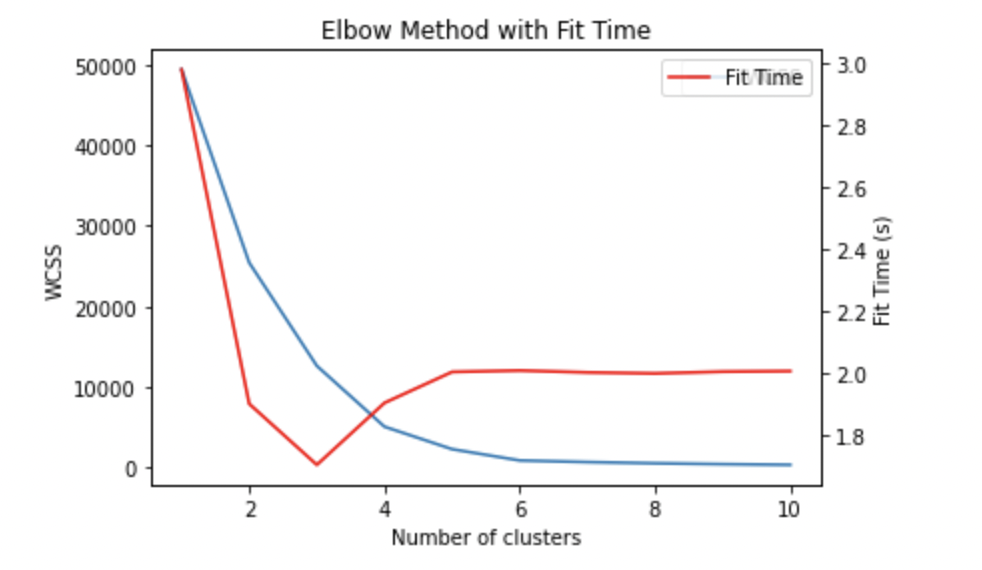
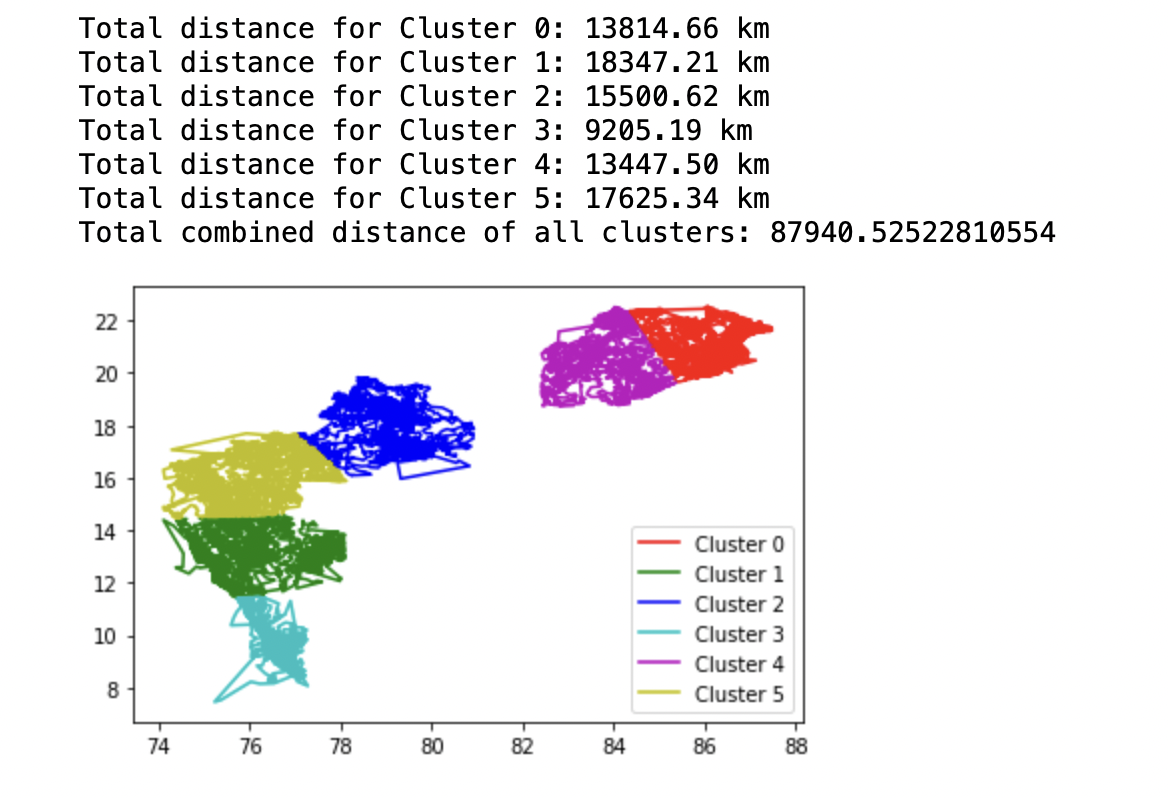
## Optimizing Delivery Routes

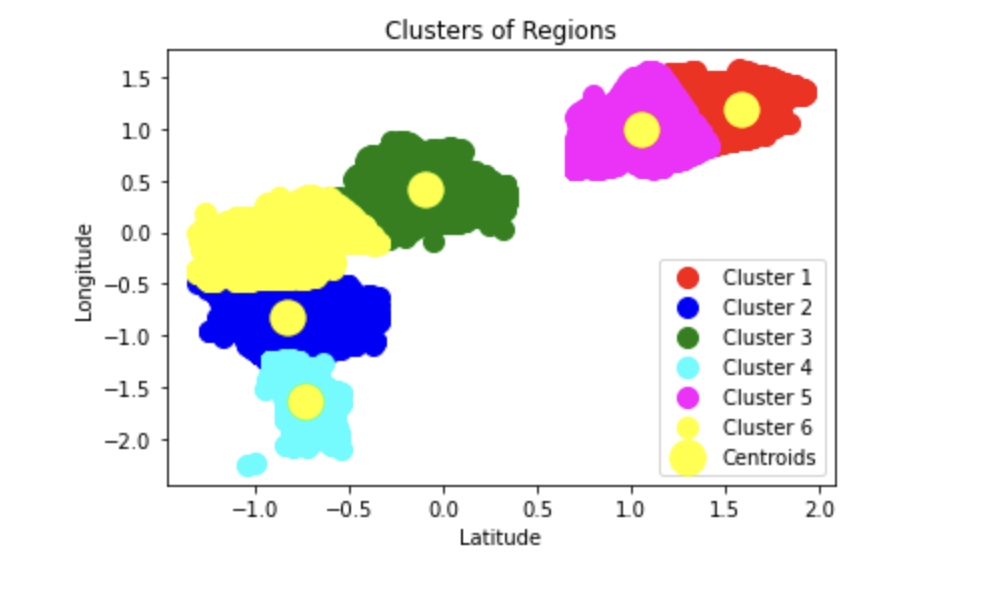
Below are the steps taken while writing the Python code:

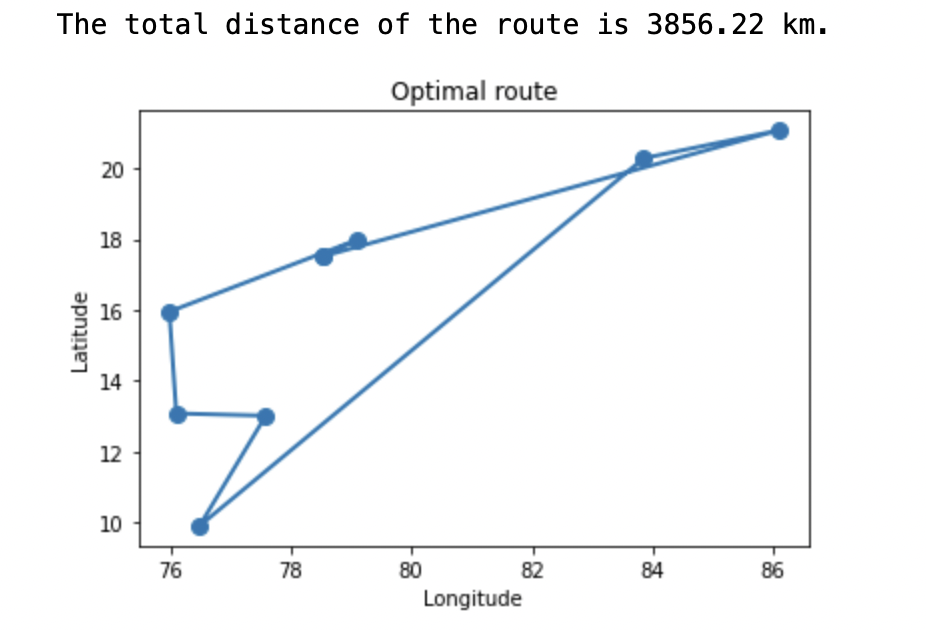
1. Importing data from Excel into a Pandas Dataframe
2. Import all of the required libraries
3. Recording the start-time at the beginning and end-time at last to calculate the total runtime of the code in Jupyter notebook
4. Null value treatment
5. Data-type evaluation. Latitude and longitude values are further filtered:
   1. -90 to +90 limits are applied on the values
   2. Regex is used to eliminate improper formats
6. Outlier analysis done on numerical data, particularly latitude and longitude values
7. A separate pandas data frame is created for states selected for route optimization
8. Outliers are observed at state level data, after outlier analysis was done as a whole.
9. Outlier analysis is done for every state in the new data frame to eliminate irregularities in the solution
10. Performing K-means Clustering on the state-level data and selecting optimal number of clusters using Elbow method



1. Optimal clusters is when the elbow curve converges / comes closest with the fit Time curve in the above plot
2. Based on the data provided on the factory location, the pincode closest to the centroid of all pincodes in the factory location is used as factory pincode
3. Solving TSP using nearest neighbor Algorithm to each cluster. Like in example shown below:

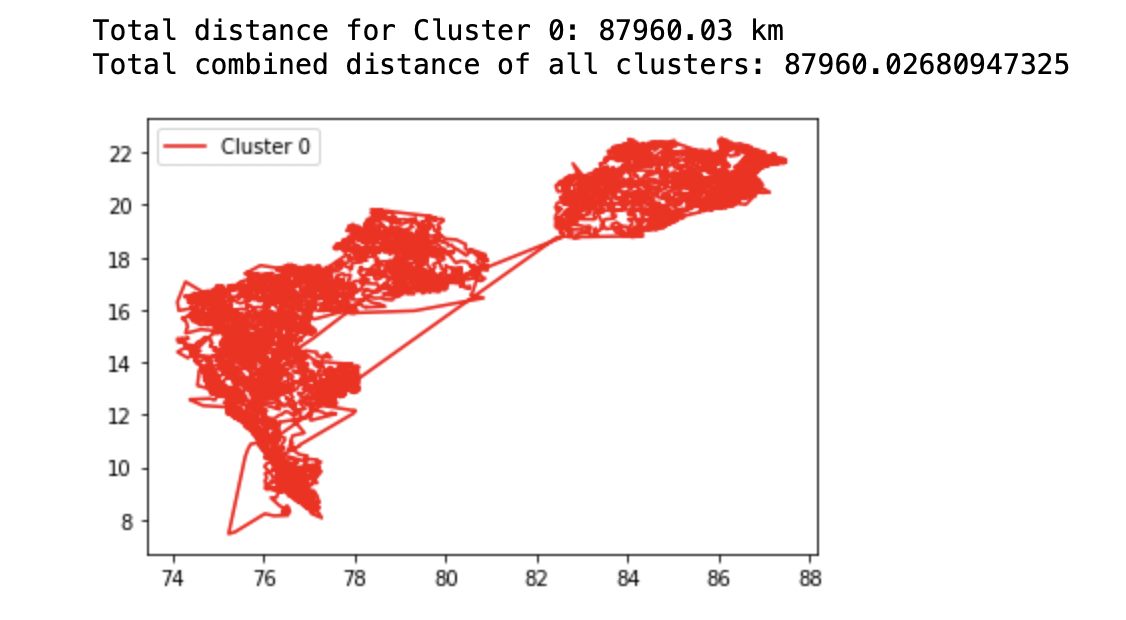
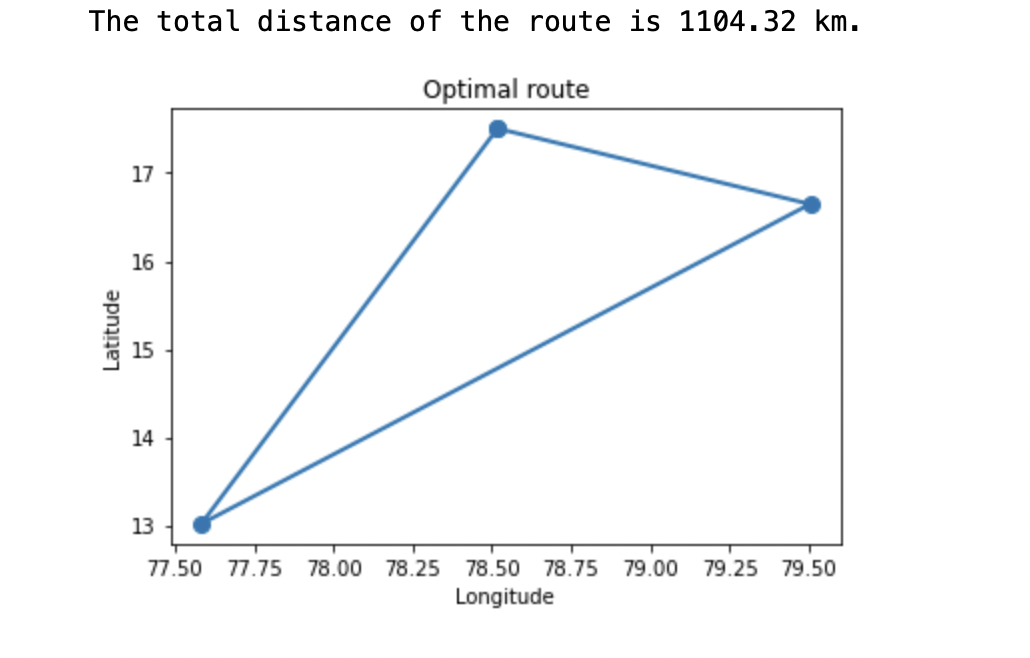
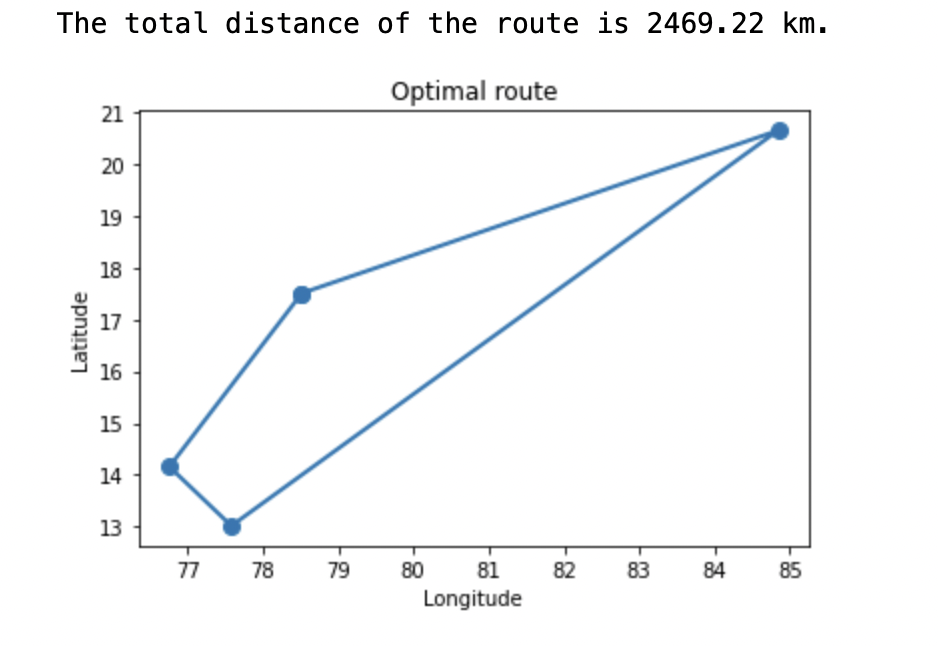
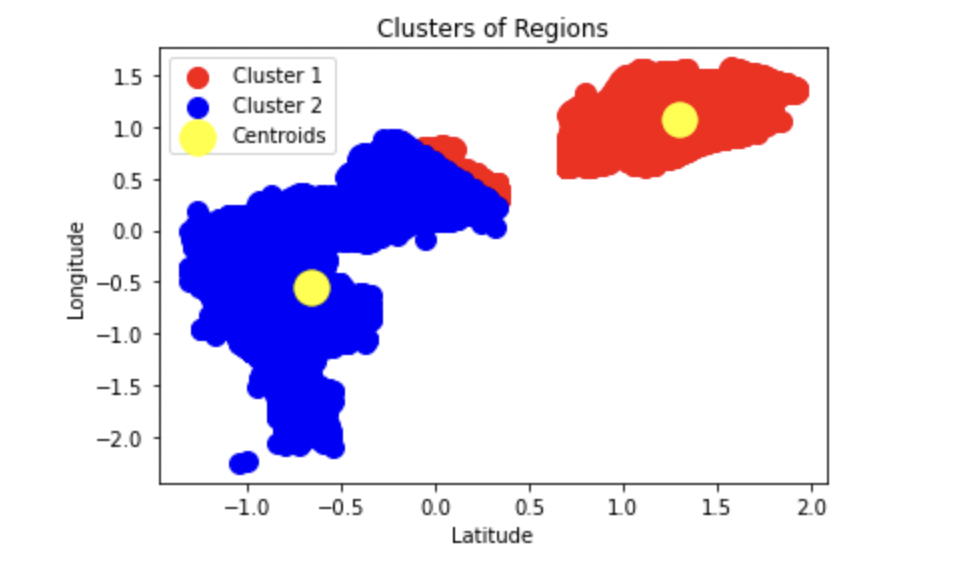
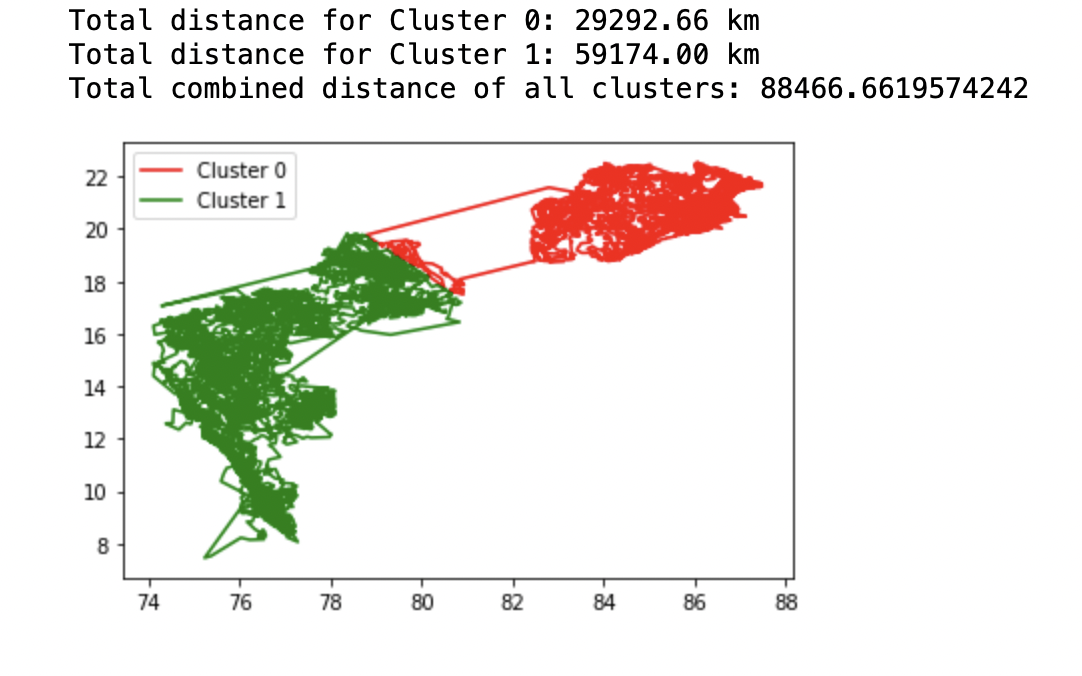


