PROJECT REPORT

Improve Room Utilization

(using Room Occupancy Detector)

PROBLEM STATEMENT:

In the IT sector, the operation and management efficiency of room reservation has been improved significantly. Nowadays, rooms can be reserved for a specific time slot. But there are a couple of drawbacks for the operation. For instance, while reserving a room, length of reservation is also determined at that time, which is not flexible for further changes. Rooms can only be reserved based on no. of time slots. This causes a lower room utilization and operational efficiency.
In order to solve this problem, we would like to collect real-time information of room occupancy. With this data, the information of room occupancy can be provided in real time, and rooms that are not in used can be available for reservation immediately even though it is reserved under previous reservation.

In this project, we are using data that can be a factor to determine room occupancy and using several classification methods to develop an effective model for room occupancy determination.

DATA SOURCE:

http://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+

DATA ATTRIBUTES :

- Date Time (yyyy-mm-dd hh:mm:ss)
- Temperature (in Celsius)
- Relative Humidity (in %)
- Light (in Lux)
- CO2 (in ppm)
- Humidity Ratio (in kgwater-vapor/kg-air)
- Occupancy (0 if not occupied and 1 if occupied)

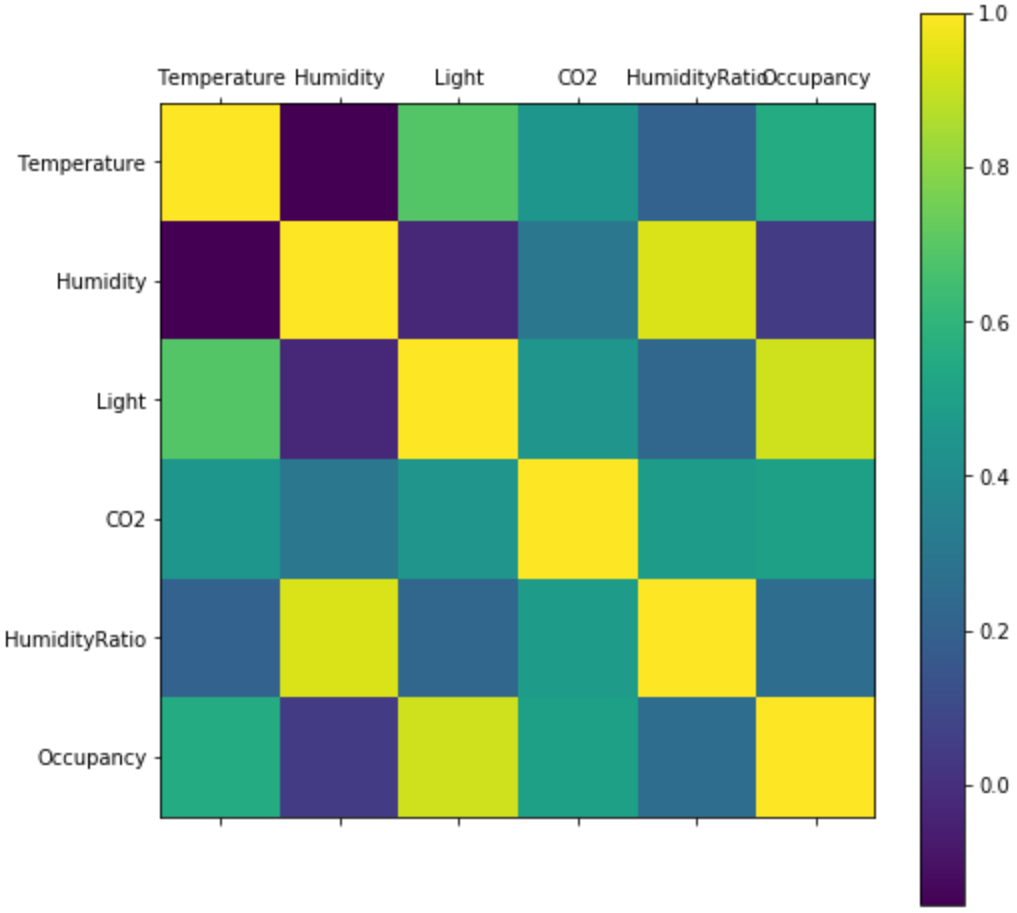
DATA PRE-PROCESSING :

- There are total 6 attributes to determine room occupancy and the dependent variable (binary) that indicates whether the room is occupied or not.

