****

**Data preparation Feature Engineering and Model exploration**

Real-Time Air Quality Prediction and Classification System

**Group – 1 Members**

1. **Mohammad Wasil Jalali**
2. **Bahir Saidi**
3. **Mohammad Aref Rezvan Panah**
4. **Habibullah Farahmand**

**Data Preparation & Feature Engineering**

### 1. Overview

The data preparation and feature engineering phase is crucial in machine learning, ensuring raw data is transformed into a clean and structured format. This process enhances model performance by improving predictive capabilities, reducing noise, and increasing robustness to real-world variations.

### 2. Data Collection

The project relies on two key data sources:

* **NEPA Dataset**: Local air quality data from Kabul, which provides historical pollutant levels but requires augmentation due to missing entries. An official letter was sent to NEPA requesting data, and they agreed to provide it.
* **OpenWeather API**: A global data source offering real-time information about pollutants and weather conditions. Some other air pollution datasets required payment, leading to the decision to use OpenWeather API. Using Python’s requests library, data was collected in a structured manner by first retrieving countries, then states, and finally pollution data based on country and state. The collected data was then stored in a database.

### 3. Data Cleaning

* **Missing Values**: The OpenWeather dataset had no missing values, but the NEPA dataset had missing entries. These were imputed using **KNN imputation** after analyzing relationships, statistical properties, and the number of missing values. For the two missing pollutants (NO & NH3), data was extracted from OpenWeather's dataset for Kabul and imputed using the **FTLRI method**.
* **Feature Consistency**: The AQI values in NEPA were calculated using the formula employed by OpenWeather.
* **Outliers**: The dataset contained significant outliers, which could not be removed. Instead:
  + **Yeo-Johnson Transformation** was applied to correct right-skewed data.
  + **Robust Scaling** was used to normalize pollutant values.
* **Duplicates**: Identified and removed.
* **Correlation, Covariance & Mutual Information Analysis**: Used to refine feature selection.

### 4. Exploratory Data Analysis (EDA)

EDA was performed to understand data distribution, relationships, and anomalies:

* **Distributions**: Normal, Kernel Density Estimation (KDE), and Gamma distributions were analyzed.
* **Stationarity Analysis**: Conducted before handling outliers.
* **Correlation & Mutual Information**: Relationships among pollutants were examined.
* **Outlier Exploration**: Scatter plots and box plots used for detection.

**Visualizations (To be included separately):**

* Pair plots showing relationships between pollutants.
* Heatmap for feature correlations.
* Distribution plots for AQI and key pollutants.

### 5. Feature Engineering

* **Feature Selection**:
  + **Dropped NH3 & NO** due to high correlation with **NO2 & O3**, as they sufficiently represent their behavior in air quality prediction.
  + **PCA (Principal Component Analysis)** was tested but retained all features since reducing features resulted in a 9% variance loss.
* **Balancing**: Used **ARIMA methods** based on country features.
* **Time Features**: Transformed the NEPA dataset into a standardized format, then extracted year, month, day, hour, and minute features while dropping the raw time feature.
* **Zero Values**: Checked using scatter plots.
* **Geospatial Clustering**:
  + Used **Elbow and Silhouette Methods** to determine optimal k clusters.
  + Clustered latitude and longitude into regional\_id, then dropped the original coordinates.

### 6. Data Transformation

* **Power Transformation (Yeo-Johnson)**: Addressed skewness.
* **Robust Scaling**: Applied to pollutants.
* **Standardization**: Applied to temporal and spatial features.

**Model Exploration**

### 1. Model Selection

Models chosen for different tasks:

* **Prediction**: LSTM & TSMixer for sequential data.
* **Classification**: RandomForestClassifier.
* **Feature Robustness**: Reinforcement Learning (PPO).

**Strengths & Weaknesses**:

* **LSTM**:
  + Strengths: Captures long-term dependencies and sequential patterns; handles missing or irregular time-series data.
  + Weaknesses: Computationally expensive, prone to overfitting.
  + **Mitigation**: Used dropout layers and early stopping.
* **CNN**:
  + Strengths: Automatically extracts features, effective in high-dimensional classification.
  + Weaknesses: Requires a large labeled dataset.
  + **Mitigation**: Combined CNN with LSTM for sequential dependencies.
* **RL (PPO)**:
  + Strengths: Dynamically adapts to feature variability, improves system robustness.
  + Weaknesses: Resource-intensive, complex implementation.
  + **Mitigation**: Used pre-trained policies and reward shaping.