1. Storing the depot locations obtained from the previous step in each cluster in a new dataframe. Example below
2. Solving TSP for factory and Depot locations



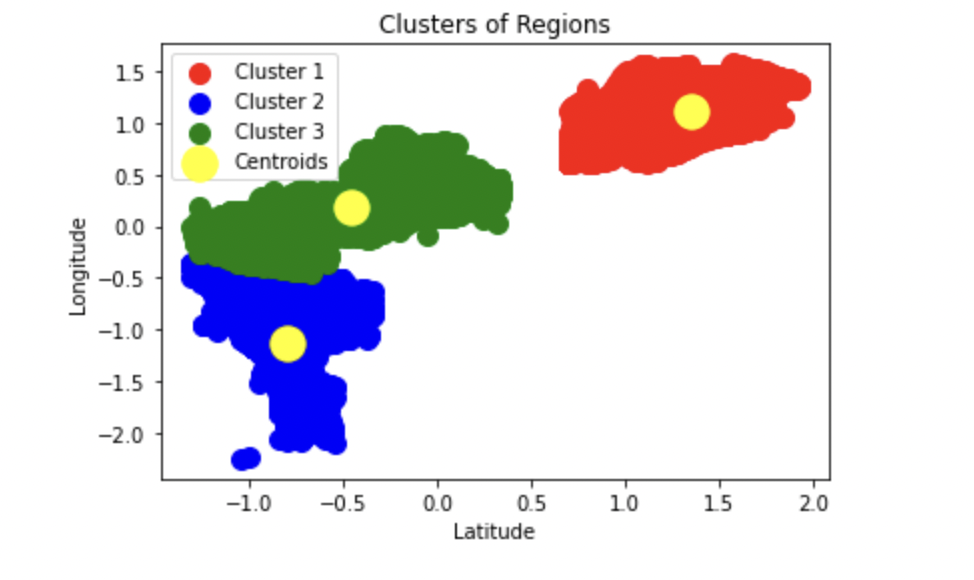
1. Output the total distance covered in all the clusters and factory - depot route
2. Measuring the total runtime of the Python notebook.

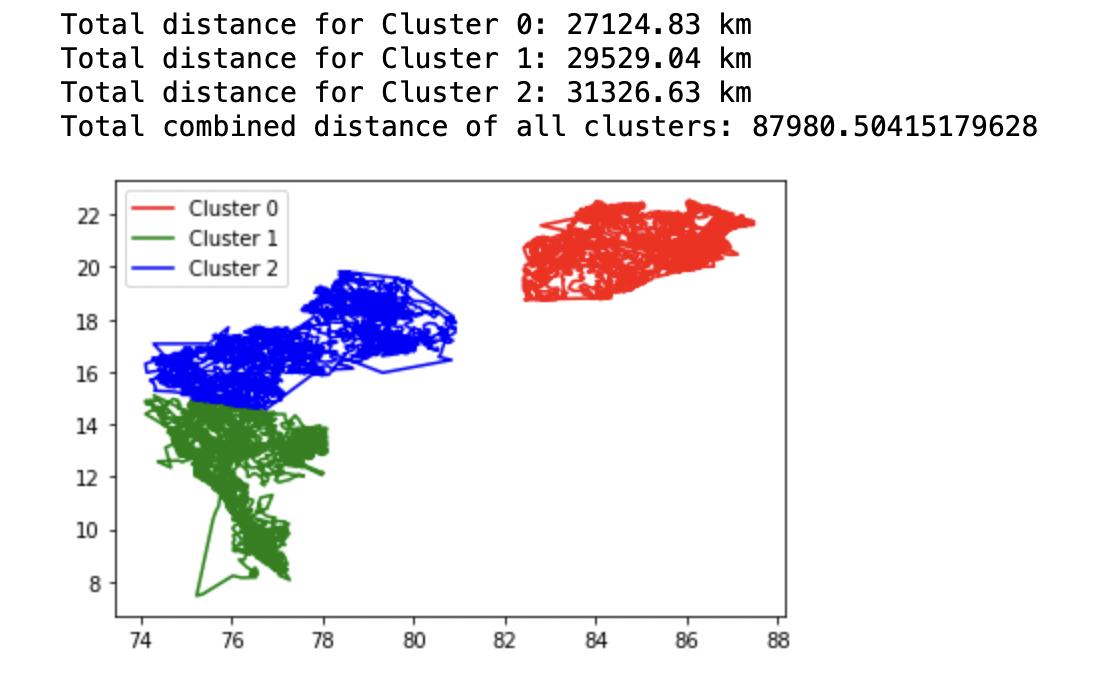
Below are the observations for state = ['TELANGANA', 'MAHARASTRA','KARNATAKA', 'ODISHA', 'KERALA']:

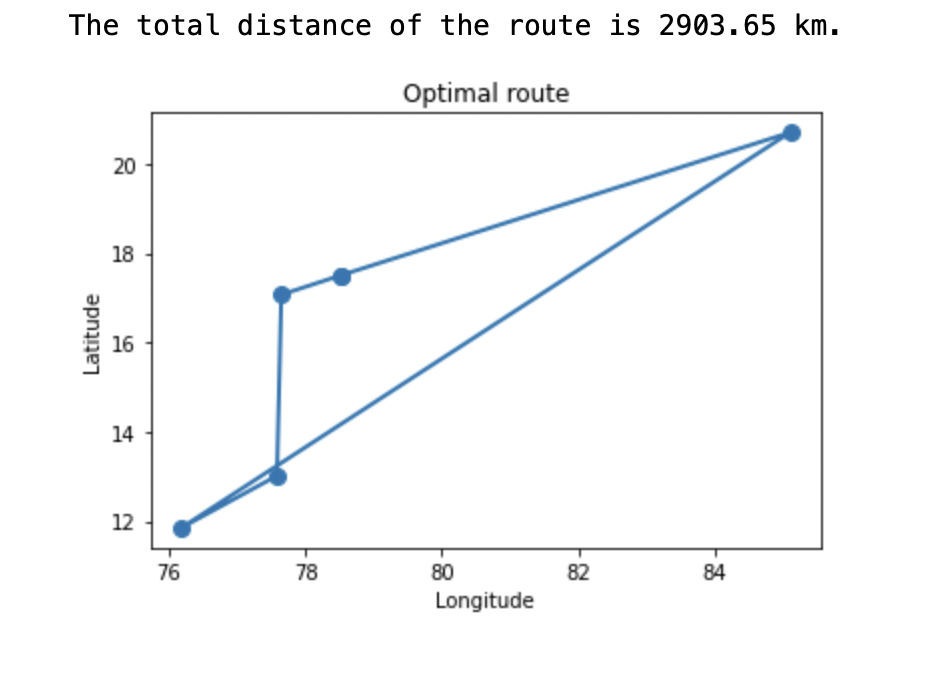
1. No. of Clusters = 1  
     
     
     
     
     
   Total time taken for the Code to run = 396.34 seconds  
   Total distance in Km (including trip from factory to depots): 89064.35 Km
2. No. of Clusters = 2  
     
     
     
     
   

