|  |
| --- |
|  |

|  |
| --- |
| **FINAL EXAM DANA-4830** |
| Predict annual average air pollution concentrations of various cities in the U.S. |
|  |
|  |
|  |
| **NAME: HARPREET KAUR** |
| **STUDENT ID: 100357359**  **SUBMITTED TO: MISS MONICA** |
|  |

**OBJECTIVE**

The main objective of air pollution data is to predict the annual average air pollution concentrations of various cities in the U.S. using predictors such as data about population density, urbanization, and road density, as well as satellite pollution data and chemical modelling data.

**STAGE 1 – EXPLORING DATA**

1. **Explore data, provide a descriptive analysis of the variables, and decide if you need to transform the variables to the right formats.**

The datasets provide the report of air pollution concentrations of various cities in the U.S. Moreover, the data set descriptions are as follows:

|  |  |  |
| --- | --- | --- |
| **TABLE 1: DATASET DESCRIPTION** | | |
| **Dataset** | **Number of rows** | **Number of Columns** |
| ***Pm2.5\_data*** | 876 | 50 |
| ***Total*** | 876 | 50 |

In the dataset, there are 50 attributes/features with values for each of the 876 monitors (observations). **Table-1**

**DESCRIPTIVE ANALYSIS OF VARIABLES**

* Checking the structure of variables, found that there were 876 observations with 50 numbers of features**.** The value which is the most important column indicates the PM2.5 concentration level average of fine particles/volume of air for 876 gravimetric monitors for 2008.

|  |
| --- |
|  |

* Checked missing data of all the variables and noted that there was no missing value in the dataset. So, we can proceed further.

|  |  |  |
| --- | --- | --- |
|  |  |  |

* Noted that some columns are numeric and do not contains numerical values example id, flips, zcta thus converting them into the factor variable.

