WEATHER FORECASTING- IMPLEMENTATION AND ANALYSIS OF DIFFERENT MACHINE LEARNING MODELS

INTRODUCTION

Weather forecasting is the application of science and technology to predict the conditions of the atmosphere for a given location and time. Weather forecasts are made by collecting quantitative data about the atmosphere at a given place and using meteorology to project how the weather conditions will be. Weather forecasting now relies on computer-based models that take many atmospheric factors into account. Human input is required to pick the best possible forecast model (machine learning model), which involves pattern recognition using the model, knowledge of model performance, and knowledge of model biases.This is the main aim of our project is to compare various machine learning models and try to find which model gives the best performance for weather forecasting. This blog contains the data collection, preprocessing of data, the classification models, time series analysis-forecasting and the neural network implemented.The inference is drawn from the implementation of different models and the best suited model is suggested for prediction tasks on real world time series data.

IMPORTANCE

There is a vast variety of end uses to weather forecasts. Weather warnings are important forecasts because they are used to protect life and property.Forecasts based on temperature and precipitation are important to agriculture, and therefore to traders within commodity markets. Weather forecasts are used by utility companies to estimate demand over the coming days. On an everyday basis, many use weather forecasts to determine what to wear on a given day. Since outdoor activities are severely curtailed by heavy rain, snow and wind chill, forecasts can be used to plan activities around these events, and to plan ahead and survive them.Not only weather forecasting is helpful in planning daily activities , it is also useful for marketing and advertising. It is also useful in the aviation sector.The aviation industry is especially sensitive to the weather,accurate weather forecasting is essential.Weather forecasts can also be helpful for online businesses and companies.Companies can use weather forecasts to plan their deliveries and also tell their customers if there will be any additional delay in delivering the products or because of bad weather conditions the customers will have to pay an additional amount.Therefore we can see that weather forecasts has a huge impact ; ranging from our personal lives to economies of countries and companies.

DATASET

We have taken our dataset from Kaggle.The data contains hourly weather data of Delhi.It has actually been collected by using the API of the website “Weather Underground”.

DATA ANALYSIS

- In the data, there are 100990 instances and 20 features.

- The dates range from 01-11-1996 to 24-04-2017 and weather recorded at regular intervals.Hence this dataset can be used as time series data as well as a normal classification problem.
- The column "_conds" has categorical values and represents the weather conditions of Delhi so we will take the dependent variable/ target variable as "_conds" that has been renamed as Conditions". Rest all the features are independent variables and will be used to predict the weather condition.
- When we found the percentage of null values.Features with percentage $>60\%$ are as follows:
  HeatIndex 71.13%
  precipitationType 100.00%
  Wgustm 98.93%
  WindChillm 99.42%

So we will drop columns with a high percentage of null values and handle the null values.

- There are two features- "Conditions" and "WindDirection" that have categorical values like smoke, Haze etc and North, south etc respectively. We will have to replace them with unique numerical values.
- We get the description of the data:- In "temperature" mean value is 25.451 and max value is 90, in humidity mean value is 57.9 and max value is 243. Here we see that the max value of temperature and humidity is unrealistic and hence some outliers are detected.

MODELS IMPLEMENTED

1. Classification models like - SVM, KNN, Random Forest,Decision tree, etc
2. Recurrent Neural networks (LSTM)
3. ARIMA model

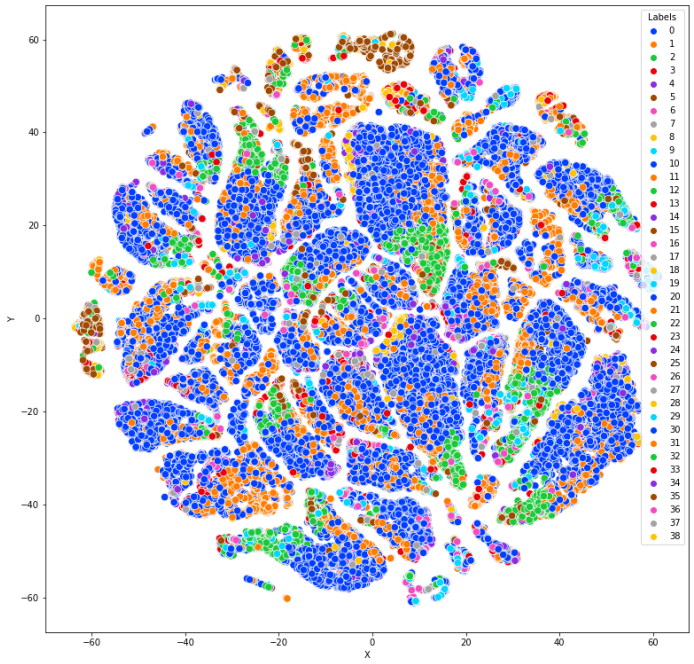
CLASSIFICATION MODELS:

GOAL

Classification based on weather conditions using different classification.