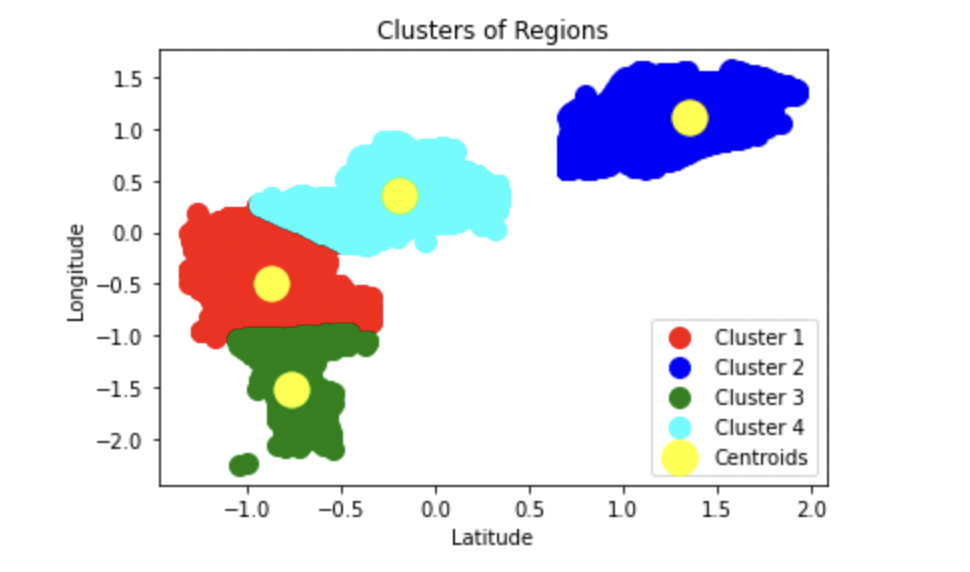
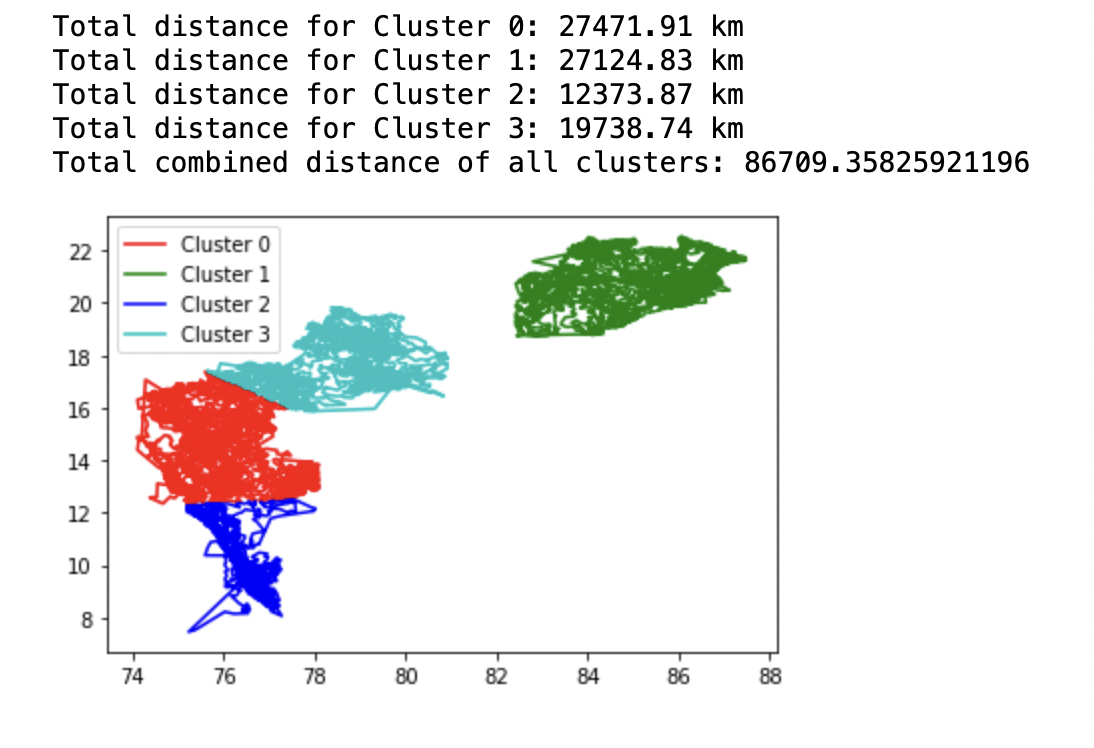
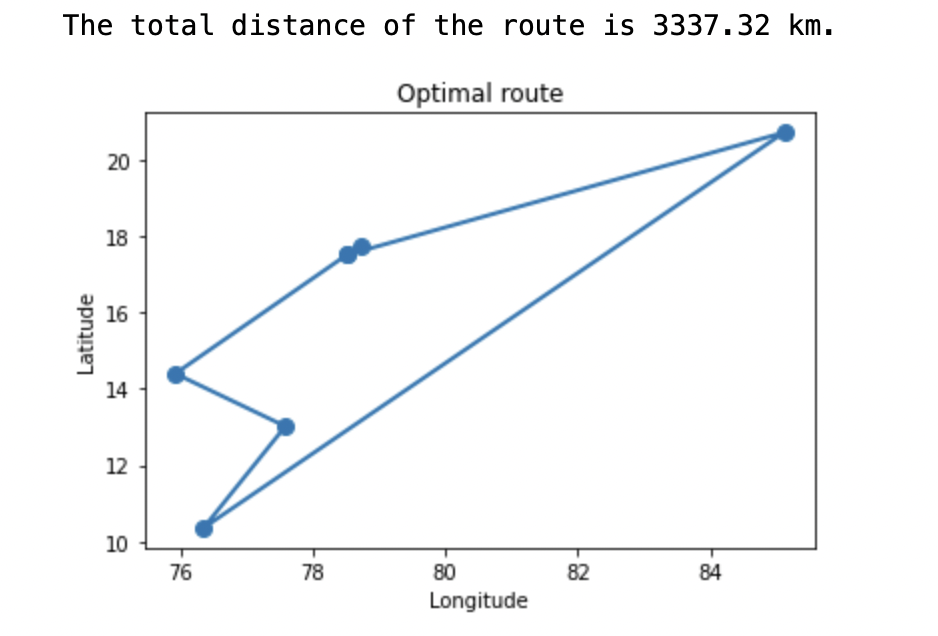
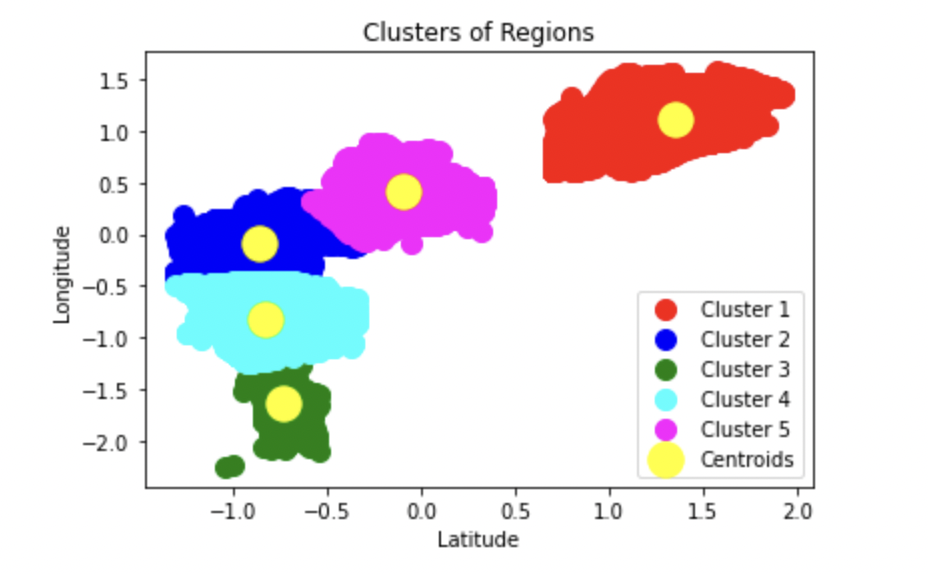
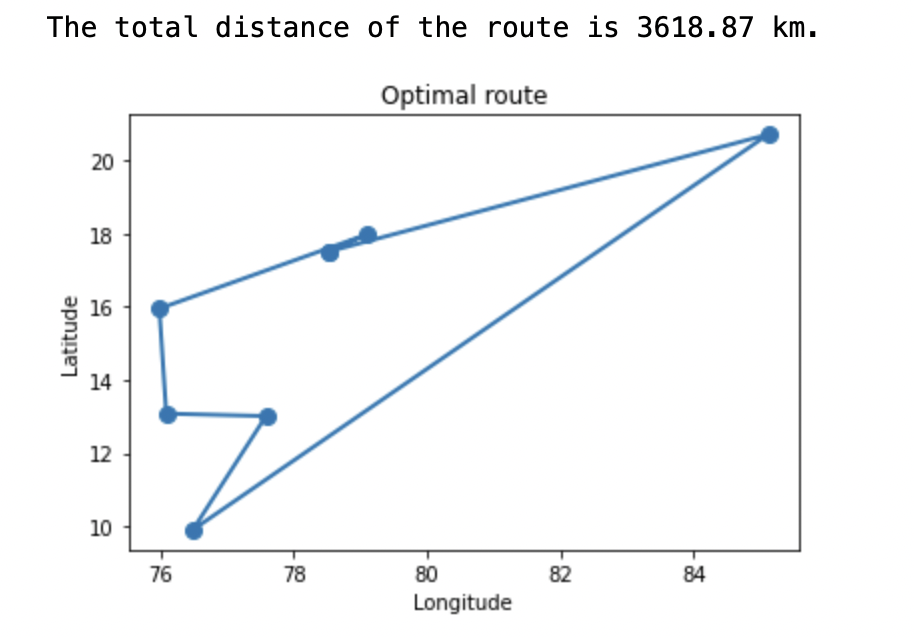