- We did not select the variable ‘date time’ as a factor to build the model assuming that it is only an index of the data, and does not have any dependent relationship with the final result.
- After data visualization, we found that the variable Humidity Ratio and Relative Humidity are linearly dependent. So, we selected only Relative Humidity as a factor to build the model.

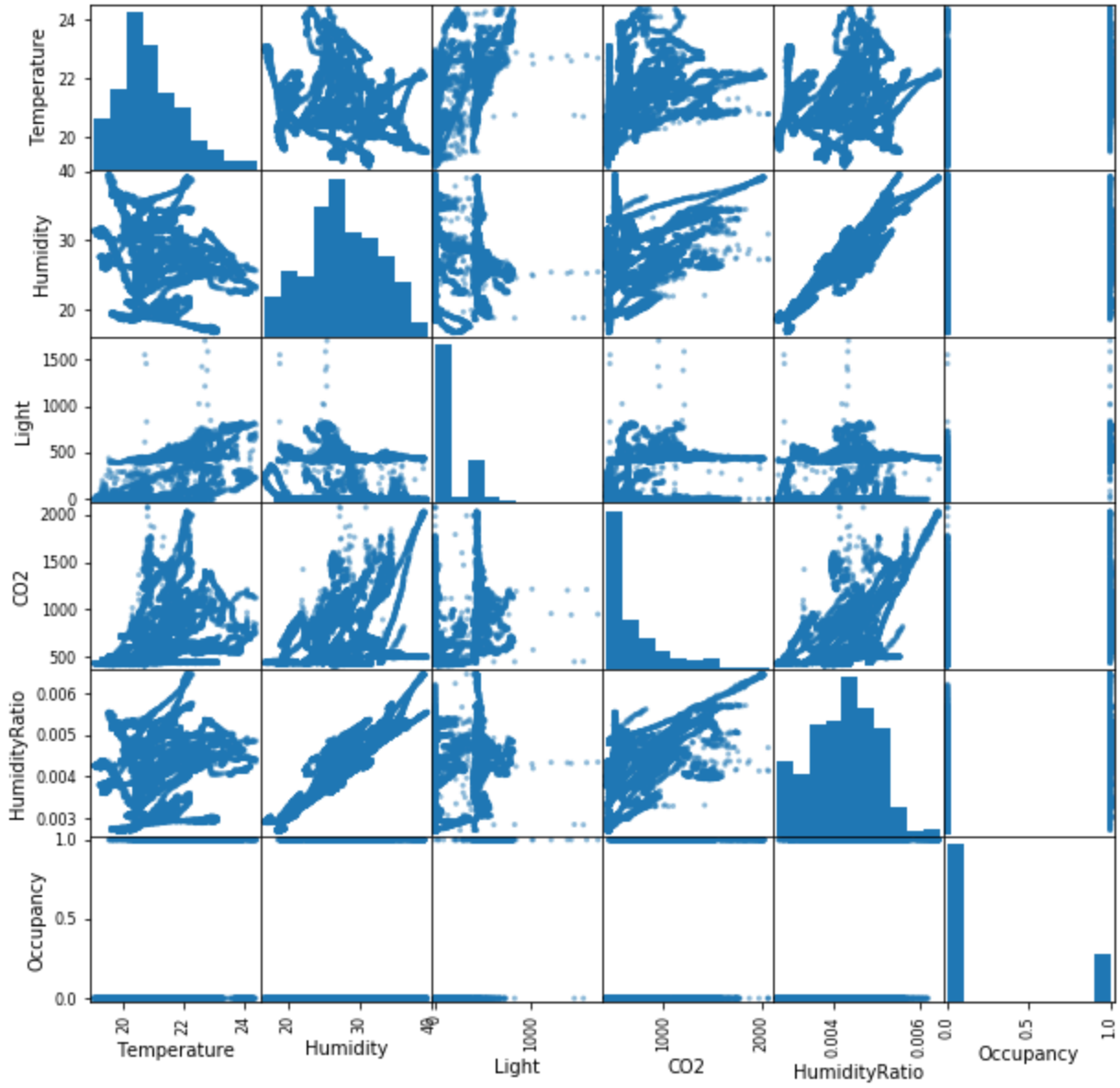
DATA VISUALISATION :