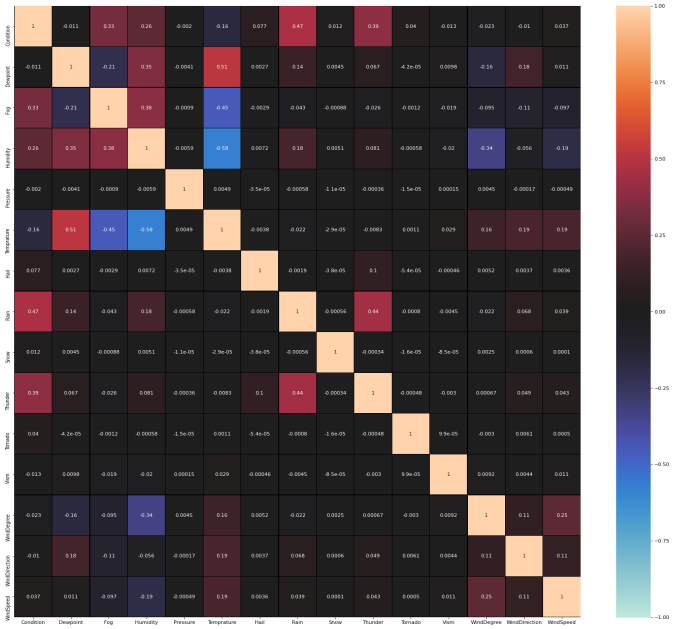
PRE-PROCESSING:

We have visualized the data using TSNE of 2 dimensions, the data is not linearly separable. In the plot we can visualise the 'Condition' column having 39 different classes.



TSNE plot to visualize the data

We have chosen different target values like 'Temperature' and 'Condition' on which we have applied different models. We have chosen different columns as target variable because some columns are correlated with other columns.



Heat map to show correlation

MODELS AND THEIR IMPLEMENTATIONS:

We have splitted the data into 70:30 ratio where 70 is for train set and 30 is for test set. We have used train test split from Sklearn to split the dataset.

(1.) LOGISTIC REGRESSION:

MODEL

Logistic Regression is one of the classification models to predict the probability of a target class. Logistic regression is used for binary classification. But If the data is Multiclass we will use the same idea of binary classification for the Multiclass Classification. This model will predict the class of dependent variables based on independent variables.

IMPLEMENTATION

We have used sklearn library for applying LogisticRegression model on our dataset. In our Model, The target columns we have chosen have multiple classes. So we have to apply 'ovr' or 'multinomial' as multi_class parameter in the logistic regression model. We have set max iter=10000 so that solver may converge easily. This model is not very good for classification as we observed that the data is not separable using these features.

(2.) SUPPORT VECTOR MACHINE:

MODEL

Support Vector Machine can be used in classification. SVM is a supervised learning algorithm. The basic idea of using the SVM on our dataset is to create a decision boundary in n dimensional space of classes. The best decision boundary will be called a hyperplane. As we have visualized the data using TSNE, the data is not linearly separable so we will use Kernel Tricks for the classification. SVM will map our data points into the space and increase the distance between features present in the dataset.

IMPLEMENTATION

We have used the sklearn library for applying the SVC model on our dataset. In our Model,We have applied Grid search CV on the dataset and got the best parameters. We have got the kernel as 'rbf', C=1 and gamma=0.1. Then we have applied the best parameter obtained from the GridsearchCV to our Model.

(3.) K NEAREST NEIGHBOUR:

MODEL

K- Nearest Neighbour can be used for Classification. An object will be classified using the majority voting of its k nearest neighbour. Here K is hyperparameter.This is a non parametric method. It is also an instance based learning algorithm. Here we have not used any underlying function while training. After visualizing data using TSNE we can visualize that there is some region where data points of a particular class are present. So we will be using the K nearest Neighbour on our Dataset.

IMPLEMENTATION

We have used the sklearn library for applying KNeighborsClassifier on our dataset. We applied GridSearchCV for obtaining the best hyperparameter. GridsearchCv returned n neighbours=9 for the dataset. Then we have applied the best parameters on the KNN model using our dataset.

(4.) DECISION TREE:

MODEL

Decision Tree is one of the classification algorithms used to predict the class of unknown datapoints by sorting them down into a Tree based structure from root to leaf. Leaf Node will decide the value of the

class for the prediction. As there are 14 features remaining after applying preprocessing on the dataset.So the decision tree will be a good option. We can predict the class of an unknown datapoint on the basis of traversal in a tree-like structure. The tree is created using the most important features in the dataset. Decision making will be done on the basis of the leaf of the tree.

IMPLEMENTATION

We have used sklearn library for applying on our dataset. We have set the parameters in the classifier as Criterion =’entropy’. We obtained the Decision Tree using sklearn library.

(5.) RANDOM FOREST:

MODEL

The Random forest is also one the classification algorithm. This will be constructed using different structures of randomly selected Decision Tree, The ensemble method is applied on these trees. All the trees present in the Random Forest will predict the class for an instance individual tree. The class which will have the highest number of votes will be known as the class of the instance. As there are 14 features remaining after applying preprocessing on the dataset.So the random forest will be a good option. It will predict the class for the instance on the basis of the majority voting. Our goal is to predict the most accurate class for the datapoint.

