**Name: MD Akif Perwez**

**B.Tech (Mechanical Engineering)**

**2+ years of experience as Technical Sales Engineer**

**PGP in Data Science, Machine Learning and Neural Networks from Data Trained in collaboration with IBM.**

Blog

**Temperature Prediction**

**Problem Definition**

**Problem Statement:**

* I got the data set from the github and the data set has some unwanted data. <https://github.com/dsrscientist/Dataset2/blob/main/temperature.csv>

**Impact**

* Lack of outliers and skewness are presents, which may not give us the best r2 score.
* NaN are values are present in the data set, which may create problem while building the model.
* Date columns should be changed as it contains DD/MM/YYYY, which also create problem for model building.

In this project, I have built and evaluated multiple linear regression models using python. I have used sciket-learn to calculate the regression, while using pandas for data management and matplotlib and seaborn as visualization. The data set for this project consists of the very popular Temperature of Korea (Seoul).

**Data Analysis**

I applied the concepts of Machine learning and Data Science to predict the temperature of South Korea (Seoul)

The target variable is of continuous type, which simply define that the problem is of linear regression type.

**Attributes are:**

* Station
* Date
* Present\_Tmax
* Present\_Tmin
* LDAPS\_RHmin
* LDAPS\_RHmax
* LDAPS\_Tmax\_lapse
* LDAPS\_Tmin\_lapse
* LDAPS\_WS
* LDAPS\_LH
* LDAPS\_CC1
* LDAPS\_CC2
* LDAPS\_CC3
* LDAPS\_CC4
* LDAPS\_PPT1
* LDAPS\_PPT2
* LDAPS\_PPT3
* LDAPS\_PPT4
* lat
* lon
* DEM
* Slope
* Solar radiation
* Next\_Tmax
* Next\_Tmin
* **Here my Target Variables are ‘Next\_Tmax’ and ‘Next\_Tmin’**

**Libraries used for Analysis and Model Building**

* **Pandas**
* **Numpy**
* **Matplotlib**
* **Seaborn**
* **Scikit learn**
* **Scipy**

Before doing the Analysis part I have to check the shape of the column.

There are 7725 rows and 25 columns.

In this data set I have 2 target columns i.e. Next\_Tmax: next day maximum temp

Next\_Tmin: next day minimum Temp

Both the target columns are of float type’s value.

**EDA (Exploratory Data Analysis)**

In statistics exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods**.**

Null values are present in all the columns “lat, lon, DEM, Slope, Solar radiation”

This data set contain a column “Date”, which has in the form of Days Months and Years.

**Checking Statistic Description.**

Statistic Description gives us the statistic over view the numeric columns.

As there are 25 columns which seems slightly difficult to check Outliers and skewness and standard-deviation. I had done visualization in next step.

**Uni-Variate Analysis**

By using uni-Variate Analysis to find the Present\_Tmax and Present\_Tmin i.e. 37.6 and 20.0 respectively.

**Bi-Variate Analysis**

* By using Scatter plot, I found that min temp is in the month of August (8th month) which is 12 degree Celsius.
* By using Scatter plot, I found that max temp is in the year of 2016 which is 38 degree Celsius.
* By using Scatter plot, I found that min temp is in the year of 2016 which is 11.3 degree Celsius.

Conclusion:

Here I conclude that 2016 is the hottest and coolest year among all the year.

I used violinplot in “Solar Radiation” column just to analyze in which part the maximum radiation occur, it is in between 5000 to 5750, whereas the maximum radiation is 6000 and minimum radiation is 4250.

Checking Correlation to optimize the Outliers and Skewness as the data set contains 27 columns and 7752 rows which is slightly difficult to predict. So I used boxplot method.

* Outliers Detection by using boxplot: Outliers and nothing but some unnecessary data present in the data set.

As far as I execute the code, I found that there are huge numbers of outliers are present in the data set.

Firstly I used IQR method to remove the outliers. I found that I lost approx. 70% data from the data set, which is obviously! Not a good sign to build a model.

Then I used Zscore to remove the data and I found that 11% is data loss in this method which is also not a good sign to build a model.

* **Skewness: Skewness is measure by the symmetry of distribution. The highest Point of a distribution is its mode. The mode is on the x-axis.**

**Skewness are of two types Right Skewness and Left Skewness.**

**In the temp data set, Skewness is there and of both the types some columns has very high skewness like 5 and 6.**

**Data cleansing**

Firstly I used IQR method to remove the outliers. I found that I lost approx. 70% data from the data set, which is obviously! Not a good sign to build a model.

Then I used Zscore to remove the data and I found that 11% is data loss in this method which is also not a good sign to build a model.

As the data set is containing the skewness I used “Power Transform” method to remove the skewness.

Power Transform method is widely used to remove the skewness as it has only one line of code and can remove all the skewness at one time.

**Conclusion:**

So from the above EDA Outliers and Skewness are present in the data set, I did not remove the Outliers because I am losing more than 10% data,

Which is not recommend as the Outliers theory.

Skewness is removed and data is ready to move into the next step which is our model building.

**Note:** Skewness is removed using “Power\_Transform”. As we have more type of method to remove the skewness like log, boxcox, sqrt ,cbrt. But its better to perform Power Transform method before just to get the result easily, incase if we get the skewness remaining in the columns we used those method to remove it.

**Data Preprocessing**

As found in the data set that we have null values are present we have to fill those null values and make our data looks good while doing model building.