|  |
| --- |
|  |
|  |

* Used skim package to check the distribution of the numerical dataset.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **skim\_variabe** | **miss** | **complete\_rate** | **mean** | **sd** | **p0** | **p25** | **p50** | **p75** | **p100** | **Histogram** | | value | 0 | 1 | 10.81 | 2.58 | 3.02 | 9.27 | 11.15 | 12.37 | 23.16 | ▂▆▇▁▁ | | lat | 0 | 1 | 38.48 | 4.62 | 25.47 | 35.03 | 39.30 | 41.66 | 48.40 | ▁▃▅▇▂ | | lon | 0 | 1 | -91.74 | 14.96 | -124.18 | -99.16 | -87.47 | -80.69 | -68.04 | ▃▂▃▇▃ | | CMAQ | 0 | 1 | 8.41 | 2.97 | 1.63 | 6.53 | 8.62 | 10.24 | 23.13 | ▃▇▃▁▁ | | zcta\_area | 0 | 1 | 183173481.91 | 542598878.48 | 15459.00 | 14204601.75 | 37653560.50 | 160041508.25 | 8164820625.00 | ▇▁▁▁▁ | | zcta\_pop | 0 | 1 | 24227.58 | 17772.16 | 0.00 | 9797.00 | 22014.00 | 35004.75 | 95397.00 | ▇▇▃▁▁ | | imp\_a500 | 0 | 1 | 24.72 | 19.34 | 0.00 | 3.70 | 25.12 | 40.22 | 69.61 | ▇▅▆▃▂ | | imp\_a1000 | 0 | 1 | 24.26 | 18.02 | 0.00 | 5.32 | 24.53 | 38.59 | 67.50 | ▇▅▆▃▁ | | imp\_a5000 | 0 | 1 | 19.93 | 14.72 | 0.05 | 6.79 | 19.07 | 30.11 | 74.60 | ▇▆▃▁▁ | | imp\_a10000 | 0 | 1 | 15.82 | 13.81 | 0.09 | 4.54 | 12.36 | 24.17 | 72.09 | ▇▃▂▁▁ | | imp\_a15000 | 0 | 1 | 13.43 | 13.12 | 0.11 | 3.24 | 9.67 | 20.55 | 71.10 | ▇▃▁▁▁ | | county\_area | 0 | 1 | 3768701992.12 | 6212829553.56 | 33703512.00 | 1116536297.50 | 1690826566.50 | 2878192209.00 | 51947229509.00 | ▇▁▁▁▁ | | county\_pop | 0 | 1 | 687298.44 | 1293488.74 | 783.00 | 100948.00 | 280730.50 | 743159.00 | 9818605.00 | ▇▁▁▁▁ | | log\_dist\_to\_prisec | 0 | 1 | 6.19 | 1.41 | -1.46 | 5.43 | 6.36 | 7.15 | 10.45 | ▁▁▃▇▁ | | log\_pri\_length\_5000 | 0 | 1 | 9.82 | 1.08 | 8.52 | 8.52 | 10.05 | 10.73 | 12.05 | ▇▂▆▅▂ | | log\_pri\_length\_10000 | 0 | 1 | 10.92 | 1.13 | 9.21 | 9.80 | 11.17 | 11.83 | 13.02 | ▇▂▇▇▃ | | log\_pri\_length\_15000 | 0 | 1 | 11.50 | 1.15 | 9.62 | 10.87 | 11.72 | 12.40 | 13.59 | ▆▂▇▇▃ | | log\_pri\_length\_25000 | 0 | 1 | 12.24 | 1.10 | 10.13 | 11.69 | 12.46 | 13.05 | 14.36 | ▅▃▇▇▃ | | log\_prisec\_length\_500 | 0 | 1 | 6.99 | 0.95 | 6.21 | 6.21 | 6.21 | 7.82 | 9.40 | ▇▁▂▂▁ | | log\_prisec\_length\_1000 | 0 | 1 | 8.56 | 0.79 | 7.60 | 7.60 | 8.66 | 9.20 | 10.47 | ▇▅▆▃▁ | | log\_prisec\_length\_5000 | 0 | 1 | 11.28 | 0.78 | 8.52 | 10.91 | 11.42 | 11.83 | 12.78 | ▁▁▃▇▃ | | log\_prisec\_length\_10000 | 0 | 1 | 12.41 | 0.73 | 9.21 | 11.99 | 12.53 | 12.94 | 13.85 | ▁▁▃▇▅ | | log\_prisec\_length\_15000 | 0 | 1 | 13.03 | 0.72 | 9.62 | 12.59 | 13.13 | 13.57 | 14.41 | ▁▁▃▇▅ | | log\_prisec\_length\_25000 | 0 | 1 | 13.82 | 0.70 | 10.13 | 13.38 | 13.92 | 14.35 | 15.23 | ▁▁▃▇▆ | | log\_nei\_2008\_pm25\_sum\_10000 | 0 | 1 | 3.97 | 2.35 | 0.00 | 2.15 | 4.29 | 5.69 | 9.12 | ▆▅▇▆▂ | | log\_nei\_2008\_pm25\_sum\_15000 | 0 | 1 | 4.72 | 2.25 | 0.00 | 3.47 | 5.00 | 6.35 | 9.42 | ▃▃▇▇▂ | | log\_nei\_2008\_pm25\_sum\_25000 | 0 | 1 | 5.67 | 2.11 | 0.00 | 4.66 | 5.91 | 7.28 | 9.65 | ▂▂▇▇▃ | | log\_nei\_2008\_pm10\_sum\_10000 | 0 | 1 | 4.35 | 2.32 | 0.00 | 2.69 | 4.62 | 6.07 | 9.34 | ▅▅▇▇▂ | | log\_nei\_2008\_pm10\_sum\_15000 | 0 | 1 | 5.10 | 2.18 | 0.00 | 3.87 | 5.39 | 6.72 | 9.71 | ▂▃▇▇▂ | | log\_nei\_2008\_pm10\_sum\_25000 | 0 | 1 | 6.07 | 2.01 | 0.00 | 5.10 | 6.37 | 7.52 | 9.88 | ▁▂▆▇▃ | | popdens\_county | 0 | 1 | 551.76 | 1711.51 | 0.26 | 40.77 | 156.67 | 510.81 | 26821.91 | ▇▁▁▁▁ | | popdens\_zcta | 0 | 1 | 1279.66 | 2757.49 | 0.00 | 101.15 | 610.35 | 1382.52 | 30418.84 | ▇▁▁▁▁ | | nohs | 0 | 1 | 6.99 | 7.21 | 0.00 | 2.70 | 5.10 | 8.80 | 100.00 | ▇▁▁▁▁ | | somehs | 0 | 1 | 10.17 | 6.20 | 0.00 | 5.90 | 9.40 | 13.90 | 72.20 | ▇▂▁▁▁ | | hs | 0 | 1 | 30.32 | 11.40 | 0.00 | 23.80 | 30.75 | 36.10 | 100.00 | ▂▇▂▁▁ | | somecollege | 0 | 1 | 21.58 | 8.60 | 0.00 | 17.50 | 21.30 | 24.70 | 100.00 | ▆▇▁▁▁ | | associate | 0 | 1 | 7.13 | 4.01 | 0.00 | 4.90 | 7.10 | 8.80 | 71.40 | ▇▁▁▁▁ | | bachelor | 0 | 1 | 14.90 | 9.71 | 0.00 | 8.80 | 12.95 | 19.23 | 100.00 | ▇▂▁▁▁ | | grad | 0 | 1 | 8.91 | 8.65 | 0.00 | 3.90 | 6.70 | 11.00 | 100.00 | ▇▁▁▁▁ | | pov | 0 | 1 | 14.95 | 11.33 | 0.00 | 6.50 | 12.10 | 21.23 | 65.90 | ▇▅▂▁▁ | | hs\_orless | 0 | 1 | 47.48 | 16.75 | 0.00 | 37.93 | 48.65 | 59.10 | 100.00 | ▁▃▇▃▁ | | urc2013 | 0 | 1 | 2.92 | 1.52 | 1.00 | 2.00 | 3.00 | 4.00 | 6.00 | ▇▅▃▂▁ | | urc2006 | 0 | 1 | 2.97 | 1.52 | 1.00 | 2.00 | 3.00 | 4.00 | 6.00 | ▇▅▃▂▁ | | aod | 0 | 1 | 43.70 | 19.56 | 5.00 | 31.66 | 40.17 | 49.67 | 143.00 | ▃▇▁▁▁ | |