IMPLEMENTATION

We have used the sklearn library for applying RandomForestClassifier on our dataset. We have set the parameters in the classifier as n estimators=250, max depth=30 and Criterion as ’entropy’. We obtain the Random Forest Tree using sklearn library.

RESULTS:

’Condition’ as the Target Value:

classification Model	Accuracy
Logistic Regression	0.4847593
Support Vector Machine	0.6359275
K Nearest Neighbour	0.6396797
Decision Tree	0.6814172
Random Forest	0.7564598

’Temperature’ as the Target Value:

classification Model	Accuracy
Logistic Regression	0.0089741
Support Vector Machine	0.3268219
K Nearest Neighbour	0.2751818
Decision Tree	0.8221414
Random Forest	0.6210738

LSTM- Long Short-Term Memory (LSTM) networks

Recurrent neural networks are a great way to capture temporal and seasonal dependencies.But they suffer from vanishing gradient problems. Hence we use LSTM , which because of its gated architecture does not suffer from the vanishing gradient problem and hence is able to capture long term dependencies easily.

GOAL

We try to do a single step prediction using the LSTM. That means that based on the observing temperature for a few days , we try to predict the next day's temperature. The number of days for which we observe the data i.e. the temperature to make a prediction is known as the **time step** . In our approach we have taken the time step to be 7 days. So that means after observing the Temperature for the past 7 days we predict the temperature of the next day.

PRE-PROCESSING THE DATA SET

Since the data set contained hourly weather data for Delhi , we resampled the data according the days and replaced each day's temperature with mean of temperature recorded hourly for that day.This was done using the Pandas resample function.Moreover the Null values were replaced by themean.In this model we have only used the temperature as the feature .All other features were removed.The Train- Test split ratio was set to 90:10 for thismodel.After that since the feature were varying in magni-tude and it is known that neural networks are very sensitive to scaling of data, we scaled the data us-ing a MinMaxScaler of sklearn and scaled all the temperatures between 0 and 1

RE-ENGINEERING THE DATA SET

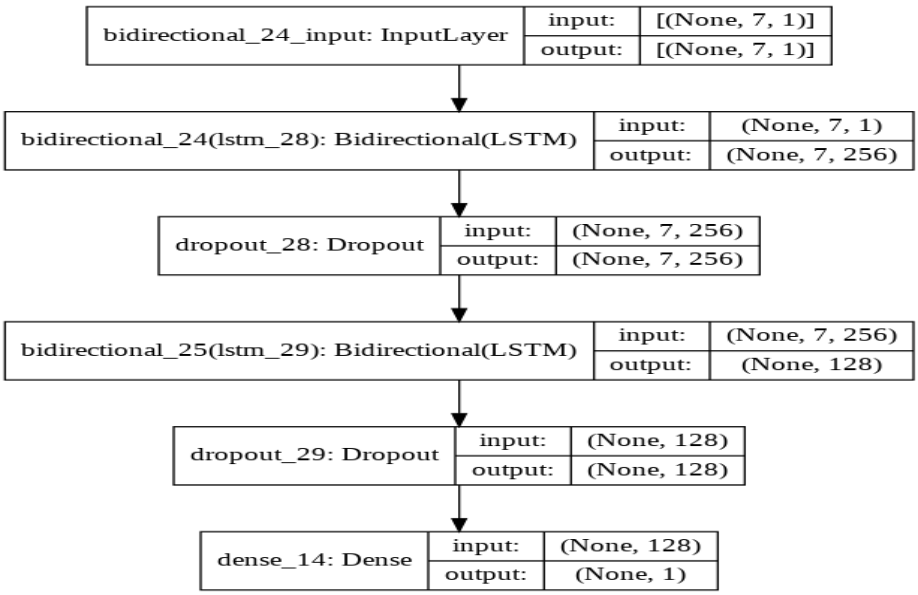
After this we had to create the train and test set for weather prediction. Each instance in both thesets is of the form : Seven days of temperature asX(IE the feature) and the Eight day 's temperature as the prediction(IE the label or Y). That way we had 6725 instances in our training instances and741 instances in our test set.The following image shows an instance from the data set. Note the temperatures have been scaled

```
The temprature are scaled
training set sample :[0.46387643 0.48044345 0.44849278 0.47492111 0.61913303 0.45263453
 0.44730942] and prediction : 0.43341782023786307
```