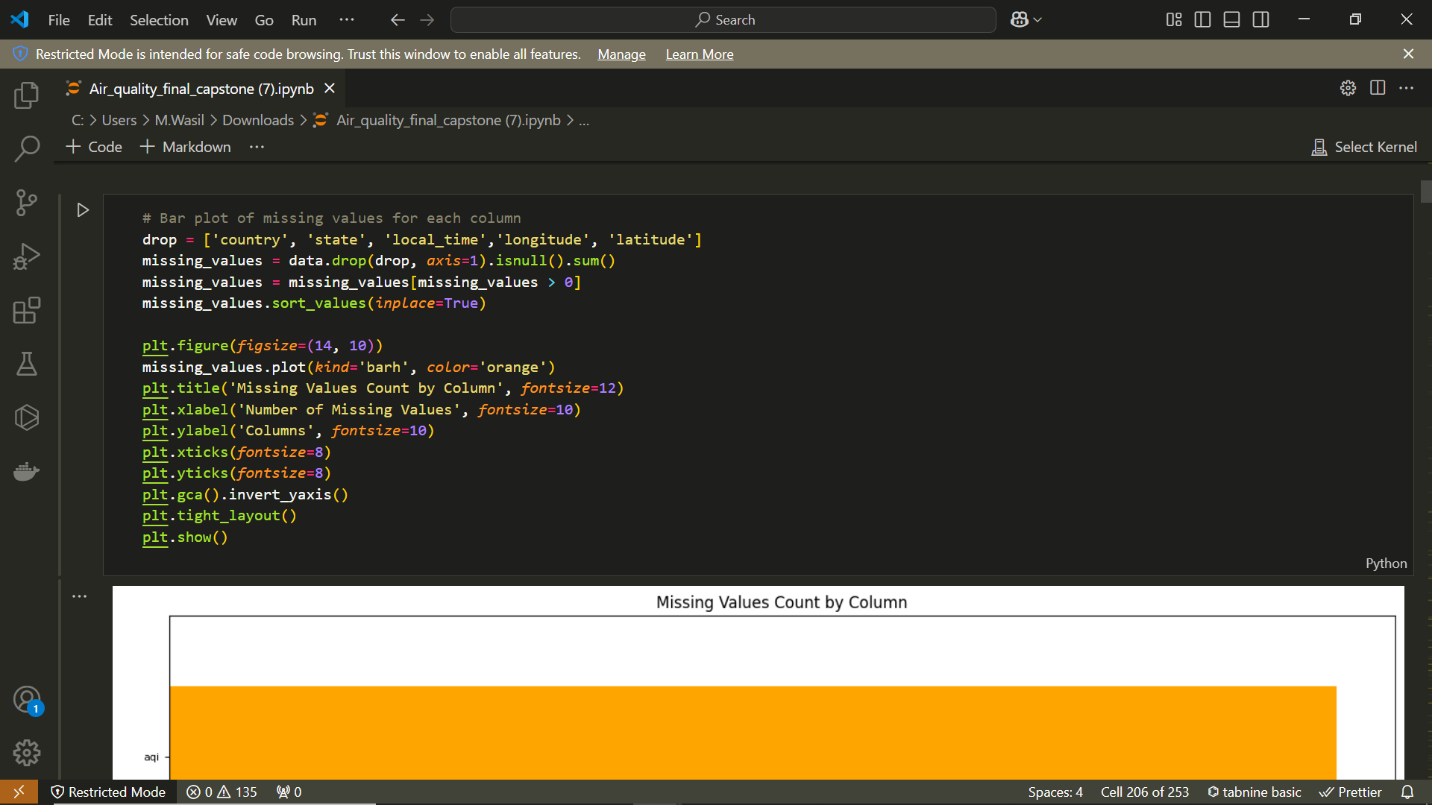
### 2. Model Training

* **Initial Models**: CNN, LSTM, and Transformer were tested but performed poorly due to data limitations (~95% accuracy on training but lower on testing).
* **Final Model Choices**:
  + **Classification**: RandomForestClassifier, which provided strong generalization.
  + **Regression**: LSTM & TSMixer. LSTM was trained with **5 layers** (LSTM 64 neurons, Layer Normalization, LSTM 32 neurons, Layer Normalization, Dense Layer). Hyperparameter tuning did not significantly improve performance. TSMixer performed slightly better and was chosen.
  + **Reinforcement Learning**: PPO was trained for long-term regression improvements.
* **Validation**: Used **K-fold cross-validation** to ensure robustness.

