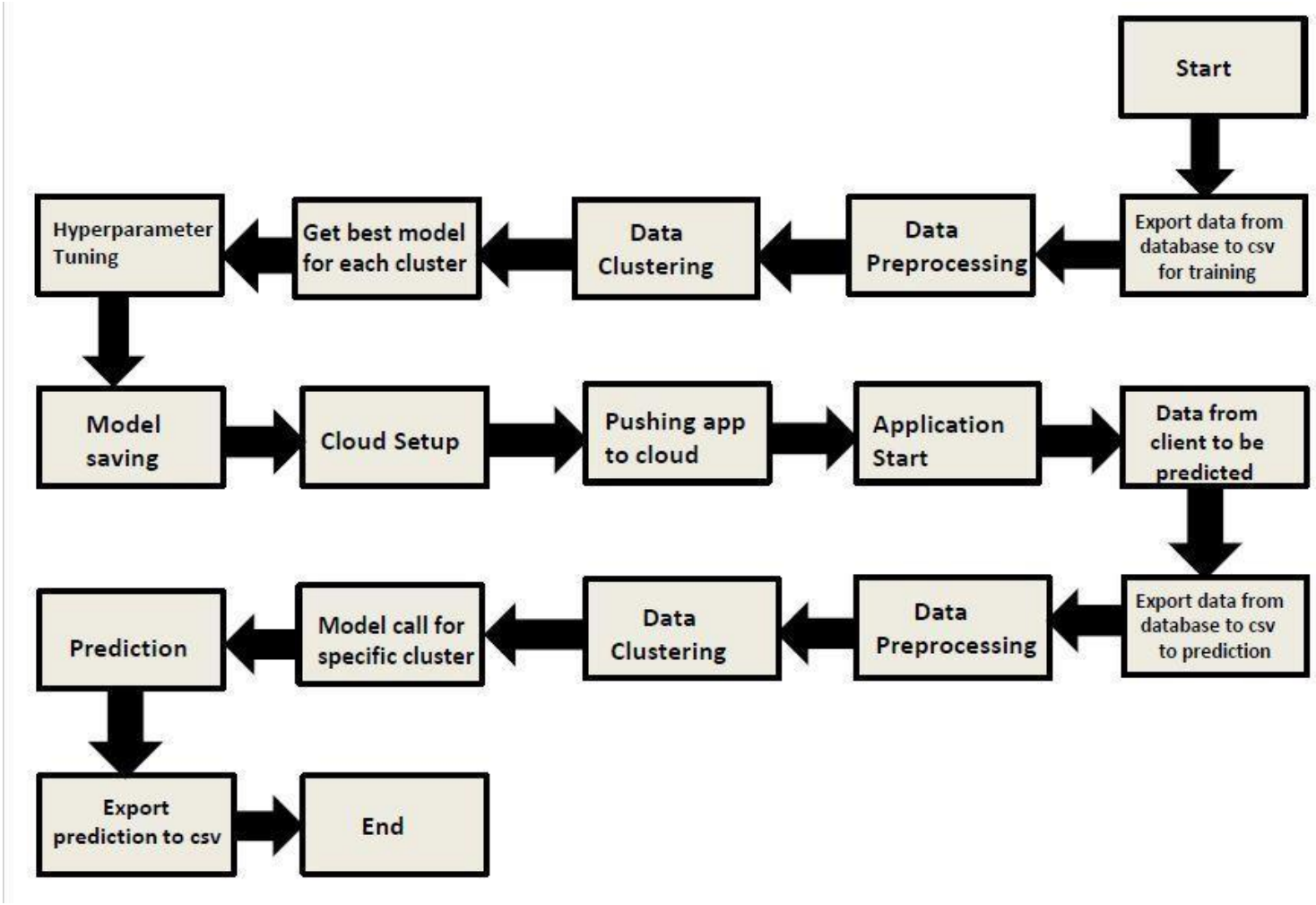
At least a person out of ten is suffered from thyroid disease in India. The disorder of thyroid disease primarily happens in the women having the age of 17–54. The extreme stage of thyroid results in cardiovascular complications, increase in blood pressure, maximizes the cholesterol level, depression and decreased fertility. The hormones, **total serum thyroxin (T4)** and **total serum triiodothyronine (T3)** are the two active thyroid hormones produced by the thyroid gland to control the metabolism of body. For the functioning of each cell and each tissue and organ in a right way, in overall energy yield and regulation and to generate proteins in the ordnance of body temperature, these hormones are necessary .