Model

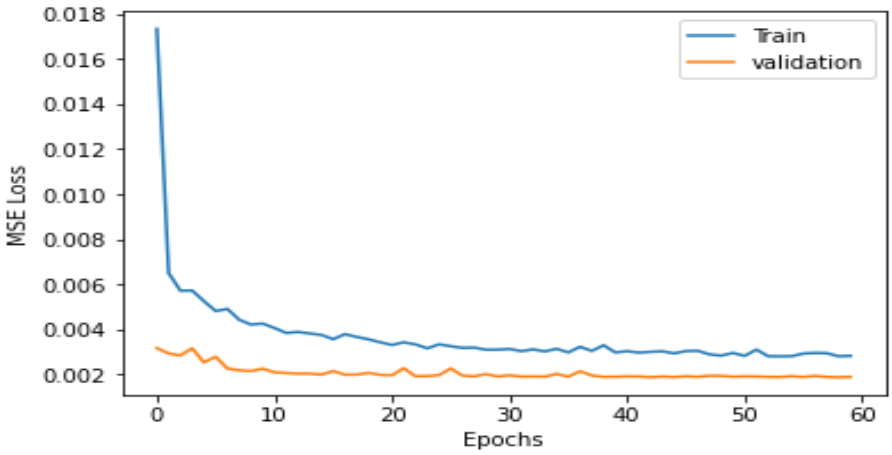
We have used the keras library for LSTM implementations.The Model that we have used contains two blocks of bidirectional LSTM and Dropoutfollowed by a dense layer. The Dropout layer has been added for regularisation and to prevent overfitting.The first bidirectional LSTM contains 128units and the second bidirectional LSTM layer contains 64 units. The Dense layer contains only one unit since it has only one output.The Metric That we use to evaluate the model is Mean squared error Loss function. Moreover the optimizer that we have used for the model is the adam optimizer.The model is visualized below :

```
Model: "sequential_14"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional_24 (Bidirectio (None, 7, 256)            133120
_________________________________________________________________
dropout_28 (Dropout)         (None, 7, 256)            0
_________________________________________________________________
bidirectional_25 (Bidirectio (None, 128)               164352
_________________________________________________________________
dropout_29 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_14 (Dense)             (None, 1)                 129
=================================================================
Total params: 297,601
Trainable params: 297,601
Non-trainable params: 0
_________________________________________________________________
```



TRAINING AND TESTING

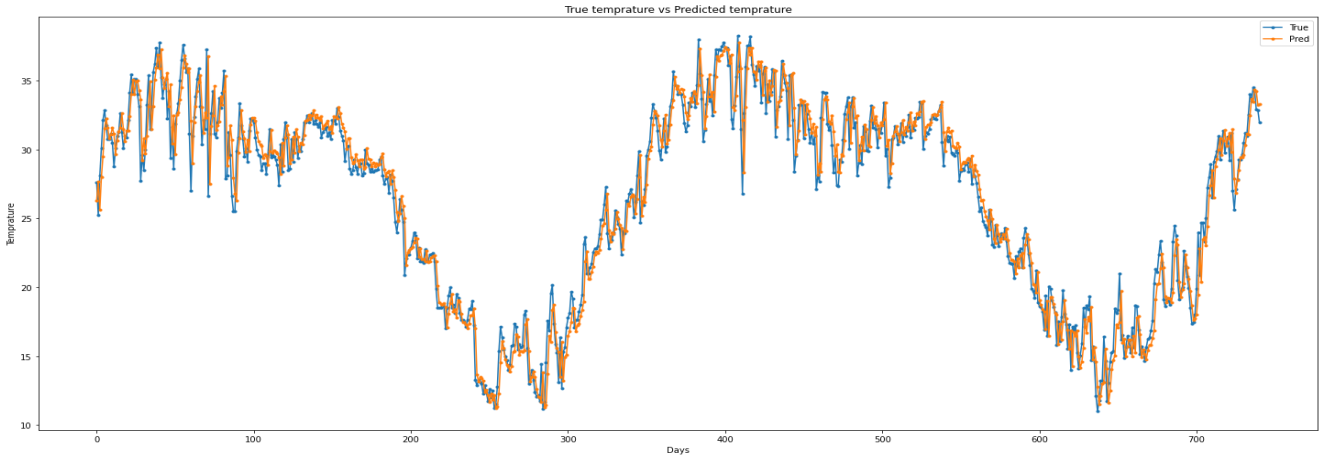
We then trained our model using 60 epochs ,batchsize = 32 and validation split size = 0.1. AfterTraining the model ,the following plot shows the train vs validation loss for 60 epochs.



After that we evaluated the model on the testset using the evaluate function of Keras library.Following image shows the loss that we get on the testset.

```
24/24 [==============================] - 0s 5ms/step - loss: 0.0021
0.002132490510120988
```

As we can see that the mean squared loss for the test set is very low and hence we can deduce that our model is performing/predicting well.The figure plotted below shows what the test data was and what our model predicted.Basically it shows the comparison between test and pred.