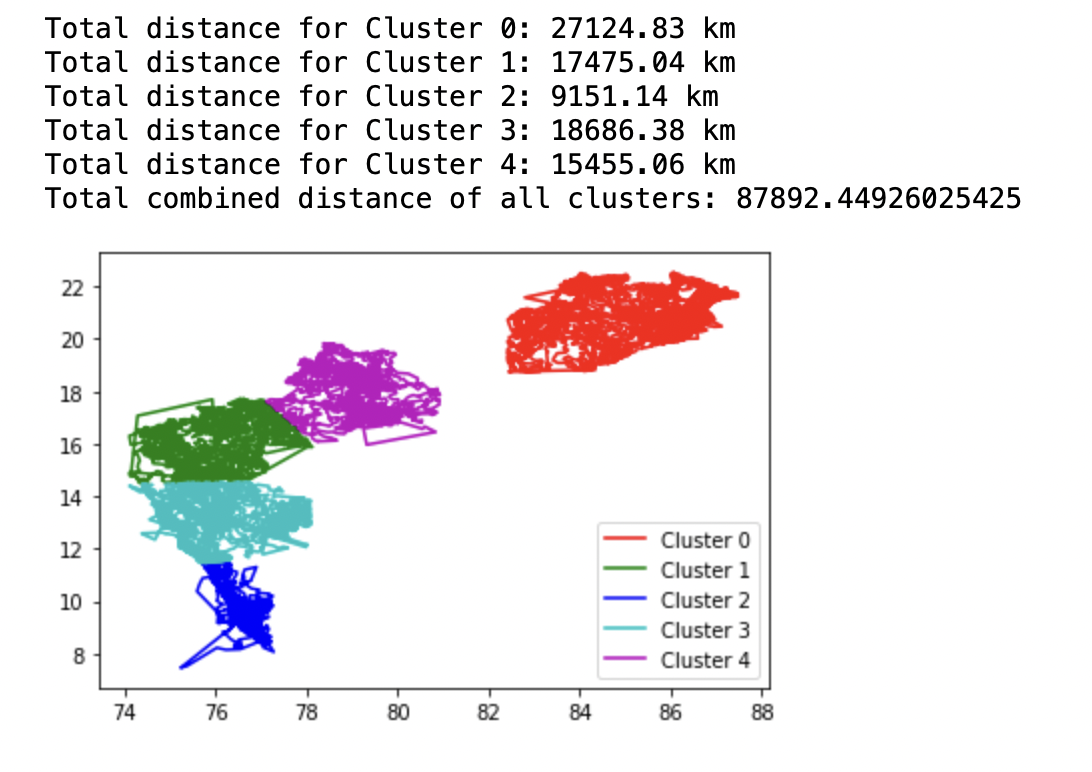
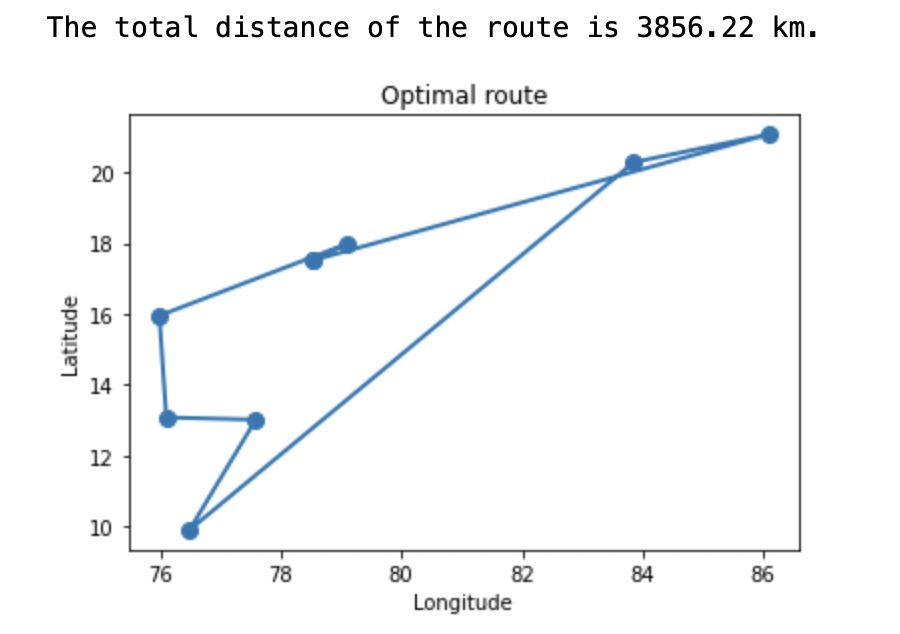
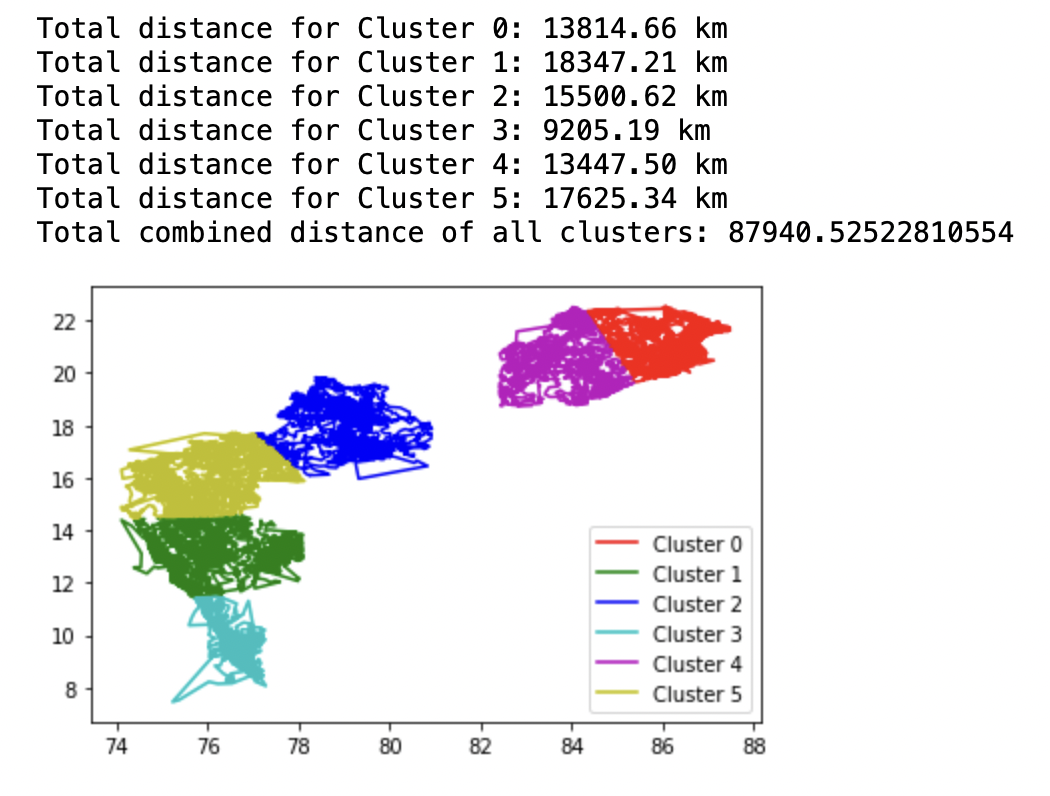
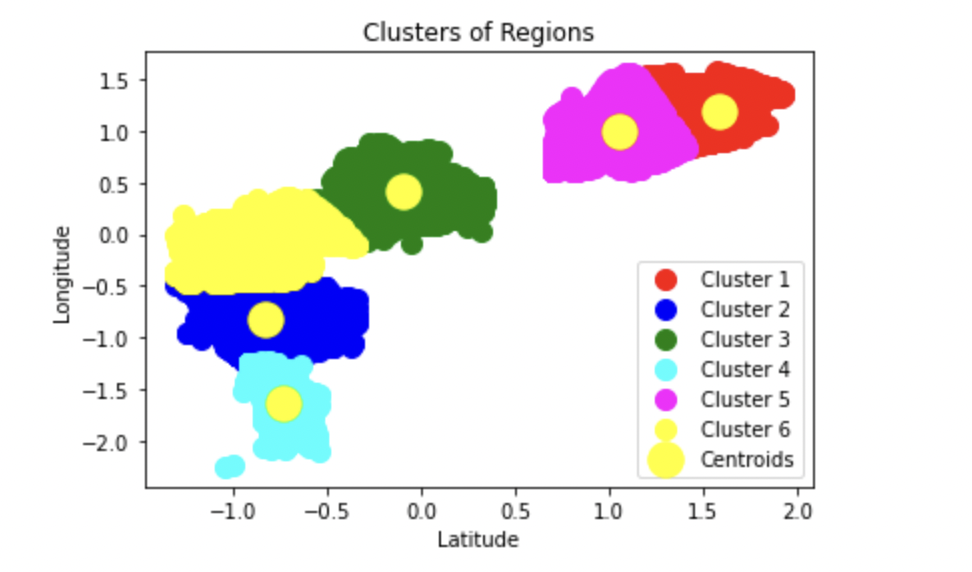
Total time taken for the Code to run = 57.23 seconds