Hyperthyroidism and **Hypothyroidism** are the most two common diseases caused by irregular function of thyroid gland. Thyroid disorder can speed up or slow down the metabolism of the body. In the world of rising new technology and innovation, health care industry is advancing with the role of Artificial Intelligence. Machine learning algorithms can help to early detection of the disease and to improve the quality of the life. This study demonstrates the how different classification algorithms can forecasts the presence of the disease. Different classification algorithms such as Logistic regression, Random Forest, Decision Tree, Naïve Bayes, Support Vector Machine, XG Boost, KNN have been tested and compared to predict the better outcome of the model.

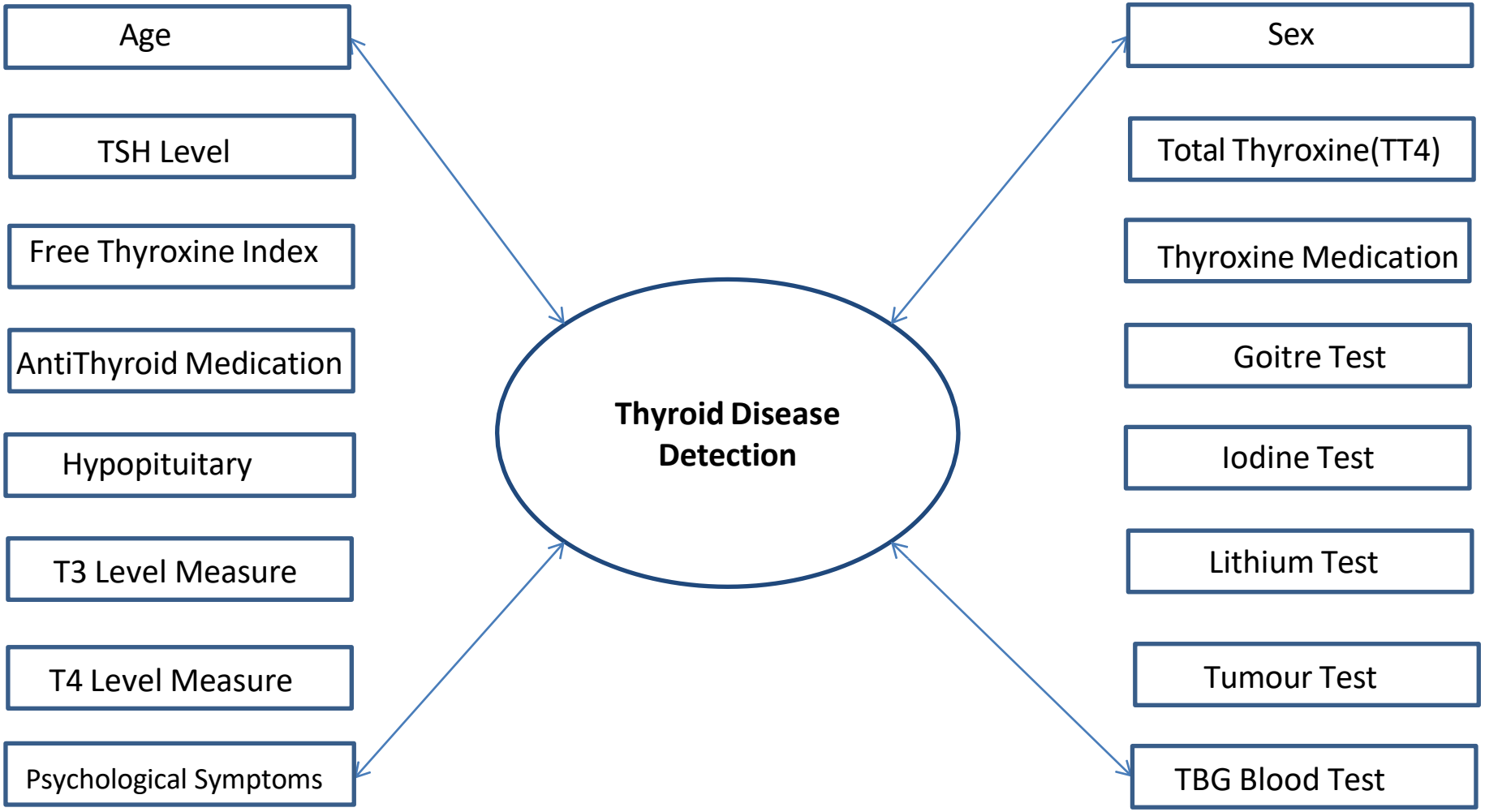
OBJECTIVE

The main goal of this project is to predict the risk of hyperthyroid and hypothyroid based on various factors of individuals. Thyroid disease is a common cause of medical diagnosis and prediction, with an on set that is difficult to fore cast in medical research. It will play a decisive role in order to early detection, accurate identification of the disease and helps the doctors to make proper decisions and better treatment.