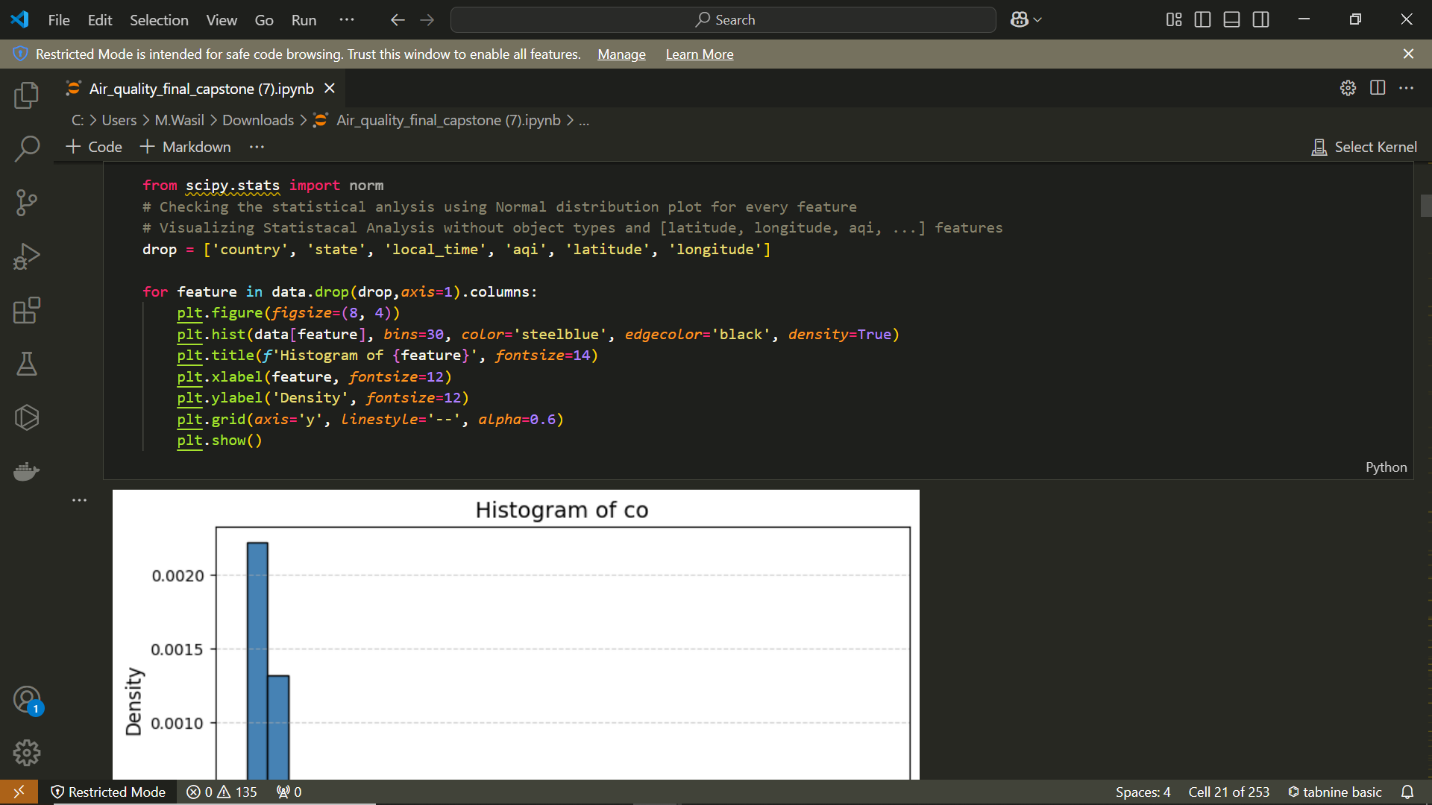
### 3. Model Evaluation

* **Classification**:
  + **Metrics**: Precision, Recall, F1-score, Confusion Matrix.
  + **Visualization**: Confusion matrix plot.
* **Regression**:
  + **Metrics**: MSE, RMSE, MAE.
  + **Visualization**: Scatter plots comparing predictions vs. actual values.