- Correlation Matrix :



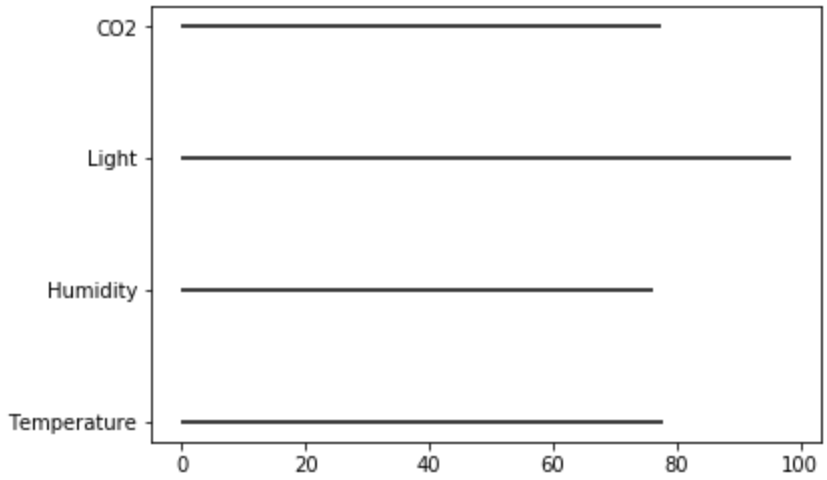
Features ‘humidityRatio’ and ‘humidity’ has a high correlation.
Also, the Label of Occupancy depends a great deal on the value of feature ‘Light’.

- Pair-wise Relationships :



Clearly, features 'HumidityRatio' and 'Humidity' are fully linear dependent. Thus, only one of the features is required.

- Accuracy Graph of Simple Logistic Regression Model on each environment measure in isolation



Only “Light” feature is required in order to achieve 99% accuracy. It is very likely that the office rooms in which the environmental variables were recorded had a light sensor that turned internal lights on when the room was occupied and off otherwise.

OVERVIEW:

- Features HumidityRatio and Humidity are fully linear dependent.Therefore, we dropped the column HumidityRatio.
- Using light feature itself gives 99% accuracy and to make the problem more challenging, we removed the Light feature. Thus, dropping the feature, to further generalize the model.

METHODOLOGY:

Generated models using Pickle:

Pickle: It is one of the python libraries which is used to perform tasks like:

- Pickling
- Unpickling

In this project we have used Pickling.

Pickling is the process of converting any Python object into a stream of bytes by following the hierarchy of the object we are trying to convert.

Work Flow:

- Import the Python serialization package pickle.

- Dump the trained model classifier with pickle.
- Open the file in write mode to save as pkl file.
- Close the pickle instances.
- Load the saved model pickle.
    - Open the model_pkl_filename in the read mode.
    - Use the pickle load method to load the saved model.
    - Use the loaded model to predict the dependent variable(Occupancy).

In this project, we have implemented four models :

A. Logistic Regression
B. Naive Bayes
C. Decision Trees
D. Random Forest

A) Logistic Regression: Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.