**Summary:**

* + No missing value in the numerical dataset.
  + **Mean** represent the average value of each column
  + **Standard deviation explains about** spread or dispersion of our data points around the mean value
  + **Quantile (p0 to p100)** tells about the minimum, maximum, range of the dataset
  + **Histogram** tells about the distribution of the numerical values**.** To illustrate some of the **Right skewerd Histogram are (**urc2013, urc2006,somecollege, associate, zcta\_area, zcta\_pop**). Left skewed (**value, log\_dist\_to\_prisec, log\_prisec\_length\_5000, aod**). Normally distributed (**hs\_orless, CMAQ, log\_nei\_2008\_pm25\_sum\_25000**).**
* Used bar plot to check count for the categorical dataset.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **variable** | **miss** | **complete\_rate** | **min** | **max** | **empty** | **n\_unique** | **whitespace** | | state | 0 | 1 | 4 | 20 | 0 | 49 | 0 | | county | 0 | 1 | 3 | 20 | 0 | 471 | 0 | | city | 0 | 1 | 4 | 48 | 0 | 607 | 0 | |

|  |
| --- |
| **STATE:** |
| CITY: |

* Factor variables achieved from skim package

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **variable** | **miss** | **complete\_rate** | **ordered** | **n\_unique** | **top\_counts** | | id | 0 | 1 | FALSE | 876 | 100: 1, 102: 1, 103: 1, 104: 1 | | fips | 0 | 1 | FALSE | 569 | 170: 12, 603: 10, 261: 9, 107: 8 | | zcta | 0 | 1 | FALSE | 842 | 475: 3, 110: 2, 160: 2, 290: 2 | |

1. **How many states are there in the dataset?**

|  |
| --- |
|  |

There are 49 states in the dataset as shown above. **Using Annex-2.**

1. **How many stations are there per city?**

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
|  |

Using ggplotly library of R can check the number of stations per city. **Using Annex- 3**. To illustrate some of them

|  |  |
| --- | --- |
| **City** | **Count of stations** |
| New York | 9 |
| Cleveland | 6 |
| Baltimore | 5 |
| Chicago | 5 |
| Detroit | 5 |

1. **Conduct a correlation test and provide the results focusing on highly correlated pairs.**

**In the correlation plot, we can see that if the correlation value is close to 1 (dark red color) then it means it has a strong correlation and if the correlation value is close to -1 (dark blue color) then it has a strong negative correlation.**

* Variables have strong correlation value as they are close to 1. To illustrate some of them are Log\_pri\_length\_5000 and log\_prisec\_length\_10000 shows correlation value of 87%, Log\_pri\_length\_5000 and log\_prisec\_length\_10000 shows correlation value of 81%,

Log\_pri\_length\_25000 and log\_prisec\_length\_10000 shows correlation value of 86% these are the road density variables.