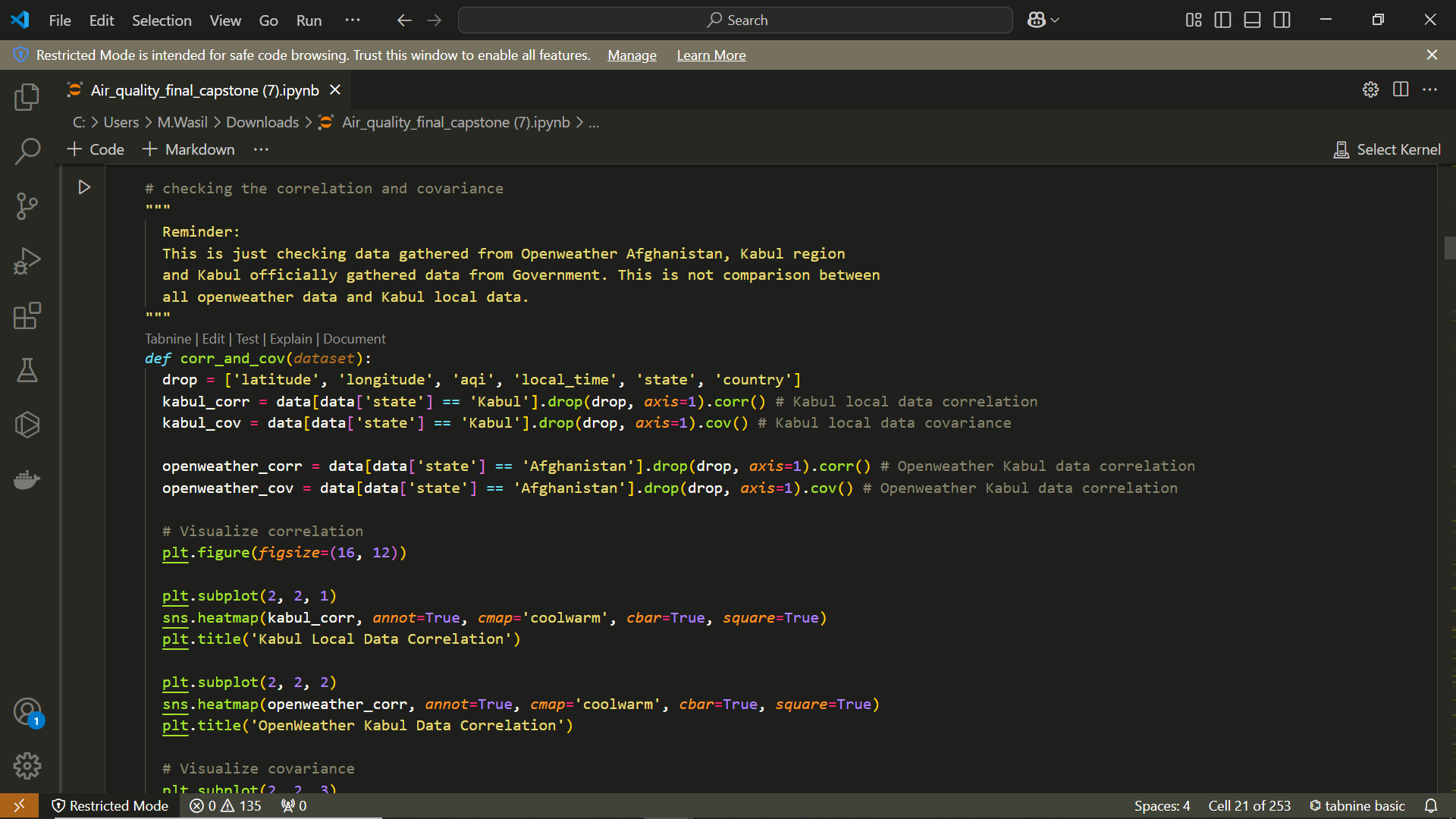
### 4. Code Implementation



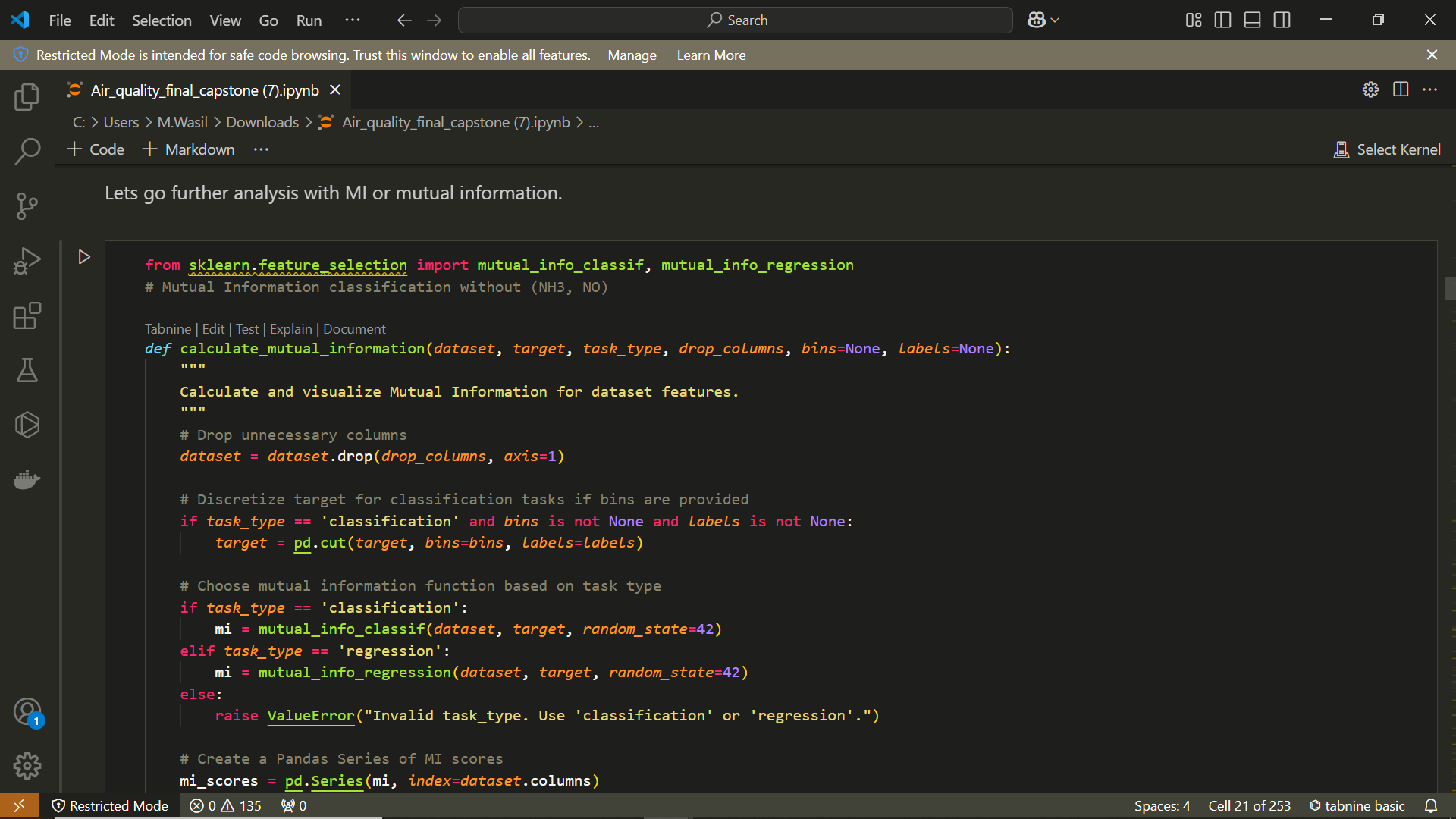