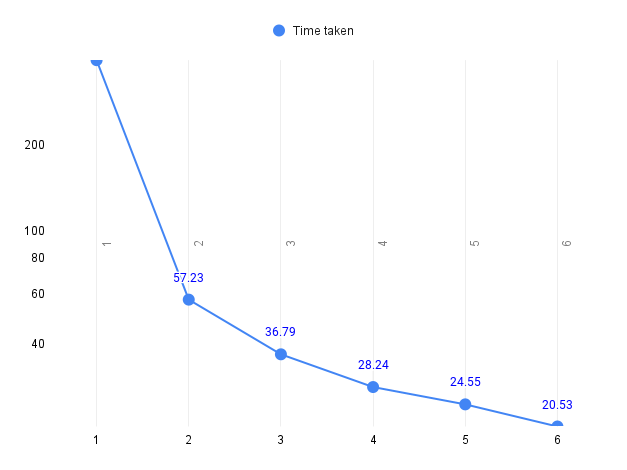
Total distance in Km (including trip from factory to depots): 90935.87 km

1. No. of Clusters = 3  
     
     
     
     
   

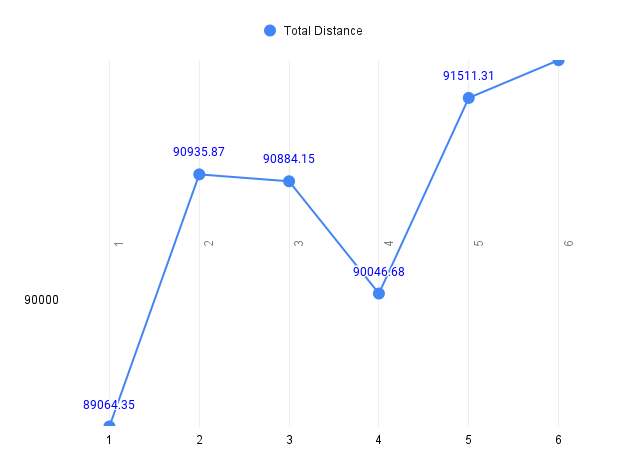


Total time taken for the Code to run = 36.79 seconds  
Total distance in Km (including trip from factory to depots): 90884.15

1. No. of Clusters = 4  
     
     
     
     
   Total time taken for the Code to run = 28.24 seconds  
   Total distance in Km (including trip from factory to depots): 90046.68
2. No. of Clusters = 5  
     
   Total time taken for the Code to run = 24.55 seconds  
   Total distance in Km (including trip from factory to depots): 91511.312  
   
3. No. of Clusters = 6  
     
     
     
     
   Total time taken for the Code to run = 20.53 seconds  
   Total distance in Km (including trip from factory to depots): 91796.74  
     
     
   



No. of cluster VS time taken to run the model

No. of cluster VS Distance covered   
  
  
\*\*Both the above plots are in log scale for better visualisation

**Conclusion:**

It is evident from the data that clustering minimises the total time taken to reach the solution but it may not be the perfect solution when the looked at the target variable alone. It is a trade-off between ‘time’ and ‘end-value’.