* imp\_a500 and imp\_a1000 have 97% , imp\_a5000 and imp\_a500 have 81% of strong correlation value. Where imp variables are development variables.
* log\_pri\_length\_5000 and log\_pri\_length\_1000 have 87% log\_pri\_length\_5000, log\_pri\_length\_15000 and log\_pri\_length\_25000 have strong correlation. Whre log\_pri are primary and secondary road length variables.
* log\_nei\_2008\_pm25\_sum\_15000,log\_nei\_2008\_pm25\_sum\_25000, log\_nei\_2008\_pm10\_sum\_10000,log\_nei\_2008\_pm10\_sum\_15000, log\_nei\_2008\_pm25\_sum\_25000 have strong correlation. Where log\_nei indicates emission variables.

|  |
| --- |
|  |

**STAGE 2 – PRE- PROCESSING THE DATASET WITH A VIEW TO DEVELOP MODELS**

1. **Report on the size of both datasets. Check the proportion of cities for both datasets.**

Split the data is to use the ratio of 2:3. And then checked the train and test dataset dimension.

|  |
| --- |
|  |

For the proportion of the dataset. **Used Annex- 5**

|  |  |
| --- | --- |
|  |  |

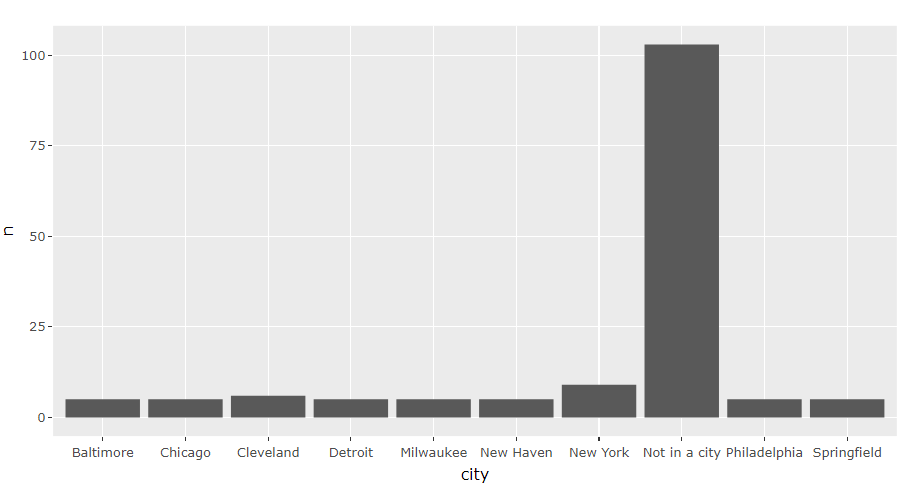
1. **Report if there is an issue with the datasets.**

I found that there were about 366 cities that were different. Then I check intersections and how many cities are common in both and I found that 63 cities were common.

This shows that some cities were missing as the actual dataset has about 607 total distinct cities but the model shows only 366 cities. If used in prediction it will bias the result. **Annex-6**

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. **Propose a meaningful way to bin the types of cities.**



I used the case when the function in mutates of dplyr library. **Used Annex-7.**

Created a two-bin one not in the city and one in the city

**Build a recipe or pipeline using the *recipes* package using the following steps on the training set:**

* After doing binning of cities. Again used train and test method to split the dataset into 2:3 ratios to remove biasess from the model.
* **This dataset used value as its outcome variable as value is the most important continuous feature of the dataset. Then followed 8-15 steps as mentioned.**