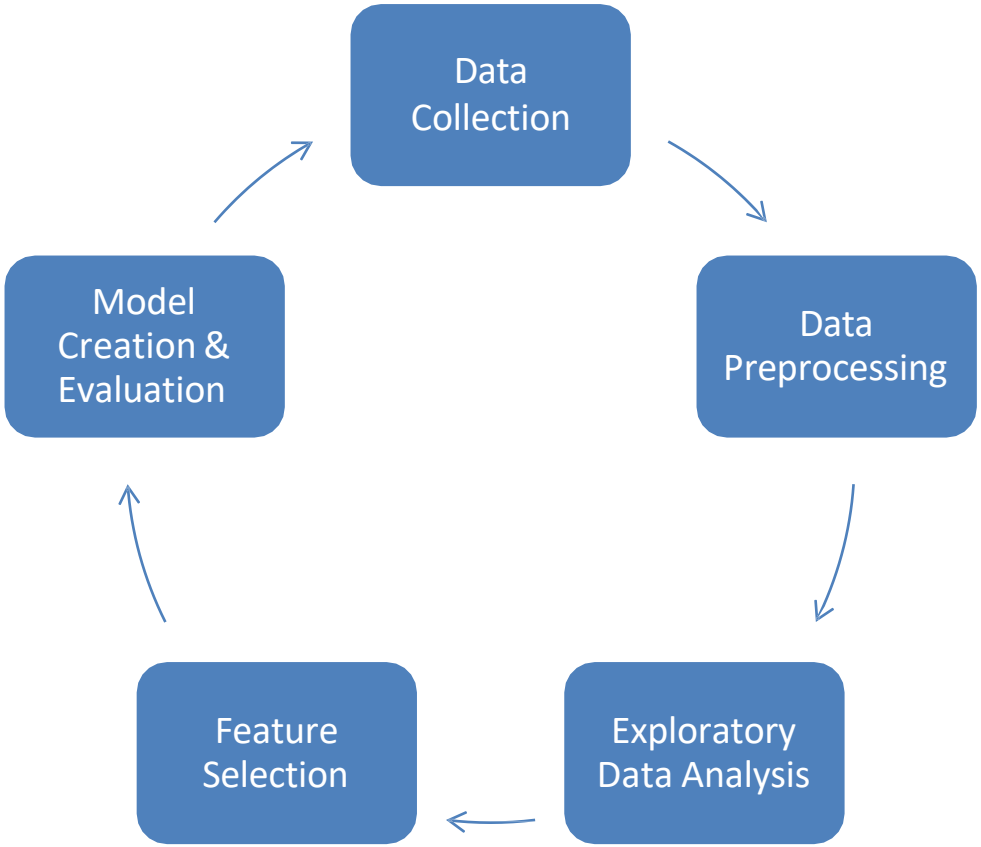
ARCHITECTURE



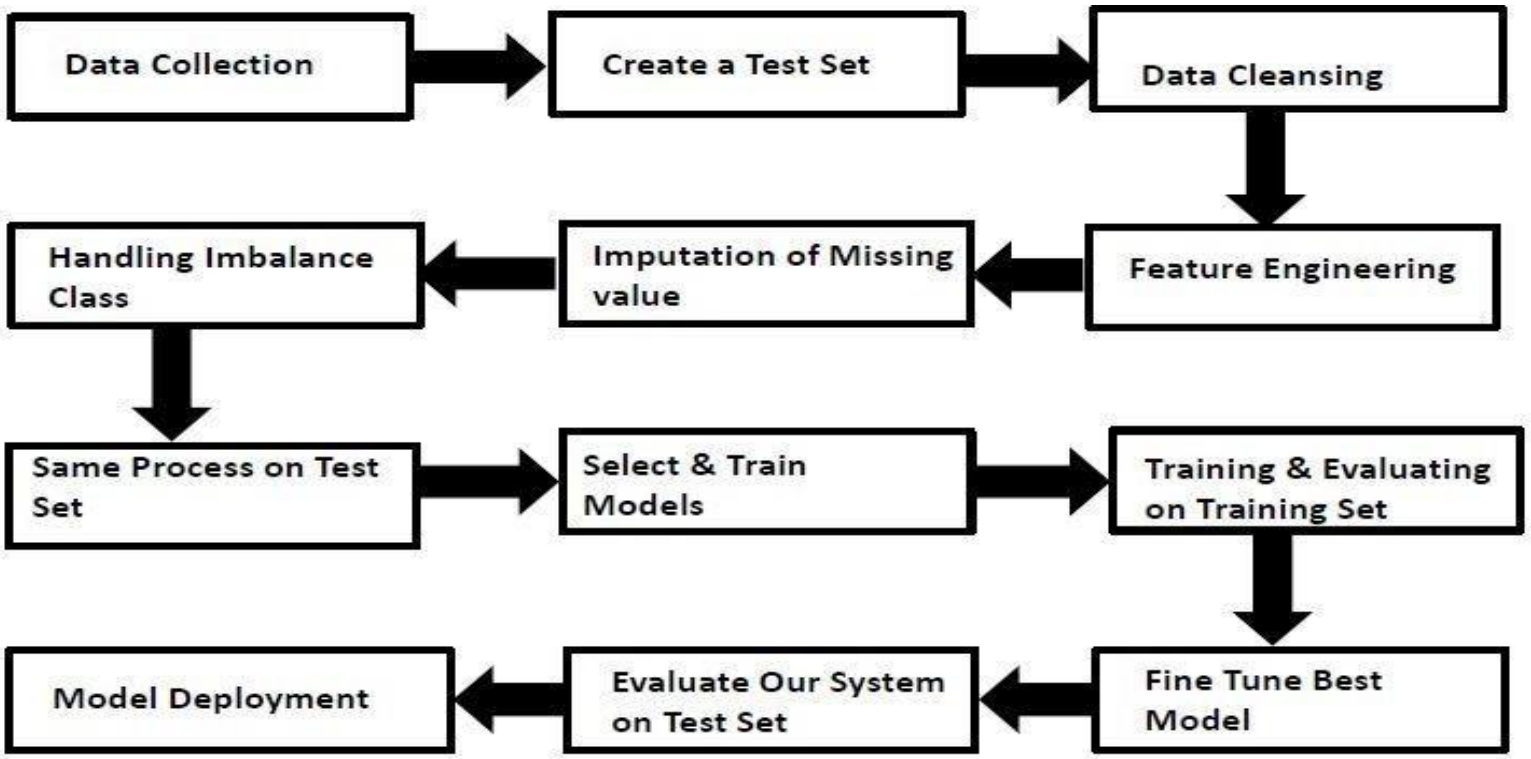
DATASET


Age
TSH Level
Free Thyroxine Index
AntiThyroid Medication
Hypopituitary
T3 Level Measure
T4 Level Measure
Psychological Symptoms
Thyroid Disease Detection
Sex
Total Thyroxine(TT4)
Thyroxine Medication
Goitre Test
Iodine Test
Lithium Test
Tumour Test
TBG Blood Test

Data Analysis Steps



MODEL TRAINING AND VALIDATION WORKFLOW



MODEL TRAINING AND VALIDATION WORKFLOW

Data Collection

- Thyroid Disease Data Set from UCI Machine Learning Repository
- For Data Set: https://archive.ics.uci.edu/ml/datasets/thyroid+disease

Data Pre-Processing

- Missing values handling by Simple imputation (Used KNN Imputer)
- Outliers' detection and removal by boxplot and percentile methods
- Categorical features handling by ordinal encoding and label encoding
- Feature scaling done by Standard Scalar method
- Imbalanced dataset handled by SMOTE -Over sampling
- Drop unnecessary columns

MODEL TRAINING AND VALIDATION WORKFLOW

Model Creation and Evaluation

- Various classification algorithms like Random Forest, XG Boost, KNN etc tested.
- Random Forest, XGBoost and KNN all were given better results. XG Boost was chosen for the final model training and testing.
- Hyper parameter tuning was performed.
- Model performance evaluated based on accuracy, confusion matrix, classification report.

XG Boost Classifier Model

INTRODUCTION

A decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.

The XGBoost Classifier is a supervised learning algorithm which we can use for regression and classification problems. It is among the most popular machine learning algorithms comes under boosting ensemble technique.

XGBoost Classifier being ensemble algorithm tends to give more accurate result. This is because it works on the principle i.e. number of weak estimators when combined forms strong estimator. Even if one or few decision tree are prone to noise, overall results would tend to be correct.

Reason to use XGBoost Classifier model:

- It has high execution speed.
- It gives better model performance.