Checking missing values



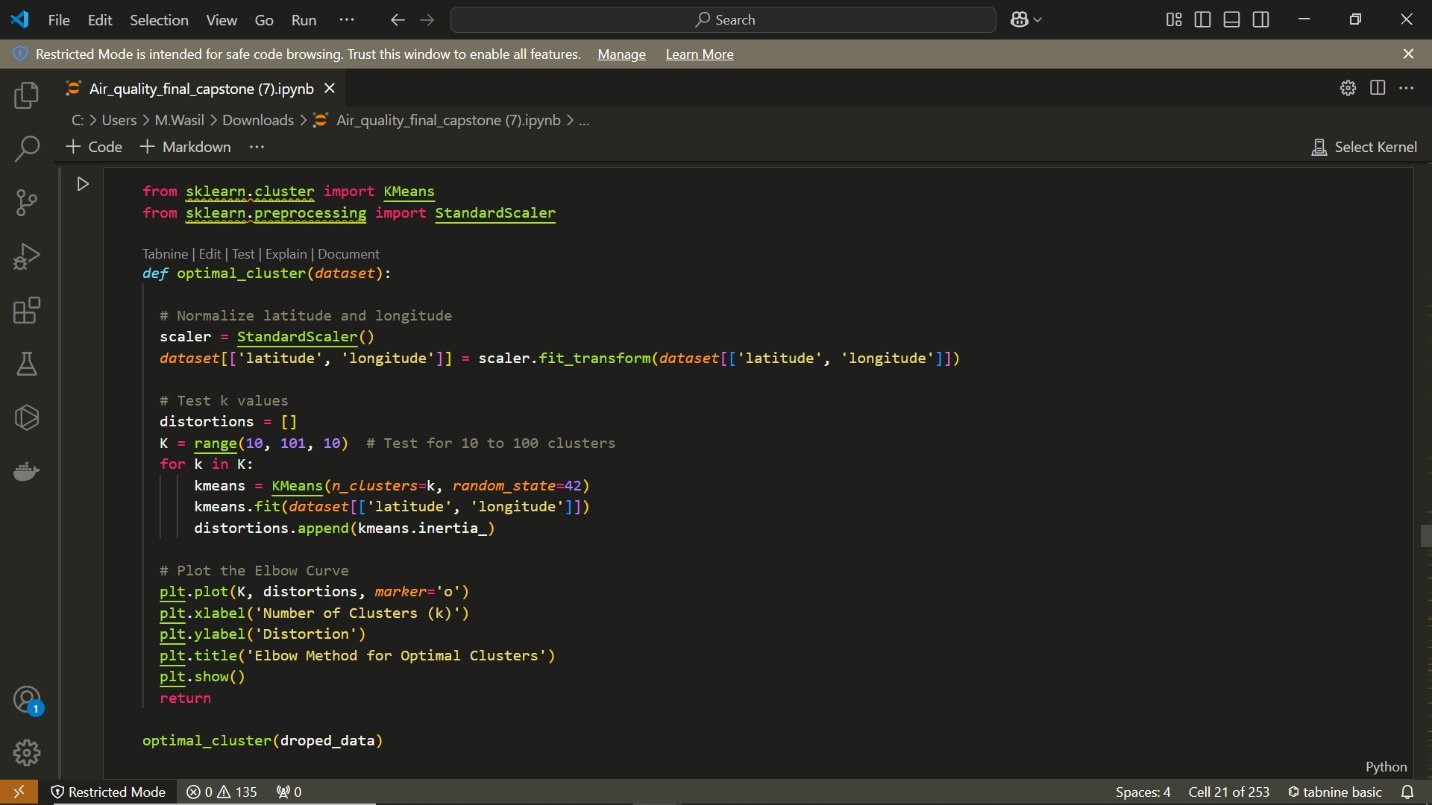
Checking normal distribution



