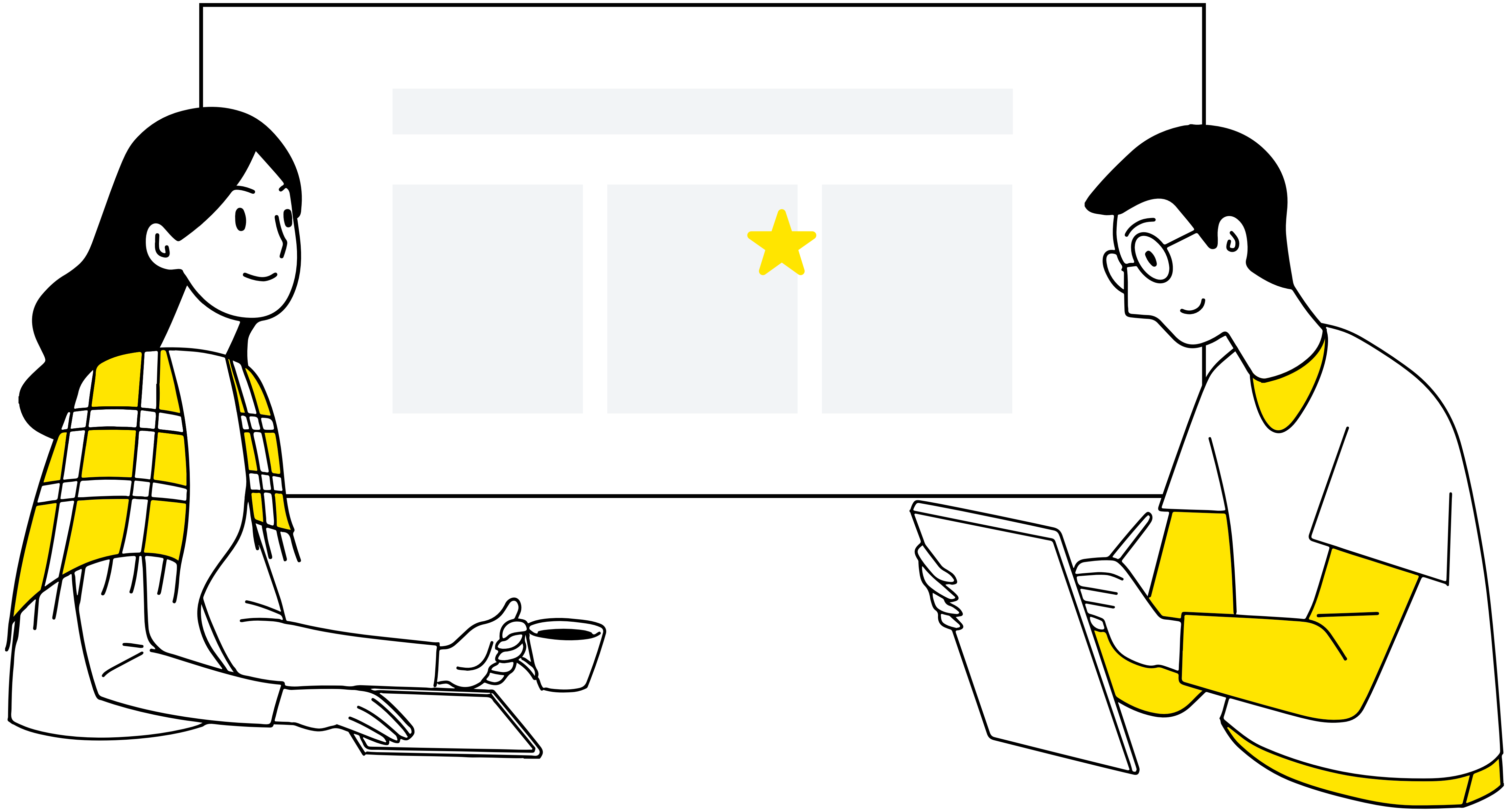
Air Quality Index Predictions



Objectives

Development of Air Quality Index Predictive Model for people living in polluted cities and want to measure the quality of the air to decide whether to go out or not. The model will determine the Quality of Air within the range from Good to Severe.