1. **Given the 'value' variable the outcome role**
2. **Given the 'id' variable a role of the 'id variable'**
3. **Given the 'fips' variable a new role of the id variable that is 'county id' as it is also an id variable only**
4. **Created a dummy variable with one-hot encoding using the function step\_dummy() in the recipe. Passed variables state, county, city, zcta to be converted into dummy variables using one-hot encoding.**
5. **Added step\_normalize() for all predictors to normalize the data.**
6. **Have a lot of variables that have a high correlation. This issue can be solved using a recipe. Used step\_corr() to remove variables with high correlation. Note exclude 'CMAQ' and 'aod' variables from correlation checking as we need these variables for prediction.**
7. **Next removed variables with nonzero variance. These are the variables that have very similar values. This can be done by adding step\_nzv() in the recipe. This also excludes 'CMAQ' and 'aod' variables.**
8. **Used prep() and bake() to apply this recipe to the training and testing dataset and extract the transformed training and testing dataset. Used prep() for training dataset and retain= TRUE to extract the training dataset with bake().**

**Note: Followed the same step from 8-15 on tests set and got 292 number of observation with 38 numbers of columns. Used Annex-8**

|  |
| --- |
| **GLIMPSE OF TRAIN DATASET AFTER APPLYING 8-15 STEP** |
|  |
| **GLIMPSE OF TEST DATASET AFTER APPLYING 8-15 STEP** |
|  |

**Dimension of both dataset is**

|  |
| --- |
|  |

**STAGE 3 – BUILDING MODELS FOR PREDICTION AND CLASSIFICATION**

1. **MODEL-1 LINEAR REGRESSION MODEL**

In linear regression checked that value is the dependent which will predict the annual average air pollution concentrations of various US cities, whereas value was normalized using the recipe. **Annex-9.**

**Relationship between Predictors and relationship model:** In the model, the value of F =10.39 which is far greater than 1, shows that there can be a relationship between predictors and response variables (Value).

**To check the significant relationship of each predictor:** If the variables have p values less than 0.05, then it show that a variable have a significant relationship. And imp\_a500, imp\_a1500 0 and many more variables is less significant because the p-value for that is greater than 0.05. So, we can remove that variable in further findings while making a new model.

**Model fit:** Multiple R squared values show the variation by the model. R square value is 39% which shows that there is a 39% variation between dependent and predictors.

The residual standard error measures the deviation from the regression line which is about 2.046.

.

|  |
| --- |
|  |
| |  |  | | --- | --- | |  |  | |

**Where the coefficients of the variables are as explained and provided in the table.**

**Where the mean absolute error is 1.49.** This implies that, on average, the difference between our model's predictions and the true quality score is about 1.49.

Whereas, the Root Mean squared error is 2.04. Which tells about residuals how far they are from the regression line or best fit line.

1. **MODEL-2 LINEAR DISCRIMINANT ANALYSIS**

**Using Annex- 10, Firstly created an AQI category from the value as provided in the link. And the variable has been linked with the recipe model for further analysis.**

**By using Annex-10**, performed Linear Discriminant Analysis. Where there were certain variables'id','fips','value','hs\_orless','hs','nohs','log\_prisec\_length\_10000','log\_nei\_2008\_pm10\_sum\_15000' which were collinear removed from the dataset to perform the analysis.

* **Prior probabilities of groups:** it tells about the proportion of the training dataset. For example, there are 70% of the training observations have good air quality, and 29% of training observations show moderate Air quality.
* **Group means:** it tells about mean in Good and Moderate Air quality index.
* **LDA Equation:**

LD1 = 0.072\*lat + ….. -0.10\*City not in a city

Moreover also did prediction on the test dataset and checked the confusion Matrix of the dataset and found that the accuracy rate of the model for predicting the annual average air pollution concentrations for US cities is 71%.

The ROC curve demonstrates the trade-off between sensitivity and specificity. Classifiers that produce curves closer to the top-left corner perform better. A random classifier is supposed to deliver points along the diagonal as a baseline. The closer the curve gets to the ROC space's 45-degree diagonal, the less accurate the test is.

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

1. **MODEL-3 RANDOM FOREST**

Random Forest method classification technique is applied to the dataset when the response variable is categorical moreover, and also can reduce the variance by building a random forest model.

Out of box estimate error is 19.86%. And when checking the accuracy we found that the accuracy of the model is 79%.