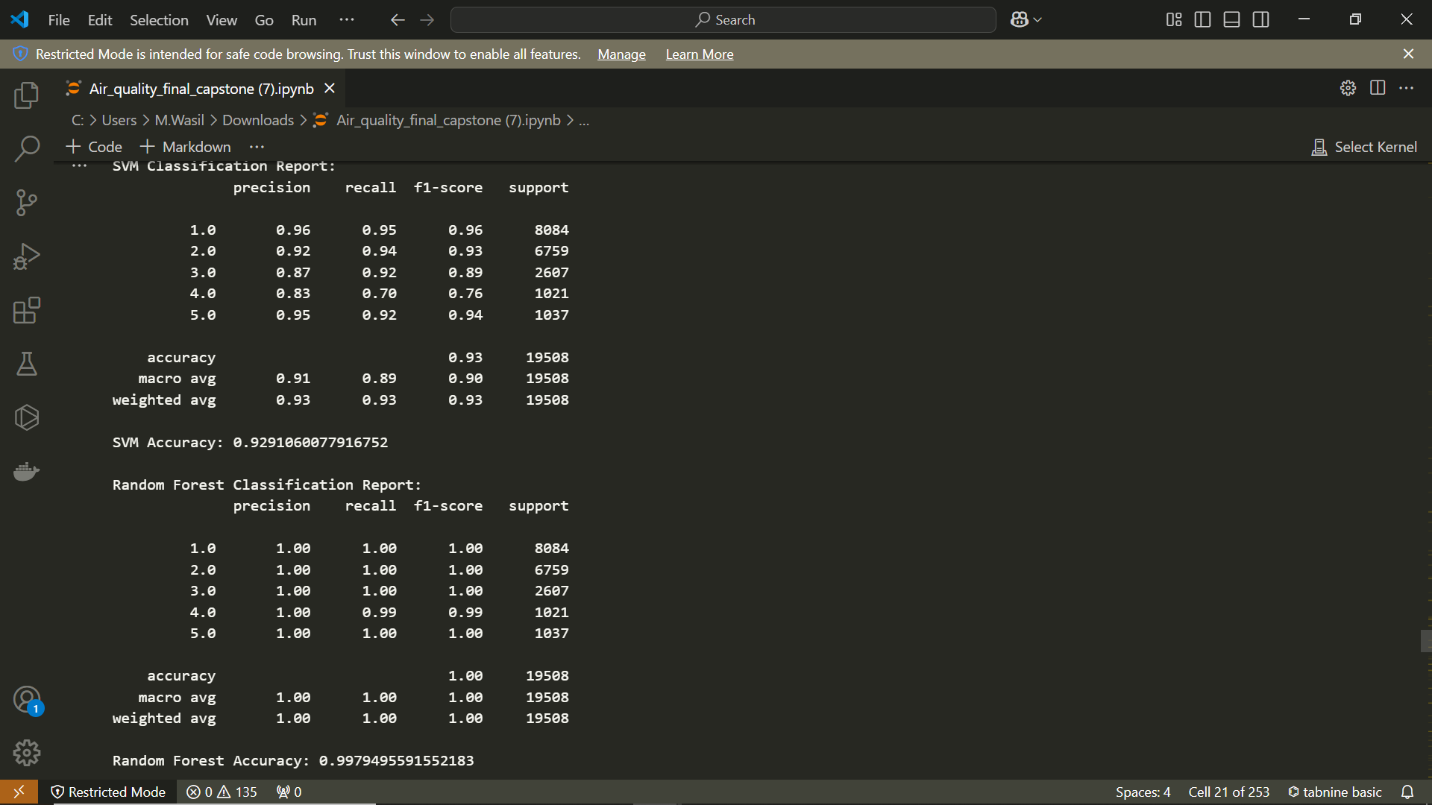
Afghanistan local data and Openweather Kabul data correlation checkup



