TEST PLAN FOR AIR POLLUTION FORECASTING IN NCR

Changelog

Version	Change Date	By	Description
001	01.11.2023	Ankita Kushwaha	Added some miscellaneous activities
002	03.11.2023	Avi Chaudhary, Avika Tyagi	Improved Stability of the website while launching

1 Introduction

Our system leverages historical air quality data, meteorological information, geographical factors, and advanced machine learning models to make predictions about future pollution levels by calculating AQI. It also offers a user-friendly interface for easy access to real-time air quality information. With this project, we endeavor to create a tool that not only aids in understanding air pollution patterns but also assists in taking preventive measures, thereby contributing to a cleaner, healthier environment.
This test plan report outlines our comprehensive approach to testing and validating the performance of our air pollution forecasting system. By adhering to rigorous testing procedures, we strive to ensure the accuracy, reliability, and usability of our solution, with the goal of enhancing public well-being and environmental sustainability.

1.1 Scope

1.1.1 In Scope

Functional Requirements:

1. The system must be able to retrieve and process air quality data from reliable sources.
2. The system should provide air pollution forecasts for a user-specified location and time frame.
3. The predictions should include various air pollutants such as PM2.5, PM10, SO2, CO, NO2, and O3.
4. The system should update the forecasts at regular intervals.

Non-Functional Requirements:

1. The system must achieve a prediction accuracy rate of at least 80% for major air pollutants.
2. The user interface should be intuitive and user-friendly to accommodate users of all technical backgrounds.

1.1.2 Out of Scope

The air pollution forecasting system does not cover forecasting air quality for indoor environments or specific indoor pollutants. It is focused solely on forecasting outdoor air pollution levels.

1.2 Quality Objective

- Ensure the Application Under Test conforms to functional and non-functional requirements.
- Ensure that the data set on which the model is trained contains all the necessary gases.
- Validate that the dataset used for training the prediction model contains all the necessary images and relevant environmental data, ensuring the accuracy and comprehensiveness of the model.
- Verify that the forecasts produced by the system include the predicted levels of various air pollutants and are accompanied by clear and informative data.
- Bugs/issues are identified and fixed before going live.

1.3 Roles and Responsibilities

- **Test Manager:** Prof. Abhishek Goyal
  Managed all test processes, including test plans, resources, costs, timescales, test deliverables and traceability.

- **Configuration Manager**: Prof. Akanksha

- **Developers:** Avi Chaudhary
  Developed the model and trained it.

- **Installation Team**: Avi Chaudhary, Avika Tyagi
  Responsible for smooth execution of the program.

- **User Interface (UI) Team:** Avika Tyagi, Ankita Kushwaha
  Responsible for ensuring a visually engaging website that enhances user experience and provides seamless access to air pollution forecasting.

- **QA Analyst:** Ankita Kushwaha, Avi Chaudhary, Avika Tyagi
  The Quality Assurance (QA) Analyst conducted testing on software, websites, and other technical products to identify and resolve bugs, defects, and other potential issues.

2 Test Methodology

2.1 Overview

Our test methodology for pollution forecasting combines manual testing and black box testing to rigorously assess the accuracy and reliability of our ML models and web development interface. Manual testing involves human-driven scenarios, while black box testing evaluates system functionality without detailed knowledge of internal structures. This comprehensive approach ensures our system's robustness and accuracy, vital for delivering reliable pollution forecasts.

2.2 Test Levels

For the pollution forecasting system, we can define the following test levels:

1. **Unit Testing:** This level focuses on validating individual components of the system, such as data preprocessing methods, ML algorithms, and web service endpoints.

2. **Integration Testing:** Integration testing checks the interactions between different system components and their compatibility, ensuring that data flows seamlessly between them.

3. **Functional Testing:** Functional testing verifies whether the system meets its functional requirements,

including web interface functionality and ML model accuracy.

4. **Usability Testing:** Usability testing assesses the user-friendliness of the web interface, evaluating navigation, user interactions, and overall user experience.

These test levels provide a comprehensive approach to validate and enhance the pollution forecasting system's reliability and functionality.

2.3 Test Completeness

- 100% test coverage
- All Manual Test cases executed.
- All open bugs are fixed or will be fixed in the next release.
- AQI is correctly predicted.
- Data ingestion and processing should be optimized for efficient handling of data in air pollution forecasting.