VarImp() Plot was created to determine which variables are most significant in determining or predicting the annual average air pollution concentrations. **Using Annex-11**

|  |
| --- |
|  |
|  |
|  |

**Note:**

In comparison to model 1 found the Accuracy rate of model-3 which is a random forest has an accuracy rate of 79%. This means the higher the accuracy of the model the better the model is for the prediction of the annual average air pollution concentrations.

Whereas, the Model-1 linear regression has about a 39% accuracy rate and the Model-2 Linear discriminant model has a 71% of accuracy rate.

After checking the RMSE of all the three models, we found that the RMSE of one model3 is lower than model1. As Root mean square criteria is that if the model has lower RMSE then it will give better performance. Hence Model3 gives better performance than Model2 and Model1.

1. **Make conclusions related to pros and cons of predictive techniques and classifications.**

**PROS OF LINEAR DISCRIMINANT ANALYSIS**

* It still beats some algorithms (logistic regression) when its assumptions are met

**CONS OF LINEAR DISCRIMINANT ANALYSIS**

* It is suitable when it contains two or more categories
* It requires normal distribution of the variables

**PROS OF RANDOM FOREST**

* It works well with the non-linear dataset
* It gives better accuracy than another classification algorithm

**CONS OF RANDOM FOREST**

* It is not used in linear methods
* It creates biasness when dealing with categorical variables**.**

**PROS OF LINEAR REGRESSION**

* It is fast than other models, not create complex calculations. Moreover works well on a large amount of dataset.
* It is the simplest linear equation that tells the relationship between dependent and independent variables.

**CONS OF LINEAR REGRESSION**

* The linear regression model is far too simplistic to account for real-world complexities.
* Outliers create an effect on output, as the best fit line tries to minimize mean squared error for the outliers as well.

**Overall Pros:**

All these modelling techniques are very helpful to know the in-depth analysis of any dataset. Moreover, it can also help in identifying the risks..

**Overall Cons:**

If the dataset is small, it will create a problem, as the algorithm fails to consider which variable is important.

1. **Make conclusions on the *rsample* and *recipes* packages to answer the reason as to why these 2 packages are used.**

**Rsample Package:**

* The package has been used to split the dataset into train and test. Where initial split is used to create a single binary split of the dataset into training and testing datasets.
* The initial split() function tells us what rows of data frame should be assigned to train and test. It does not split the dataset

**Recipe Package**

* It is used to prepare the dataset for data analysis.
* Because after splitting, we checked if the dataset can be used in the model.
* It is standardized for the sequence of steps used to pre-process the dataset.
* Creating a recipe specifies how to create a data frame of predictors - it specifies which variables to use and the pre-processing steps, but it does not execute these steps or create the data frame of predictors.

To illustrate:

Id is not introduced in the model because it contains county numbers and the number of particular monitoring stations assigned to them in the model will create noise.

1. **Make recommendations for other experts in the data analytics field as to what lessons have been learnt to attain a high prediction rate.**

**Recommendation:**

* The main thing is to explore the dataset before starting the analysis.

**To illustrate**: which variables are significant for the dataset, details about the dataset, and also the most important thing which variable we are going to predict as it will be our output variable.

* Normalize the dataset, as most of the time; the dataset is skewed, which shows that the dataset has outliers.
* It is important to check the correlation also in the dataset. To check if there exists a correlation in any variable.

**To illustrate**: If we are going to use linear regression then we can run into the problem of multi-collinearity, and we do not want redundant variables as they create noise in our dataset.

* Check the modelling techniques which suit the model best.

**To illustrate** in this dataset applied three techniques linear regression, Linear Discriminant analysis. By checking the accuracy and RMSE found that random is the best method as its accuracy rate is high and has low RMSE from three models.