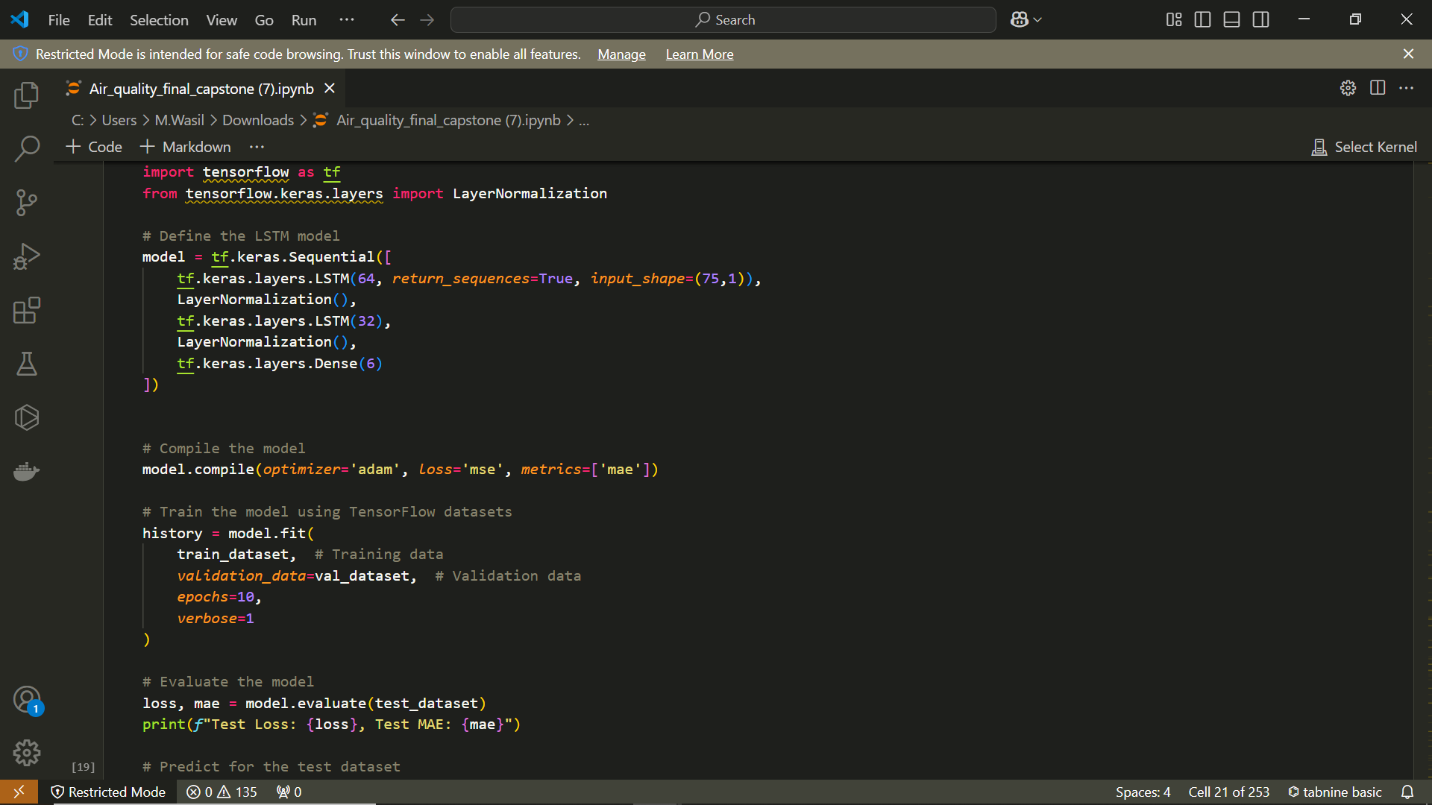
Checking feature importance using Mutual Information