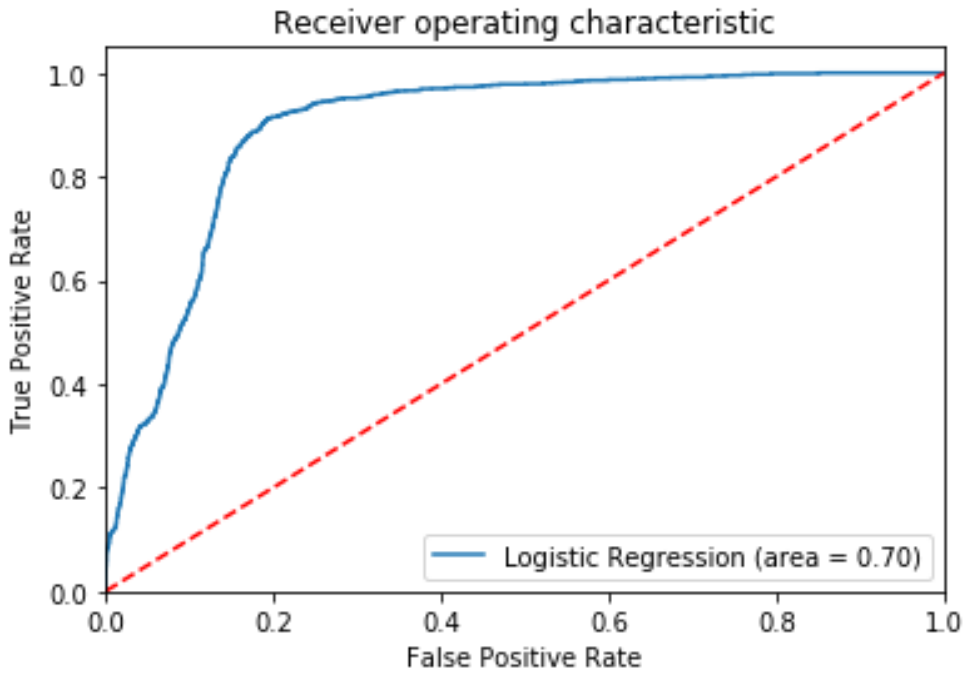
- Accuracy : 0.8150

- Confusion Matrix :
  [4328 378]
  [763 699]
  This result tells us that we have *4328+699* correct predictions and *763+378* incorrect predictions.

- Report :
    - *Precision:* It is the accuracy of positive predictions.
      Precision = TP/(TP+FP)
    - *Recall:* It is sensitivity or true positive rate i.e. Fraction of positives that were correctly identified
      Recall = TP/(TP+FN)
    - *F1-score (F-Score or F-Measure):* It is helpful for comparing two classifiers. It is calculated by finding the harmonic mean of Precision and Recall.
      F1 = 2*(Precision*Recall)/(Precision + Recall)

Occupancy	Precision	Recall	F1-score	Support
0.0	0.85	0.92	0.88	4706
1.0	0.65	0.48	0.55	1462
Avg/Total	0.80	0.82	0.80	6168

- ROC curve :



B) Naive Bayes: A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

- Accuracy: 0.8280

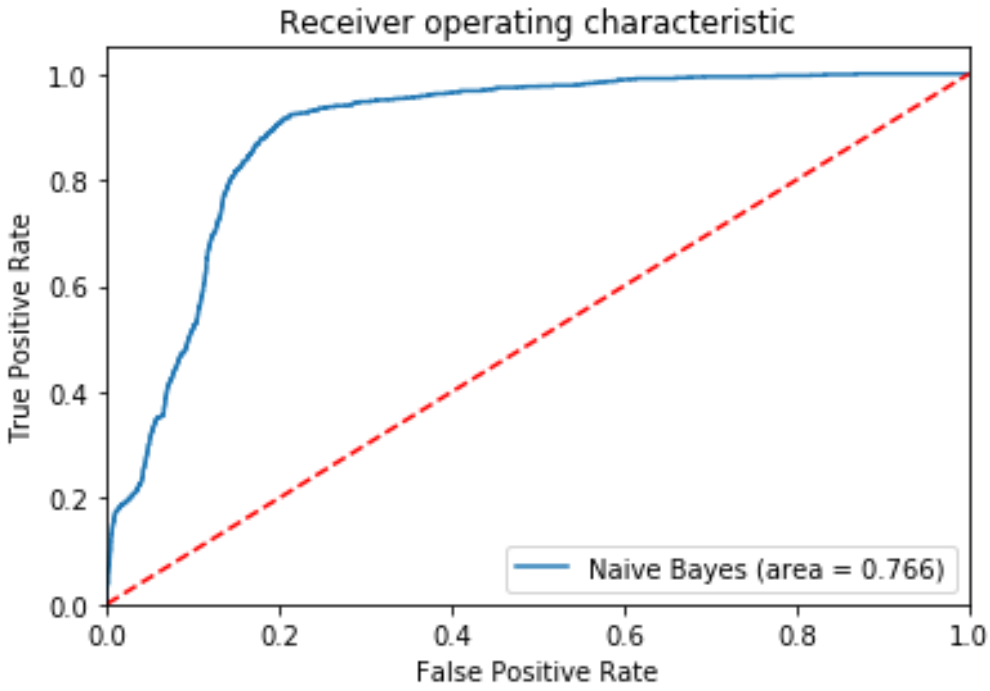
- Confusion Matrix:
  [4160 546]
  [515 947]