MODEL PREDICTION RESULTS ON TEST DATASET

Classification Report

	precision	recall	f1-score	support
0.0	0.94	1.00	0.97	33
1.0	1.00	0.36	0.53	14
2.0	0.99	1.00	1.00	1068
3.0	1.00	1.00	1.00	1156
accuracy			1.00	2271
macro avg	0.98	0.84	0.87	2271
weighted avg	1.00	1.00	1.00	2271

Confusion Matrix

```
[[  33    0    0    0]
 [   2    5    7    0]
 [   0    0 1068    0]
 [   0    0    0 1156]]
```

DATABASE CONNECTION & DEPLOYMENT

Database Connection

- Cassandra Database used for this project.

```
Connected as upendra.kumar48762@gmail.com.
Connected to cndb at cassandra.ingress:9042.
[cqlsh 6.8.0 | DSE DB 4.0.0.6815 | CQL spec 3.4.5 | Native protocol v4]
Use HELP for help.
token@cqlsh> select * from db.Good_Raw_Data;

 age | class                    | fti | fti_measured | goitre | hypopituitary | i131_treatment | lithium | on_antithyroid_medication
| on_thyroxine | pregnant | psych | query_hyperthyroid | query_hypothyroid | query_on_thyroxine | referral_source | sex | sick | t3
| t3_measured | t4u   | t4u_measured | tbg | tbg_measured | thyroid_surgery | tsh    | tsh_measured | tt4 | tt4_measured | tumor
-----+--------------------------+-----+--------------+--------+---------------+----------------+---------+---------------------------
+--------------+----------+-------+--------------------+-------------------+--------------------+-----------------+-----+------+------
+-------------+------+--------------+-----+--------------+-----------------+-------+--------------+-----+--------------+-------
  23 |                'negative' | 242 |          't' |   'f' |           'f' |          'f' |     'f' |                          'f'
|          'f' |      'f' |   'f' |                't' |                 'f' |                'f' |          'other' | 'F' |  'f' |    4
|         't' | 0.84 |          't' | '?' |              'f' |             'f' | 0.015 |          't' | 204 |          't' |  'f'
  53 |                'negative' | 186 |          't' |   'f' |           'f' |          'f' |     'f' |                          'f'
|          't' |      'f' |   'f' |                'f' |                 'f' |                'f' |          'other' | 'F' |  'f' |  3.4
|         't' | 1.08 |          't' | '?' |              'f' |             'f' | 0.005 |          't' | 201 |          't' |  'f'
  91 |                'negative' | 132 |          't' |   'f' |           'f' |          'f' |     'f' |                          'f'
|          'f' |      'f' |   'f' |                'f' |                 'f' |                'f' |            'SVI' | 'F' |  'f' |  1.3
|         't' | 0.96 |          't' | '?' |              'f' |             'f' |  0.15 |          't' | 126 |          't' |  'f'
```

Model Deployment

- The final model is deployed on Heroku using Flask framework.

HEROKU

FREQUENTLY ASKED QUESTIONS

Q1) What is the source of data?

The data for training is obtained from famous machine learning repository.
UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/thyroid+disease

Q2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q3) What's the complete flow you followed in this Project?

Refer slide 7th, 8th and 9th for better understanding.

Q4) After the File validation what you do with incompatible file or files which didn't pass the validation?

Files like these are moved to the Achieve Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q5) How logs are managed?

We are using different logs as per the steps that we follow in training and prediction like model training log and prediction log etc. And then sub log are inside those folder.

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

- Removing unwanted attributes
- Visualizing relation of independent variables with each other and output variables
- Checking and changing Distribution of continuous values
- Removing outliers
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.

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

- First Data validation done on raw data and then good data insertion happen in DB.
- Then Data preprocessing done on final CSV file received from DB.
- We did clustering over the data to divide it on desired cluster based on elbow method.
- Various model such as Decision Tree, Random Forest and XGBoost models are trained on all clusters and based on performance, for each cluster different model is saved.

Q 8) How Prediction was done?

- The testing files are shared by the client .We Perform the same life cycle till the data is clustered .
- Then on the basis of cluster number model is loaded and perform prediction. In the end we get the accumulated data of predictions.

Q 9) What are the different stages of deployment?

- After model training and finalizing all models. We created required files for deployment.
- Finally deployed our model over various cloud platforms such as Heroku and AWS.

Q 10) How is the User Interface present for this project?

- For this project I have made two types of UI.
- First is for bulk prediction.
- Second is for one user input prediction.
- Both UI are very user friendly and easy to use.

THANK YOU