**R-CODE**

**#LIBARIES**

library(tidyverse)

library(readr)

library(plotly)

library(skimr)

library(dplyr)

library(ggcorrplot)

library("writexl")

library(tidymodels)

library(recipes)

library(vip)

df <- read\_csv("pm25\_data.csv")

View(df)

**ANNEX-1:**

#structure of data

str(df)

#checking missing value

missing <- sapply(df,function(x)sum(is.na(x)))

data\_missing <- data.frame(missing)

View(data\_missing)

#converting three variables into factor

df <- df %>% mutate(id = as.factor(id)) %>%

mutate(fips = as.factor(fips)) %>%

mutate(zcta = as.factor(zcta))

#checking again three variables

str(df)

df\_skimmed <- skimr::skim(df)

df\_skimmed1 <- data.frame(df\_skimmed)

View(df\_skimmed1)

skim(df)

#bar plot for categorical values

g <- ggplot(df, aes(x = state)) + geom\_bar()

ggplotly(g)

p <- ggplot(df, aes(x = county)) + geom\_bar()

ggplotly(p)

q <- ggplot(df, aes(x = city)) + geom\_bar()

ggplotly(q)

**#ANNEX-2**

df\_state <- unique(df$state)

df\_state

**#ANNEX-3**

df\_stations\_per\_city = df %>% count(city)

View(df\_stations\_per\_city )

#can change slicer accordingly

g <- df\_stations\_per\_city %>%

arrange(desc(n)) %>%

slice(2:10) %>%

ggplot(., aes(x=city, y=n))+

geom\_bar(stat='identity')

ggplotly(g)

**#ANNEX-4**

corr <- round(cor(df %>%select\_if(is.numeric),method = "pearson"), 2)

ggcorrplot(corr, method = "square", type = "upper", ggtheme = ggplot2::theme\_minimal, title = "Correlation plot",

show.legend = TRUE, legend.title = "Correlation Plot", show.diag = FALSE,

colors = c("blue", "white", "red"), outline.color = "Black",

hc.order = FALSE, hc.method = "complete", lab = FALSE,

lab\_col = "black", lab\_size = 4, p.mat = NULL, sig.level = 0.05,

insig = "pch", pch = 4, pch.col = "black",

pch.cex = 5, tl.cex = 12, tl.col = "black", tl.srt = 45,

digits = 2)

df\_cor <- data.frame(corr)

View(df\_cor)

write\_xlsx(df\_cor,"df\_cor.xlsx")

**#ANNEX-5**

#splitting the data

set.seed(123)

split\_df <-initial\_split(df, prop = 2/3)

split\_df

train<-training(split\_df)

dim(train)

test<-testing(split\_df)

dim(test)

#checking the proportion of cities

train\_cities <- train %>% distinct(city)

dim(train\_cities)

test\_cities <- test %>% distinct(city)

dim(test\_cities)

a <- train %>%

count(city) %>%

mutate(prop = n/sum(n))

View(a)

b <- test %>%

count(city) %>%

mutate(prop = n/sum(n))

View(b)

**#ANNEX-6**

# different cities

dim(setdiff(train\_cities, test\_cities))

# cities that overlap

dim(intersect(train\_cities, test\_cities))

n\_unique(df$city)

**#ANNEX-7**

g <- df\_stations\_per\_city %>%

arrange(desc(n)) %>%

slice(1:10) %>%

ggplot(., aes(x=city, y=n))+

geom\_bar(stat='identity')

ggplotly(g)

df\_bin <- df %>%

mutate(city = case\_when(city == "Not in a city" ~ "Not in a city",

city != "Not in a city" ~ "In a city"))

str(df\_bin)

#**ANNEX-8 TRAIN SET AND TEST**

set.seed(1234)

df\_bin\_split <-initial\_split(df\_bin, prop = 2/3)

df\_bin\_split

df\_bin\_train <-training(df\_bin\_split)

df\_bin\_test <-testing(df\_bin\_split)

View(df\_bin\_test)

library(recipes)

rec <-df\_bin\_train %>%

recipe(value~.) %>%

update\_role(everything(), new\_role = "predictor") %>%

update\_role(value, new\_role = "outcome role") %>%

update\_role(id, new\_role = "id variable") %>%

update\_role("fips", new\_role = "county id") %>%

step\_dummy(state, county, city, zcta, one\_hot = TRUE) %>%

step\_corr(all\_numeric(), - CMAQ, - aod)%>%

step\_nzv(all\_numeric(), - CMAQ, - aod) %>%

step\_normalize(all\_predictors())

a <- prep(rec, retain = TRUE)

preproc\_train <- bake(a, new\_data = NULL)

glimpse(preproc\_train)

preproc\_test<- bake(a, new\_data = df\_bin\_test)

glimpse(preproc\_test)

dim(preproc\_train)

dim(preproc\_test)