This result tells us that we have *4160+947* correct predictions and *515+546* incorrect predictions.

- Report:

Occupancy	Precision	Recall	f1-score	Support
0.0	0.89	0.99	0.89	4706
1.0	0.63	0.65	0.64	1462
Avg/Total	0.83	0.83	0.83	6168

- ROC curve:



Our Ideology behind decision tree:

Bayes Theorem states: P(A/B)=(P(B/A)*P(A))/P(B)

Using the Bayes Theorem, it will find the probability of the occupancy(A) given the various features(B) with a very strong assumption that the presence of one feature doesn't affect the presence of another.

Calculating, the P(B/A) (i.e. probability of feature 'CO2'(B) given 'label of Occupancy(A)(i.e. 0 or 1)) doesn't give us complete information. Because with some tweaks in temperature, light and humidity level, there is an equal probability for both the labels of Occupancy(i.e. 0/1) for the same level of 'CO2'. Therefore, calculating P(A/B) (i.e. probability of label(A) on test data given feature ''CO2'(B)) also doesn't give us complete information since it is calculated using P(B/A).

While the decision tree uses a completely different approach.
In the construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.
It provides a clear indication of which fields are most important for prediction or classification.

C) Decision Trees:

Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is like a tree structure, where each internal node denotes a condition on a feature, each branch represents an outcome of the condition, and each leaf node holds a binary decision.

- Accuracy: 0.9797

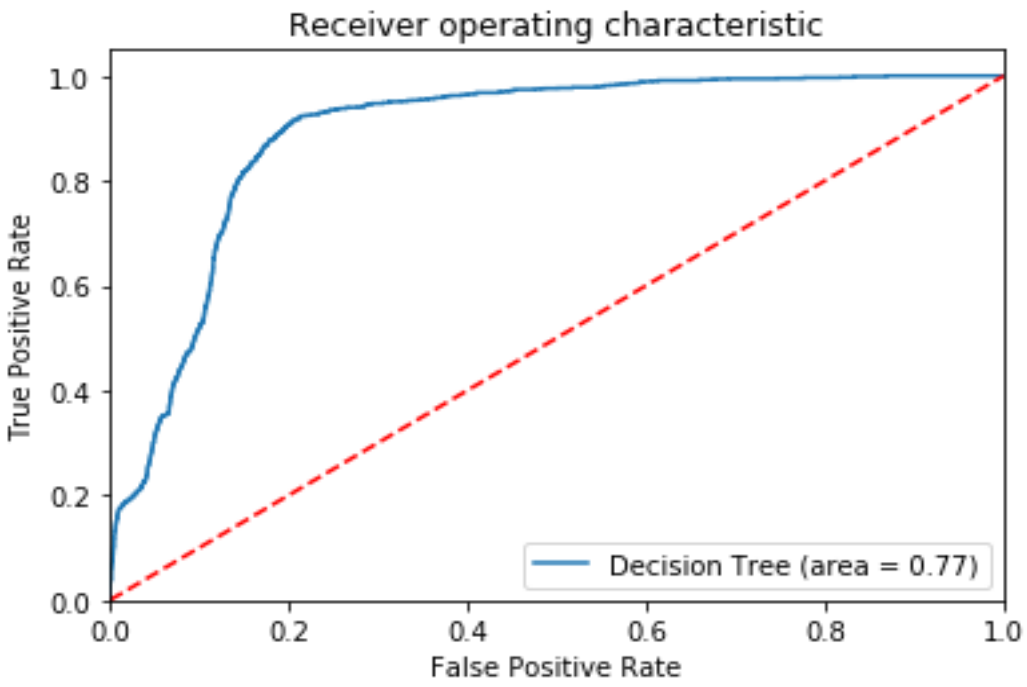
- Confusion Matrix:
  [4654 52]
  [73 1389]