I used Simple-Imputer in this because out of 27 columns 24 columns has the nan values. So in the Simple imputer there is a method called “Most Frequent” it help us to field the nan values very effectively and does save our time to write other program in order to fill it by mean median and mode.

**Feature Engineering:**

Feature engineering is the process of using domain knowledge to extract features from raw data. A feature is a property shared by independent units on which analysis or prediction is to be done. Features are used by predictive models and influence results.

I applied this theory in my ‘Date’ column to extract the days, months and years. And then I drop the ‘Date’ column.

This will help for model building.

**Standard-Scaler:**

Standard Scaler follows the Standard Normal Distribution (SND). Therefore, it makes mean=0 and scale the data to unit Variance

I used it to make the data into Standard normal distribution before building the model. It help us to give best accuracy score. If the data is equally balanced before model building it shows that your all the above Analysis are good and you are in your next step.

**Train Test Split**

Before building the model, we have to do train-test-split, that our target variables are in y group and remaining columns are in x group.

X=df.drop([“Next\_Time”,”Next\_Tmin”,axis=1])

Y=df[“Next\_Time”,”Next\_Tmin”,]

**Finding Best Random State**

After splitting the model, will find the best random state to put those x and y values and build our model.

In my case I got best random state in 9 with 77% accuracy score.

Now we do train\_test\_split and give the test size that this much data go into test and will do prediction on the model building.

I put 30% data into test and remaining 70% into training

**Here I used four method for model building**

1. **Linear Regression**
2. **RandomForestRegressor**
3. **KNeighborsregressor**
4. **DecisionTreeRegressor**

**While making a model we need to follow few steps:**

Suppose x is any model, we will assign this x with any variable suppose Y is the variable.

**Now let’s follow the step:**

* Y= x,where x is the model and y is any variable.
* Y=x.fit(x\_train,y\_train), here we are fitting those values of x and y. As already mentioned above that training part should 70% data. So 70% data will go in it.
* Pred = Y.predict(x\_test), here we are predicting the remaining 30% data.

After all these process, now we can get the score from ‘y\_test’.

* Print(r2\_score(y\_test,pred)

**Output:**

After following the above steps, I got the score of every model.

* In Linear Regression I got the r2 score of **77%**
* In RandomForestRegressor I got the r2 score of **90%**
* In KNeighbors I got the r2 score of **87%**
* In DecisionTreeRegressor I got the r2 score of **77%**

**By seeing the above accuracy score still we are not sure that RandomForestregressor has maximum Accuracy Score and is suitable for the final model.**

**We will to do Cross Validation for all the above model.**

Cross validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a simple parameters called ‘k’ that refers to the number of groups that a give data sample is to be split into. As such, the procedure is often called k-fold cross validation.

* **I used k=5 here**

**After Cross Validation I found that**

* **LR=65**
* **RF=70**
* **Kn=53**
* **DTR=38**

From the above r2 score in both of them, now we have to check the minimum difference.

So I got minimum difference i.e. 12 in Linear Regression.

Now will do the Hyper Parameter Tuning with Linear Regression.

**Hyper Parameter Tuning:**

In machine learning hyper parameter is a parameter whose value is set before the learning process.

It help us to increase the Accuracy score for any model we are making.

For this I used “GridSearchCV”.

Now in hyper parameter tuning we have to define the parameter first in order to check which parameter is the best one.

**Steps do to for hyper parameter tuning**

Creating parameter list to pass the in GriSearchCv

* parameters= {"n\_jobs":(x), where x is any integer values.

"fit\_intercept":["True"]

"criterion":["mess","mae"]}

* GCV=GridSearchCV (LinearRegression(), parameters, cv=5), here GCV is any variable we can use anything as per our ease.
* GCV.fit(x\_train,y\_train), passing the values of x and y.
* GCV.best\_params\_ (This code will give the best parameter from the above)
* lr\_mod= LinearRegression (fit\_intercept="True", n\_jobs=2, normalize=”False”), so here I got ‘True and ‘False’ and ‘2’ as best parameter which will give me the maximum r2 score. **lr\_mod** is the model name.
* lr\_mod.fit(x\_train,y\_train)
* pred=lr\_mod.predict(x\_test)
* print(r2\_score(y\_test,pred)\*100)

**By doing this I got 78% r2\_score.**

**Note:**

Sometimes, the r2 score will as same before the hyper parameter tuning, it doesn’t mean that your model is bad.

We can use this model for prediction.

**Conclusion:**

Here, I have come to the end of the project on the topic “Temperature Prediction”

I tried my best to include all the necessary points that are required related to the given data set.

This project took me through various phases of project development and gave me real insight into the world of Machine Learning and Data Scientist.

After Hyper Parameter Tuning I got the r2 score which is 78% for Linear Regression.

**Saving The Model**

Joblib.dump(lr\_mod,”Advertising.pkl”)

Note: here lr\_mod is my model name

My purpose of choosing this project is to observe the climate changes. As this planet is our motherland and we have to do something to protect this.

By doing all the analysis I come to a point that climate change is real and we may not skip the scenario that is happening all around the world.

**“LET US NOT TAKE THIS PLANET FOR GRANTED, I DO NOT TAKE TODAY AND TONIGHT FOR GRANTED”**

**https://github.com/akifperwez/Project-Evaluaion-Project3/blob/main/Temperature%20Prediction%20Project.ipynb**