For instances we have shown the test data ,its corresponding temperatures and what our models have predicted. It is shown in the figure below.

```
prediction for [25.625 26.    27.125 28.25  24.375 24.75  26.25 ] : 27.625000000000004 and actual prediction 26.31756019592285
prediction for [26.    27.125 28.25  24.375 24.75  26.25  27.625] : 25.250000000000004 and actual prediction 27.677963256835938
prediction for [27.125 28.25  24.375 24.75  26.25  27.625 25.25 ] : 28.125 and actual prediction 25.65427017211914
prediction for [28.25  24.375 24.75  26.25  27.625 25.25  28.125] : 30.125000000000004 and actual prediction 27.998825073242188
prediction for [24.375 24.75  26.25  27.625 25.25  28.125 30.125] : 32.125 and actual prediction 29.51041030883789
prediction for [24.75  26.25  27.625 25.25  28.125 30.125 32.125] : 32.875 and actual prediction 31.529685974121094
prediction for [26.25  27.625 25.25  28.125 30.125 32.125 32.875] : 31.75 and actual prediction 32.25750732421875
prediction for [27.625 25.25  28.125 30.125 32.125 32.875 31.75 ] : 30.75 and actual prediction 31.596433639526367
prediction for [25.25  28.125 30.125 32.125 32.875 31.75  30.75 ] : 30.75 and actual prediction 30.827430725097656
prediction for [28.125 30.125 32.125 32.875 31.75  30.75  30.75 ] : 31.125 and actual prediction 31.0904541015625
prediction for [30.125 32.125 32.875 31.75  30.75  30.75  31.125] : 30.500000000000004 and actual prediction 31.576457977294922
prediction for [32.125 32.875 31.75  30.75  30.75  31.125 30.5  ] : 28.750000000000007 and actual prediction 31.153579711914062
prediction for [32.875 31.75  30.75  30.75  31.125 30.5   28.75 ] : 30.125000000000004 and actual prediction 29.675670623779297
prediction for [31.75  30.75  30.75  31.125 30.5   28.75  30.125] : 31.0 and actual prediction 30.65622329711914
prediction for [30.75  30.75  31.125 30.5   28.75  30.125 31.   ] : 31.250000000000004 and actual prediction 31.29393768310547
prediction for [30.75  31.125 30.5   28.75  30.125 31.    31.25 ] : 32.625 and actual prediction 31.573312759399414
prediction for [31.125 30.5   28.75  30.125 31.    31.25  32.625] : 31.250000000000004 and actual prediction 32.61513137817383
prediction for [30.5   28.75  30.125 31.    31.25  32.625 31.25 ] : 30.125000000000004 and actual prediction 31.49150276184082
prediction for [28.75  30.125 31.    31.25  32.625 31.25  30.125] : 31.0 and actual prediction 30.611568450927734
prediction for [30.125 31.    31.25  32.625 31.25  30.125 31.   ] : 30.875000000000004 and actual prediction 31.35294532775879
```

Moreover our r2 score is 0.9399976005818501 which is quite good for our model.

ARIMA

GOAL

Taking temperature of Delhi (15 years) which is captured at regular intervals and has seasonal patterns and to predict the temperature of the next 1 year of Delhi.

WHAT IS TIME SERIES DATA

It is a collection of data points collected at constant time intervals, some increasing or decreasing trend and some form of seasonality, specific to a particular time frame.

BASICS OF ARIMA

The **A**uto **R**egressive **I**ntegrated **M**oving **A**verage Model is a time series model. It is a standard statistical model whose components are:

- **Autoregression (AR)** - a model of order p shows a changing variable that regresses on its own lagged, or prior, values. If p=3, predictor for X(t) will be:

  $$X(t) = \mu + X(t-1) + X(t-2) + X(t-3) + \varepsilon t$$

  where ε is the error term.
- **Integrated (I)** - represents the differencing that means the data values are replaced by the difference between the data values and the previous values.
- **Moving average (MA)** - refers to a model of order q that incorporates the dependency between an observation and a residual error(generated from a moving average model). If q=3, then predictor X(t) will be:

  $$X(t)= \mu + \varepsilon t + \theta 1.\varepsilon(t-1) + \theta 2.\varepsilon(t-2) + \theta 3.\varepsilon(t-3)$$

  Where error term (ε) is obtained from the difference from past term.

Finding parameters for ARIMA:

- **Autocorrelation Function (ACF)**: It just measures the correlation between two consecutive (lagged versions).
- **Partial Autocorrelation Function (PACF)**: is used to measure the degree of association between X(t) and X(t-p).

- ‘p’ is the point where PACF crosses the upper confidence level, ‘q’ is the point where ACF crosses the upper confidence level, ‘d’tells differences for stationarity.

LOADING THE DATA

To store the time information and perform faster operations, the best library is “Pandas” that has dedicated operations for handling Time Series objects:

- **parse dates**: It specifies the column which contains the date-time information.In this case, its ´datetime utc´ .
- **index col:** The index has to be the variable telling the date-time information.

dtype =’datetime[ns]’ which confirms that it is a datetime object.