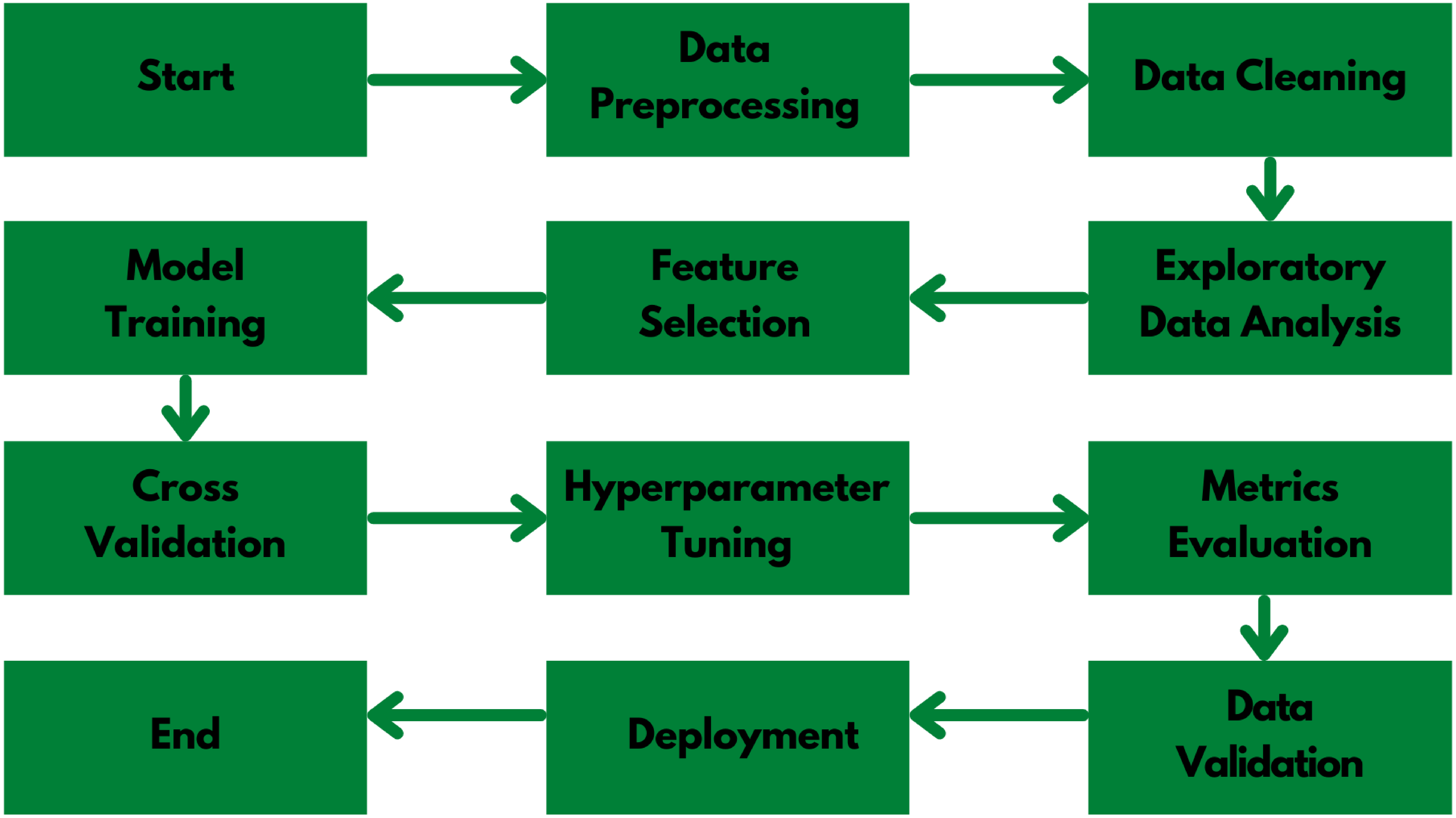
Benefits

- Predict the Quality of Air based on the given parameters
- Provides insights about the Quality of air.
- Provides the relation between the parameters.
- Provides AQI values that help to alter the area.

Data Sharing Agreement

- Sample File Name: **city_day.csv**
- Cleaned Data File Name: **final_data.csv**
- Number of Columns: **6**
- Number of Rows: **22618**
- Columns Name: **PM2.5 (Particulate Matter), NO2 (Nitrogen Di-oxide), CO (Carbon Mono-oxide), SO2 (Sulphur Di-oxide), O3 (Ozone), AQI (Air Quality Index)**
- Columns Datatype: **float64**