This result tells us that we have *4654+1389* correct predictions and *73+52* incorrect predictions.

- Report:

Occupancy	Precision	Recall	f1-score	Support
0.0	0.98	0.99	0.99	4706
1.0	0.96	0.95	0.96	1462
Avg/Total	0.98	0.98	0.98	6168

- ROC curve:



4) Random Forests:

Random forest is based on tree-based method. The tree based method will start with the entire space and recursively divide it into smaller regions based on the value of input variables. It is basically a collection of decision trees. At the end, our model gives out 0 or 1 depending on the occupancy status.

An important parameter for random forest is the number of trees to grow for the model. If we grow too less number of trees, the result is of high randomness, which does not represent the model's true predictability very well. If we grow too many trees, it will cost a lot of time and computational resource. So we repeat the application with different number of trees, starting from 100 to 1500 to find out the appropriate number of trees. As show in below plot, the average AUC converged to 0.97, we believe this reflect this model's true predictability, therefore 1500 is the desired level of tree numbers to grow.

- Accuracy: 0.9833

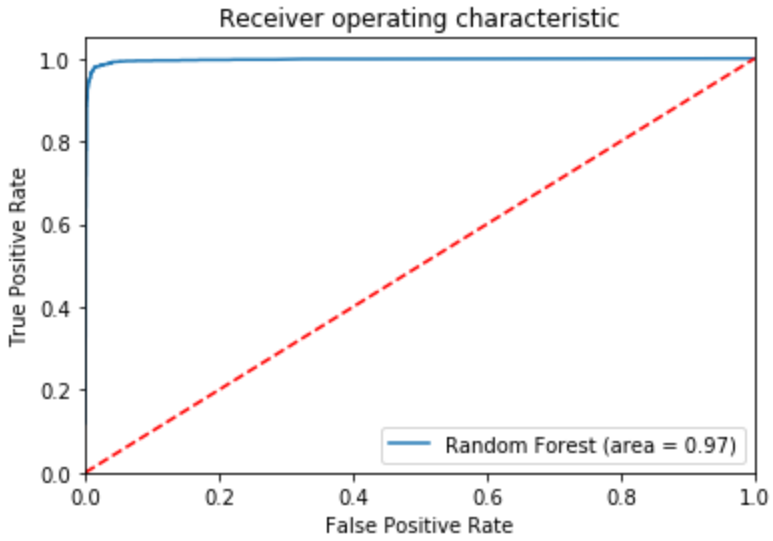
- Confusion Matrix:
  [4671 35]
  [68 1394]

This result tells us that we have *4671+1394* correct predictions and *68+35* incorrect predictions.

- Report:

Occupancy	Precision	Recall	f1-score	Support
0.0	0.99	0.99	0.99	4706
1.0	0.98	0.95	0.96	1462
Avg/Total	0.98	0.98	0.98	6168

- ROC curve:



CONCLUSION:

In this project, we implemented Logistic Regression, Naive Bayes, Decision Trees and Random Forest to develop an effective model for room occupancy determination.
Out of all these, Random Forest gave us the best performance with an AUC of 97% using 3 features (i.e. Temperature, humidity and CO2).

For future development, factors like Season, Place, Weather report need to be taken into consideration as we noticed that time span of the data used is only of one week and for a particular room. For a large scale data which includes location of different parts of the world, these factors will play a vital role.

By- Team XYZ

- Tanu Bordia (IMT2016002)
- Apoorva Choudhary (IMT2016028)

- Mudita Baid (IMT2016038)