PREPROCESSING APPLIED

Feature Engineering

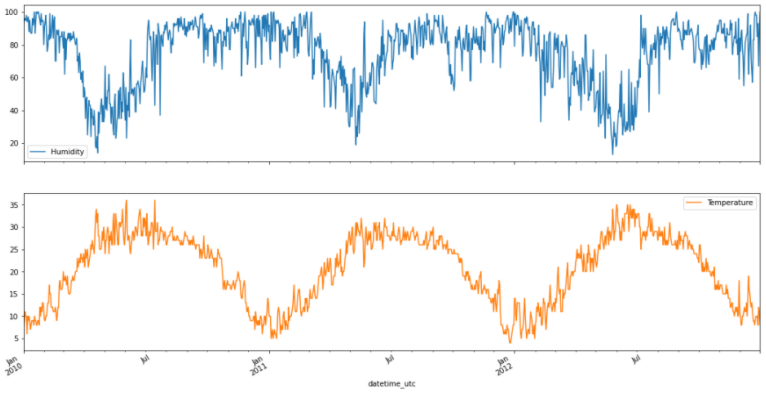
- The important columns are ” conds”,” tempm” and ” hum” so only they are taken and renamed to ”Condition”, ”Temperature” and ”Humidity” respectively.
- Index is of object type which needs to be converted to datetime else we cannot do scaling further.
- Treatment of categorical value - ”Condition”: giving each categorical value, a unique numerical value.

Data Cleaning

- Total Count and percentage of null is found as: Condition: 0.071294%, Humidity: 0.749579% Temperature: 0.666403%. So we filled the previous valid value.
- Outlier : Maximum temperature: 90 and humidity: 243 is unrealistic. So we kept rows with Temperature < 50 and humidity < 100.

VISUALIZATION OF SEASONALITY AND TREND

Plotting the 20 years data is overcrowded so we take data of year 2010 to 2012 to show seasonal change in temperature and humidity.



The figure shows that temperature is high in April to July and low during November to January and accordingly humidity.

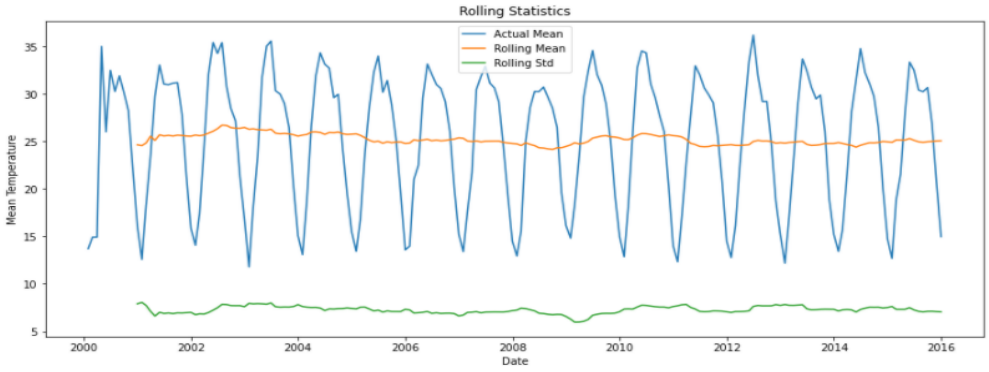
IMPLEMENTATION

Train data= Temperature from year 2000 to 2015.
Test data= Temperature from year 2016 to 2017.

CHECKING STATIONARITY

Generally people assume that the time series data is stationary but we have proven that it has constant statistical properties (Constant mean, Constant variance) over time, using following two methods:

- **Rolling Statistics:** Moving average/standard deviation means that at an instance 't', we have taken average/standard deviation of the last 12 months.
- **Dickey-Fuller Test:** The test's results consist of a "test statistic" and some "Critical Values" for different confidence levels. If the 'Test Statistic' ¡ 'Critical Value', then the time series is stationary



Constant mean and standard deviation

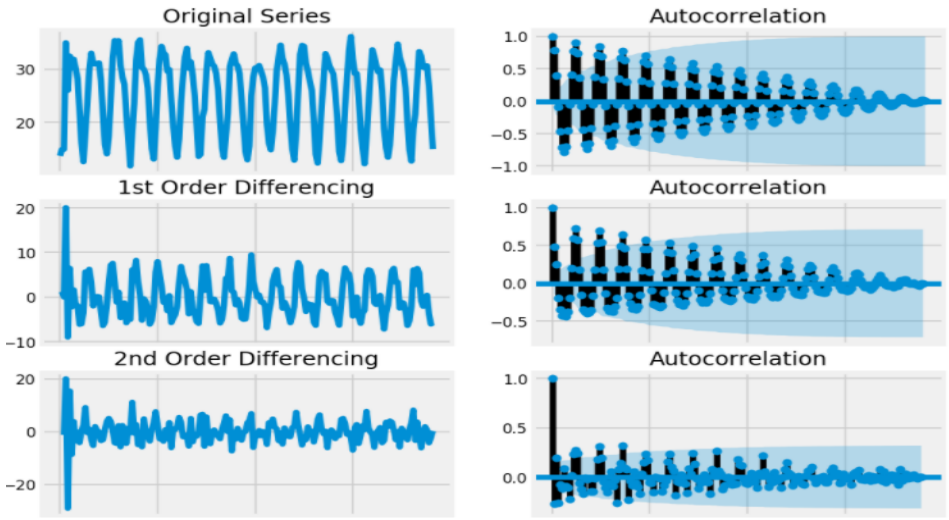
```
Test statistic:  -2.126995250298025
Critical Value (1%)    -3.467420
Critical Value (5%)    -2.877826
Critical Value (10%)   -2.575452
```

Result of dicky-fuller

We observed constant statistics, and our Test statistic is less than Critical Values, so we already have stationary Time series. In the case of non-stationary, techniques used are Decomposing and Differencing.