Architecture



Start

Data Preprocessing

Data Cleaning

Exploratory Data Analysis

Feature Selection

Model Training

Cross Validation

Hyperparameter Tuning

Metrics Evaluation

Data Validation

Deployment

End

Step 1: Data Preprocessing and EDA

- Raw data contains 16 columns: City, Date, PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3, Benzene, Toluene, Xylene, AQI, AQI_Bucket
- We used **DataPrep** module to perform EDA on the data where we find our dataset contains 18.7% of missing data where features like Xylene, Toluene, Benzene contains 19%, 27% and 61% missing values respectively. Since most of the cities don't take these parameters while calculating AQI, so we dropped them.
- We only selected the important parameters and dropped the unnecessary columns.
- We removed Date and City since it is not dependent to AQI.
- We also removed Nx and No since we are considering NO2 already.
- AQI_Bucket can be calculated using AQI values and since it was categorical data, we removed AQI_Bucket too

Step 2: Linear and Lasso Regression

- We first tried to create a base Linear regression model to identify the relation with different parameters.
- With Linear regression, we found, PM2.5 is highly correlated with AQI.
- Every parameter we took was sparse and therefore haven't shown any kind of relation with other parameters.
- Initially we got 83.97% R^2 for training and 85.6% R^2 for testing dataset, which shows overfitting. But after visualization, it was clear that Linear regression is not a right model.
- After Cross-Validation, we got the high MSE.
- Even after implementing Lasso Regression, we got the similar result. Thus we moved on to another model.

Step 3: Decision Tree Regressor

- After Linear Regression, we tried to implement the decision trees which initially gave us 100% R2 score on training data whereas 78% R2 score on the testing data. This clearly shows, our model has overfitted very badly.
- To reduce the overfitting problem, we used the 5-fold cross validation which reduced the problem somehow but still the model was dumb.
- To further solve the problem, we applied the hyper parameter tuning. We used GridSearchCV which took around 17 minutes but the results were pretty good.
- After Hyperparameter tuning, the R2 score was 77%. Even after trying out different parameters, we were not able to achieve more than 80% of the R2 score.

Step 4: XGBoost Algorithm

- Next, we tried the XGBoost algorithm to improve the accuracy of the model.
- Before tuning, XGBoost gave us 97% R2 score on train data and 87% R2 score on test data.
- After trying multiple hyper parameters, we got the highest R2 score as 94% for train and 88% for test data.
- XGBoost Algorithm was more generalized and real-world model.
- When we tested XGBoost algorithm on the unseen data, we encountered that our model was overfitting for high AQI values. Keeping this in mind, we moved to the next model.

Step 5: Random Forest Classifier

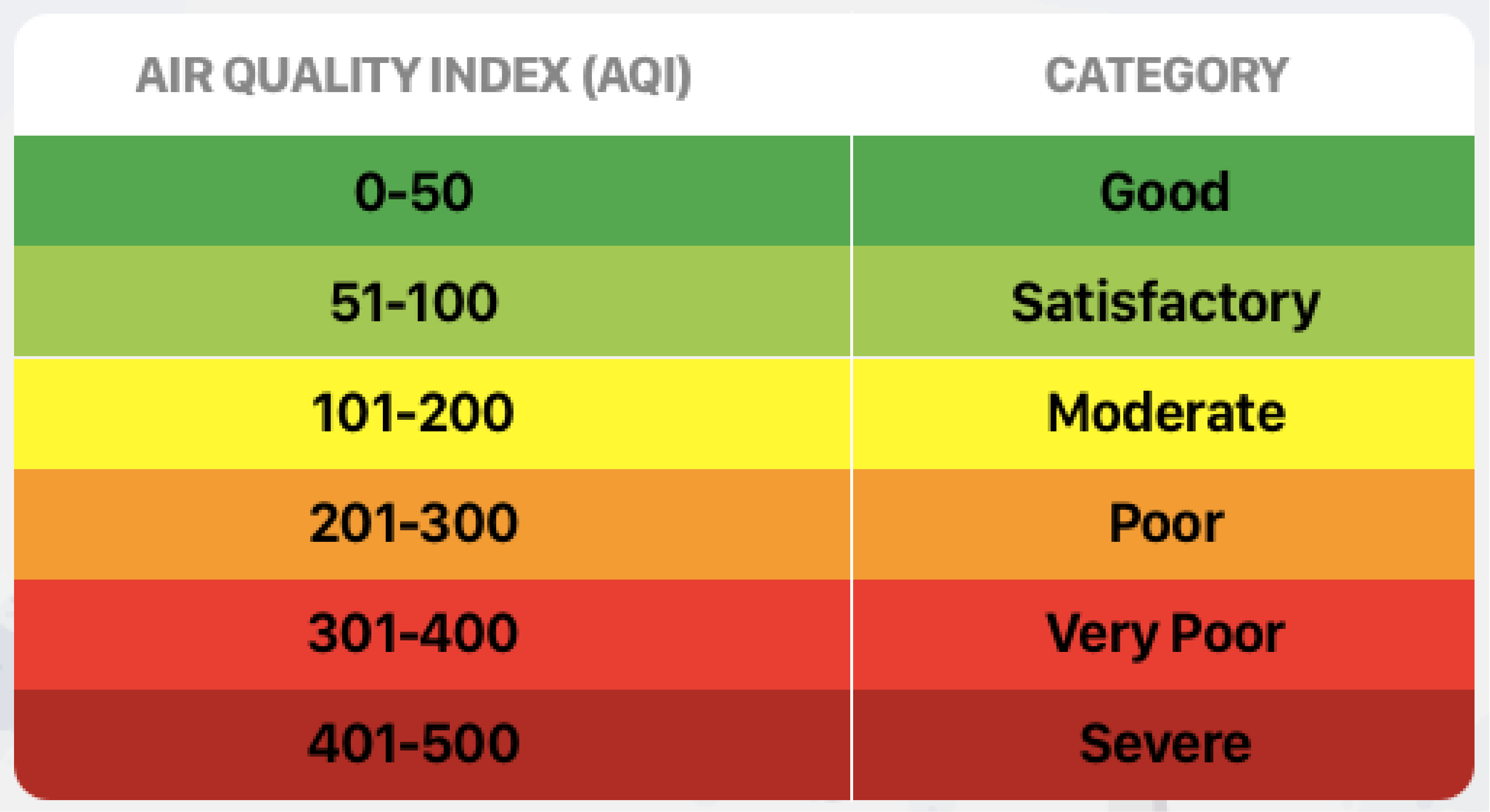
- Random Forest Classifier initially overfitted the model by 98.5% on train and 90% on test data.
- After Hyper parameter tuning, we got the improved R2 score of 91 on train and 89 on test data. This is the best generalized model we achieved so far.
- We then converted all the AQI value greater than 500 to be equal to 500.
- After normalizing the AQI values, we then again applied Random Forest.
- Although Random Forest don't get affected by outliers but after normalizing AQI, we got much better model which gave us 92% accuracy on training data and 91% of the accuracy on the testing data.
- Random Forest when tested on the unseen data, we got pretty near or quite correct values.