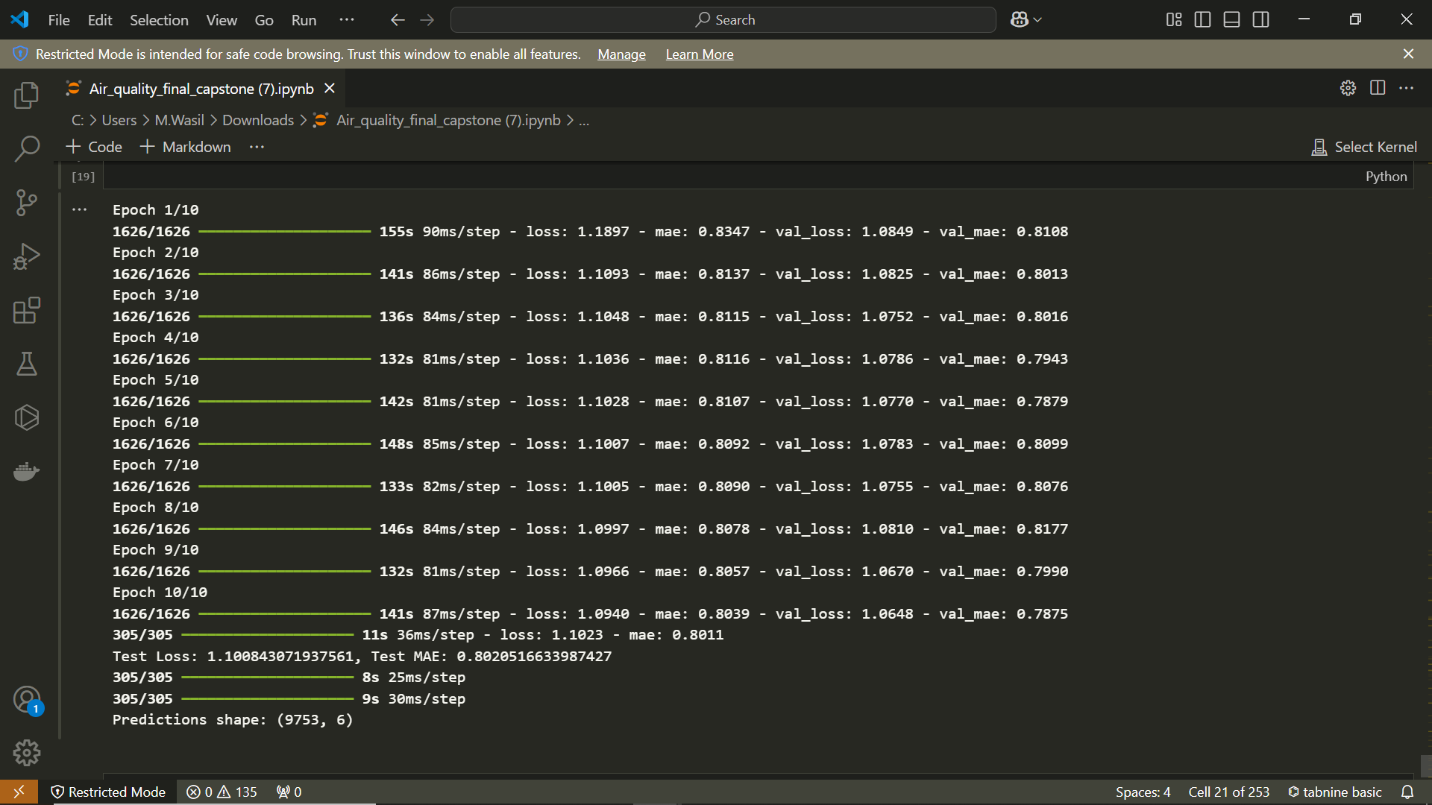
Finding best K for clustering using Elbow method



Classification Evaluation metrics



LSTM model training for Regression



LSTM Epochs