CHECKING DIFFERENCING

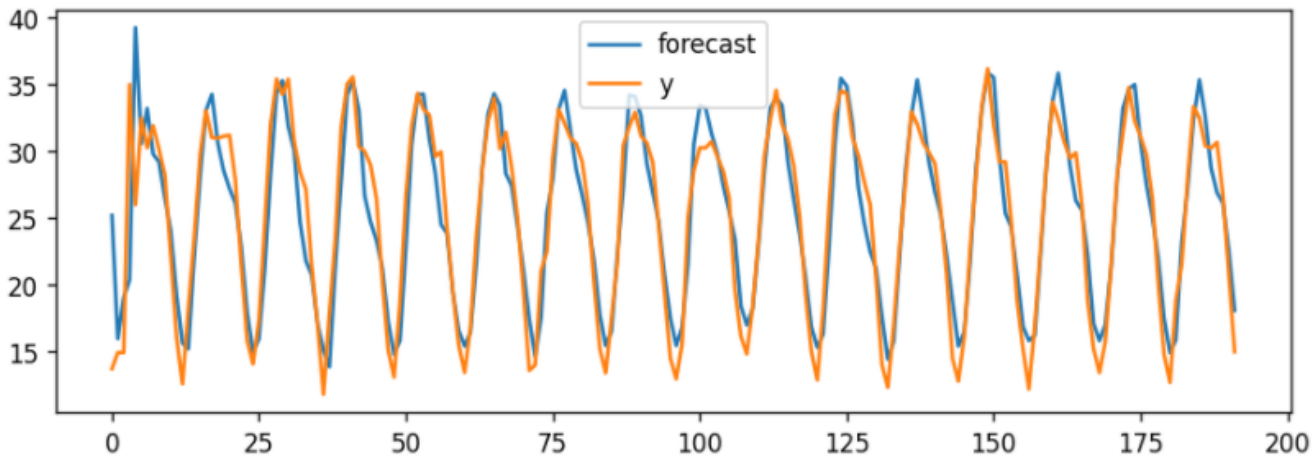
Plotting the original series, the first differencing and the second differencing:



We can see that the original series is perfectly stationary itself, So no need of differencing.
From the confidence intervals, We take p=2, q=2 and d=0, basis of performance comparison = RSS(residual error).

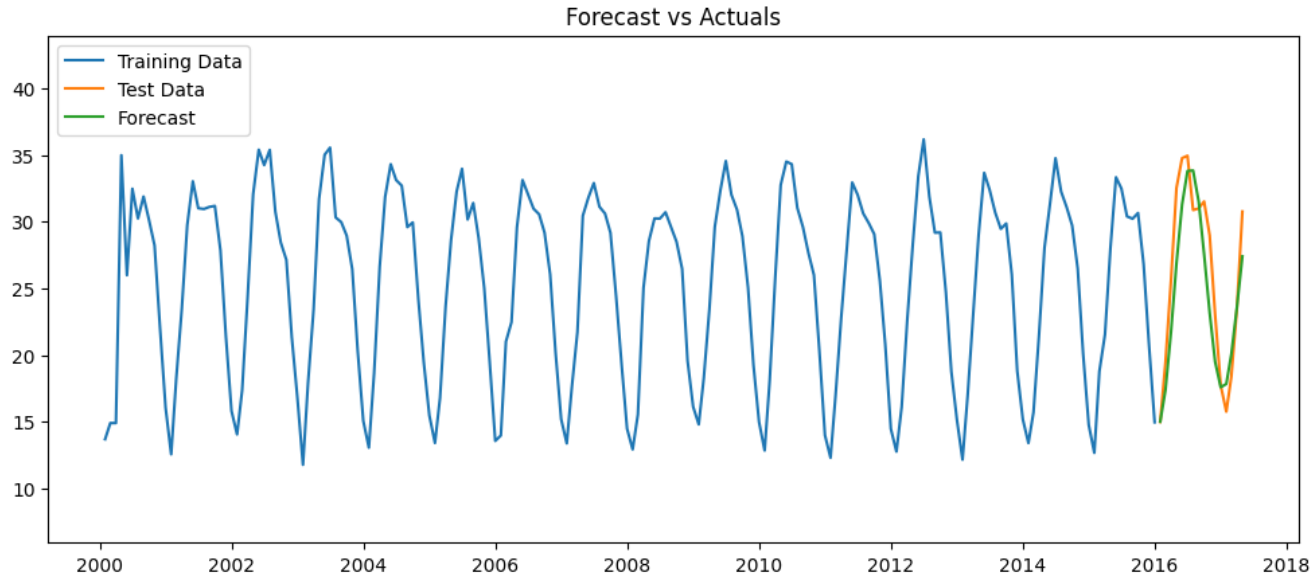
TRAINING THE MODEL

The actual data and the fitted one looks like:



FORECASTING

Using the ARIMA model on the train data of last 15 years,we have forecasted the temperature of next 1 year:



We can see that the prediction for test data replicates the actual test to a great extent. The mean absolute percentage error ie MAPE is 0.94 which makes the accuracy around 90.6%

CONCLUSION AND INFERENCE

1. 20 yrs of Delhi Weather data was collected.
2. Aim was to apply different models to find the forecast of the weather and do classification of data.
3. The classification model gives the following result:

Accuracy for 'Condition' as the Target Value:

- Logistic Regression : 0.4847593
- K Nearest Neighbour : 0.6396797
- Decision Tree : 0.6814172
- Random Forest : 0.7564598

Accuracy for 'Temperature' as the Target Value:

- Logistic Regression : 0.0089741
- Support Vector Machine : 0.3268219
- K Nearest Neighbour : 0.2751818
- Decision Tree : 0.8221414
- Random Forest : 0.6210738

4. The Data is time-series data.
5. The ARIMA Model is implemented with parameters p= 2,d=0 and q=2. The prediction is done for the entire one year from the data of the last 15 years and we can see that in the plot, the forecasted data nearly replicates the actual test data. MAPE value is 0.94, so accuracy can be 90.62%
6. The LSTM model is implemented that gives a r2-score of 0.9399

BLOG AUTHORS AND CONTRIBUTIONS

Aman Dhapola(https://www.linkedin.com/in/amandhapola)

Collection of data from Kaggle, Data Analysis and Preprocessing, Applying Long Short Term Memory(LSTM) Neural Network Model ,Re-engineering data set for LSTM,Visualization of actual vs forecasted Temperature, Conclusion, Literature Survey

Amulya Agrawal(https://www.linkedin.com/in/amulya-agrawal-463623149)

Collection of data, Data Analysis and Preprocessing- Feature Engineering and Data Cleaning, Visualization of seasonality, Checking Stationarity (Rolling Statistics and Dickey-Fuller Test) and Differencing, AutoCorrelation and partial autocorrelation, Finding parameters, Auto regressive integrated Moving Average(ARIMA), Training ARIMA, Forecast, Conclusion, Literature Survey

Vandana Gupta(https://www.linkedin.com/in/vandana-gupta-64728013a)

Collection of data, Data Analysis and Preprocessing, visualization and correlation Heatmap, Dimension reduction and TSNE, **Classification Models- SVM, KNN, Decision trees, Random Forest and Logistic Regression**, Comparison between all the different classification models, Conclusion, Literature Survey

ACKNOWLEDGEMENT

We take this opportunity to thank our Professor **Dr. Tanmoy Chakraborty** and our Teaching Assistant Vivek Reddy for their constant support and guidance throughout this project as part of the Machine Learning(PG) Course 2020.

- Professor: linkedin.com/in/tanmoy-chakraborty-89553324
- Prof. Website: faculty.iiitd.ac.in/~tanmoy/
- Teaching Fellow: Ms. Ishita Bajaj
- Teaching Assistants: Pragya Srivastava, Shiv Kumar Gehlot, Chhavi Jain, Vivek Reddy, Shikha Singh, and Nirav Diwan.

REFERENCES

1. Introduction to Time Series and Forecasting by Brockwell and Davis
2. Weather Forecasting Using Merged Long Short-Term Memory Model (LSTM) and Autoregressive Integrated Moving Average (ARIMA) Model by Afan Galih Salman, Yaya Heryadi, Edi Abdurahman and Wayan Suparta
3. A Prediction of Precipitation Data Based on Support Vector Machine and Particle Swarm Optimization (PSO-SVM) Algorithms by Jinglin Du, Yayun Liu, Yanan Yu and Weilan Yan 1,2
4. C. Kunjumon, S. S. Nair, D. Rajan S., P. Suresh and S. L. Preetha, "Survey on Weather Forecasting Using Data Mining," 2018 Conference on
5. Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, 2018, pp. 262-264, doi:10.1109/ICEDSS.2018.8544326.

6. Poornima, S.; Pushpalatha, M. Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units. Atmosphere 2019, 10, 668.
7. Shivhare, Nikita & Dwivedi, S.B. & Rahul, Atul & Dikshit, Prabhat. (2019). ARIMA based daily weather forecasting tool: A case study for Varanasi.Mausam. 70. 133-140.
8. Kotsiantis, Sotiris & Kanellopoulos, Dimitris & Pintelas, P.(2006). Data Preprocessing for Supervised Learning. International Journal of Computer Science. 1. 111-117.
9. Palomares-Salas, J. C., De La Rosa, J. J. G., Ramiro, J. G., Melgar, J., Aguera, A., & Moreno, A.(2009, May). ARIMA vs. Neural networks for wind speed forecasting. In 2009 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (pp. 129-133). IEEE.
10. Ahmadi, A., Zargaran, Z., Mohebi, A., & Taghavi, F.(2014, July). Hybrid model for weather forecasting using ensemble of neural networks and mutual information. In 2014 IEEE Geoscience and Remote Sensing Symposium (pp. 3774-3777). IEEE.