Step 6: Artificial Neural Network

- ANN when applied to the dataset, gave us 88% of the accuracy.
- ANN model was generalized and gave the less MSE too, but when applied on unseen data, it was somewhat overfitting.
- We then replaced the AQI values with the AQI_Bucket column which contains the categorical values.
- After performing one hot encoding, we converted AQI_Bucket values into numerical data.
- ANN when applied to the data with AQI_Bucket gave us the similar result with low accuracy.
- After going through every model and testing them all with unseen data, we finalized Random Forest model since it was more generalized and gave pretty good accuracy.

Prediction and Deployment

- We made the web app with the help of Streamlit as a frontend tool and used the RF model and deployed our app.
- The model used to predict the AQI values and based on the AQI ranges, we decide the Quality of the Air.
- We used the below standard AQI LookUp table to decide the category.



AIR QUALITY INDEX (AQI)	CATEGORY
0-50	Good
51-100	Satisfactory
101-200	Moderate
201-300	Poor
301-400	Very Poor
401-500	Severe

Frequent Q&A

Q) What is the source of the data?

- Data was collected from Kaggle, but city specific data can be collected from Central Board of Pollution website.

Q) What is the complete flow of your project?

- Refer to slide no 4 for better understanding.

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

- In data pre processing, we analyzed the data, found the important features, and based on the domain knowledge, we eliminated the unnecessary columns. We also tried to fill Missing Values with mean, median and mode but still the data have the same correlations. Thus removing the columns with high NaN values was the better option for us.

Frequent Q&A

Q) How did you choose the model?

- After implementing hyper parameter tuning, we were able to do model selection based on the metrics and how it was performing on unseen data. The final model we chose was Random Forest Classifier.

Q) How did you calculated the AQI?

- Central Board of Pollution has already made the categories of different AQI based on the AQI values. Kindly refer to slide 11 to see the table. For example, if the model predicted AQI value as 153, the Air Quality will be shown as **"Moderate".** Thus, just by predicting AQI values, our deployed model can match the values with categories and show the output.

Frequent Q&A

Q) What are the different stages of deployment?

- When the model was ready, we deployed the model using Streamlit on Heroku and performed some test.
- We added the common values or ranges of each parameter to better help the user to select the value.
- We then created the Explore page with the help of our EDA notebook where we embedded different types of graphs for each parameters.
- We then added the animations and improved the user interface.
- Once everything was finalized, we deploy it in production.