#**ANNEX-8 TEST SET**

library(recipes)

rec\_test <- df\_bin\_test %>%

recipe(value~.) %>%

update\_role(everything(), new\_role = "predictor") %>%

update\_role(value, new\_role = "outcome role") %>%

update\_role(id, new\_role = "id variable") %>%

update\_role("fips", new\_role = "county id") %>%

step\_dummy(state, county, city, zcta, one\_hot = TRUE) %>%

step\_corr(all\_numeric(), - CMAQ, - aod)%>%

step\_nzv(all\_numeric(), - CMAQ, - aod) %>%

step\_normalize(all\_predictors())

a\_test <- prep(rec\_test, retain = TRUE)

#preproc\_test <- bake(a\_test, new\_data = NULL)

#glimpse(preproc\_test)

dim(preproc\_test)

**#ANNEX-9 LINEAR REGRESSION MODEL**

model1 <- preproc\_train %>% dplyr::select(-c('id','fips')) %>%

lm(value~.,data=.)

summary(model1)

p1 <- predict(model1, preproc\_test)

p1

MAE\_val =MAE(p1 ,preproc\_test$value)

MAE\_val

RMSE\_val = RMSE(p1 ,preproc\_test$value)

RMSE\_val

#**ANNEX-10 LINEAR DISCRIMINANT MODEL**

df\_bin\_train <- df\_bin\_train %>% mutate(AQI\_Category= case\_when(value >= 150.5 ~ 'Very unhealthy 201 to 300',

value >= 55.5 ~ 'Unhealthy 151 to 200',

value >= 35.5 ~ 'Unhealthy for sensitive group 101 to 150',

value >= 12.1 ~ 'Moderate 51 to 100',

TRUE ~ 'Good 0 to 500'

))

df\_bin\_train = df\_bin\_train %>% mutate(AQI\_Category= as.factor(AQI\_Category))

df\_bin\_test <- df\_bin\_test %>% mutate(AQI\_Category= case\_when(value >= 150.5 ~ 'Very unhealthy 201 to 300',

value >= 55.5 ~ 'Unhealthy 151 to 200',

value >= 35.5 ~ 'Unhealthy for sensitive group 101 to 150',

value >= 12.1 ~ 'Moderate 51 to 100',

TRUE ~ 'Good 0 to 500'

))

df\_bin\_test= df\_bin\_test %>% mutate(AQI\_Category= as.factor(AQI\_Category))

preproc\_train$AQI\_Category <- df\_bin\_train$AQI\_Category

preproc\_test$AQI\_Category <- df\_bin\_test$AQI\_Category

library("MASS")

model2 <- preproc\_train %>% dplyr::select(-c('id','fips','value','hs\_orless','hs','nohs','log\_prisec\_length\_10000',

'log\_nei\_2008\_pm10\_sum\_15000')) %>%

lda(AQI\_Category~.,data=.)

model2

prob <- predict(model2,preproc\_test,type="response")

prob

mean(prob$class ==preproc\_test$AQI\_Category)

confusiontab <- table(prob$class,preproc\_test$AQI\_Category)

sum(diag(confusiontab))/sum(confusiontab)

library(caret)

confusionMatrix(confusiontab)

library(pROC)

roc(preproc\_test$AQI\_Category ~ prob$x) %>% plot(asp = NA)

#**ANNEX-11 RANNDOM FOREST MODEL**

library(randomForest)

preproc\_train$id <- NULL

preproc\_train$fips <- NULL

preproc\_train$value <- NULL

model3 <- preproc\_train %>% randomForest(AQI\_Category~.,data=.)

model3

importance(model3)

varImpPlot(model3)

predValid <- predict(model3, preproc\_test)

confusionMatrix(predValid,preproc\_test$AQI\_Category)

**References:**

1. Understand *rsample* package: <https://www.rdocumentation.org/packages/rsample/versions/0.0.5/topics/initial_split>
2. Understand *recipes* package:

<https://www.rdocumentation.org/packages/recipes/versions/0.1.17/topics/recipe>

<https://recipes.tidymodels.org/reference/>

1. Understand the usage of random forest:

<https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/randomForest>

**THANK-YOU**