3 Test Deliverables

- **Test Case**

Case id	Test case Name	Description	Input data						Expected Output	Actual output	Status
			SO2i	NOi	O3i	PM25i	PM10i	COi			
1	Predict Good Air Quality	The model must be able to predict and label air quality as "Good" when the AQI falls within the specified range f	29	16	9	10	20	39	GOOD	GOOD	PASS
2	Predict Moderate Air Quality	The model must be able to predict and label air quality as "Moderate" when the AQI falls within the	29	10	84	20	50	39	MODERATE	MODERATE	PASS
3	Predict Poor Air Quality	The model must be able to predict and label air quality as "Poor" when the AQI falls within the spec	59	40	84	70	50	99	POOR	POOR	PASS
4	Predict unhealthy Air Quality	The model must be able to predict and label air quality as"unhealthy" when the AQI falls within the	149	120	184	220	100	100	UNHEALTHY	UNHEALTHY	PASS
5	Predict Very unhealthy Air Quali	The model must be able to predict and label air quality as"Very unhealthy" when the AQI falls withi	293	326	239	310	220	220	VERY UNHEALTHY	VERY UNHEALT	PASS
6	Out of the Range	Input or data that falls outside the expected or acceptable range.	500	0	0	-240	0	-4E+16	GOOD	ERROR	FAIL
7	Out of the Range	input or data that falls outside the expected or acceptable range.	500	0	0	0	0	1E+43	HAZARDOUS	ERROR	FAIL

- **Test case Output Images**

Test case 1:

```
1 DT2.predict([[50,30,30,50,100,89]])
```

```
array(['Poor'], dtype=object)
```

Test case 2:

```
1 DT2.predict([[29,10,14,50,30,19]])
```

```
array(['Good'], dtype=object)
```

Test case 3:

```
1 DT2.predict([[39,120,84,80,40,69]])
```

```
array(['Moderate'], dtype=object)
```

Test case 4:

```
1 DT2.predict([[409,120,184,220,100,209]])
```

```
array(['Unhealthy'], dtype=object)
```

Boundary value analysis:

Test case 5:

```
1 DT2.predict([[500,0,0,0,0,10000000000000000000000000000000000000000000000000000000000000000000000000000]])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [60], in <cell line: 1>()
----> 1 DT2.predict([[500,0,0,0,0,10000000000000000000000000000000000000000000000000000000000000000000000000000]])

File ~\anaconda3\lib\site-packages\sklearn\tree\_classes.py:467, in BaseDecisionTree.predict(self, X, check_input)
    444 """Predict class or regression value for X.
    445
    446 For a classification model, the predicted class for each sample in X is
   (...)
    464     The predicted classes, or the predict values.
    465 """
    466 check_is_fitted(self)
--> 467 X = self._validate_X_predict(X, check_input)
    468 proba = self.tree_.predict(X)
    469 n_samples = X.shape[0]

File ~\anaconda3\lib\site-packages\sklearn\tree\_classes.py:433, in BaseDecisionTree._validate_X_predict(self, X, check_input)
    431 """Validate the training data on predict (probabilities)."""
    432 if check_input:
--> 433     X = self._validate_data(X, dtype=DTYPE, accept_sparse="csr", reset=False)
    434     if issparse(X) and (
    435         X.indices.dtype != np.intc or X.indptr.dtype != np.intc
    436     ):
    437         raise ValueError("No support for np.int64 index based sparse matrices")
```

Test case 6:

```
1 DT2.predict([[500,0,0,-240,0,-432123467545453789262538173934936318339]])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [79], in <cell line: 1>()
----> 1 DT2.predict([[500,0,0,-240,0,-432123467545453789262538173934936318339]])

File ~\anaconda3\lib\site-packages\sklearn\tree\_classes.py:467, in BaseDecisionTree.predict(self, X, check_input)
    444 """Predict class or regression value for X.
    445
    446 For a classification model, the predicted class for each sample in X is
   (...)
    464     The predicted classes, or the predict values.
    465 """
    466 check_is_fitted(self)
--> 467 X = self._validate_X_predict(X, check_input)
    468 proba = self.tree_.predict(X)
    469 n_samples = X.shape[0]

File ~\anaconda3\lib\site-packages\sklearn\tree\_classes.py:433, in BaseDecisionTree._validate_X_predict(self, X, check_input
    431 """Validate the training data on predict (probabilities)."""
    432 if check_input:
--> 433     X = self._validate_data(X, dtype=DTYPE, accept_sparse="csr", reset=False)
    434     if issparse(X) and (
    435         X.indices.dtype != np.intc or X.indptr.dtype != np.intc
    436     ):
    437         raise ValueError("No support for np.int64 index based sparse matrices")
```

Test case 7:

```
1 DT2.predict([[0,0,0,0,0,0]])
```

```
array(['Good'], dtype=object)
```

4 Resource & Environment Needs

4.1 Testing Tools

No testing tool is required. Manual Testing is done.

4.2 Test Environment

The following software's are required in addition to client-specific software.

- **Operating System**: Windows 8 and above
- **Development Environment**: Jupyter, PyCharm,,VS Code (Visual Studio Code) for scripting and coding related to data analysis and air pollution forecasting.
- **Data Format**: Ensure that the input data files are in the required format, such as CSV, Excel, or any other applicable format, for air pollution data analysis and forecasting.

5 Terms/Acronyms

TERM/ACRONYM	DEFINITION
AQI	Air Quality Index
SO2i	Sulfur Dioxide Index
NOi	Nitric Oxide Index
O3i	Ozone Index
PM2.5i	Particulate Matter with a diameter of 2.5 micrometers or smaller Index
PM10i	Particulate Matter with a diameter of 10 micrometers or smaller Index
COi	Carbon Monoxide Index