Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Essentials of Data Analytics - (CSE3506)

Faculty – Lakshmi Pathi Jakkamputi Sir

Lab-3

Harshanth k Prakash
19BCE1293
L21-22 Slot

Tasks for Week-3: Regression and Forecasting on Weather Data
Perform multi-regression and forecasting on weather related dataset "weatherHistory2016.csv"

AIM

To understands Multilinear regression and forecasting on given Weather dataset using R and also verify the null hypothesis.

Algorithm

1. Start
2. Read the dataset as weatherData.
3. Create a data-frame with temperature, app temp, humidity, wind speed and wing bearing columns.
4. Checking the correlation between the column variables with the independent variables.
5. If the value of cor test is >0.5 consider for building the model.
6. Use lm for generating the model.
7. Check the p-value , if <0.05 this model is significant else not.
8. **Next is Forecast** the hours in the given dataset to generate a time series data of temperature.
9. Plot the data.
10. Use adf.test to check stationary of values and auto.arima to generate the data model.
11. Use forecast command to forecast for the next 30 days (24*30) , time series data.
12. Stop.

Statistics

Case 1: Multi linear regression Model.

```
> #Correaltion between dependent and inpedent variable
> cor(input$Temperature,input$ApparentTemperature)
[1] 0.9945785
> cor(input$Temperature,input$Humidity)
[1] -0.6148516
> cor(input$Temperature,input$WindSpeed) #remove from the model
[1] -0.1336635
> cor(input$Temperature,input$WindBearing) #remove from the model
[1] 0.1343402
> cor(input$Temperature,input$Visibility) #remove
[1] 0.4669728
```

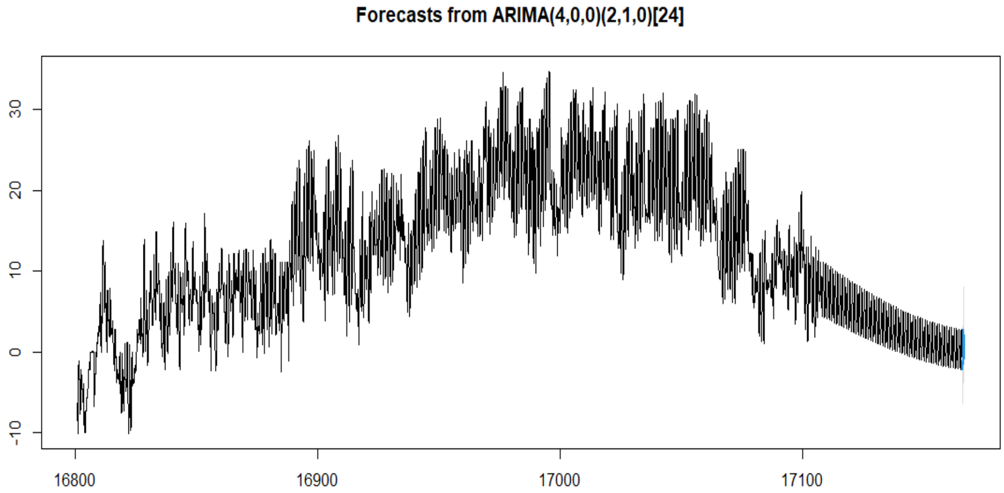
```
> model <- lm(Temperature~Humidity+ApparentTemperature, data =input)
> print(model)

Call:
lm(formula = Temperature ~ Humidity + ApparentTemperature, data = input)

Coefficients:
        (Intercept)             Humidity  ApparentTemperature
             4.5598              -2.1522               0.8477

> summary(model)

Call:
lm(formula = Temperature ~ Humidity + ApparentTemperature, data = input)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6822 -0.4686  0.0962  0.5131  2.0212

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          4.559791   0.363004  12.561  < 2e-16 ***
Humidity            -2.152178   0.396124  -5.433 1.62e-07 ***
ApparentTemperature  0.847662   0.007483 113.276  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8393 on 197 degrees of freedom
Multiple R-squared:  0.9906,    Adjusted R-squared:  0.9905
F-statistic: 1.038e+04 on 2 and 197 DF,  p-value: < 2.2e-16
```

Case 2: Forecasting / Time series data model

```
> data <- ts(data$Temperature..C., start = as.Date("2016-01-01"), end = as.Date("2016-12-31"), frequency = 24)
> adf.test(data)

        Augmented Dickey-Fuller Test

data:  data
Dickey-Fuller = -2.0933, Lag order = 20, p-value = 0.5389
alternative hypothesis: stationary
```

```
> weathermodel = auto.arima(data, ic="aic", trace=TRUE)

 Fitting models using approximations to speed things up...

 ARIMA(2,0,2)(1,1,1)[24] with drift         : Inf
 ARIMA(0,0,0)(0,1,0)[24] with drift         : 44069.51
 ARIMA(1,0,0)(1,1,0)[24] with drift         : 23353.65
 ARIMA(0,0,1)(0,1,1)[24] with drift         : 34905.76
 ARIMA(0,0,0)(0,1,0)[24]                    : 44067.69
 ARIMA(1,0,0)(0,1,0)[24] with drift         : 25336.32
 ARIMA(1,0,0)(2,1,0)[24] with drift         : 22488.23
 ARIMA(1,0,0)(2,1,1)[24] with drift         : Inf
 ARIMA(1,0,0)(1,1,1)[24] with drift         : Inf
 ARIMA(0,0,0)(2,1,0)[24] with drift         : 43570.42
 ARIMA(2,0,0)(2,1,0)[24] with drift         : 22302.42
 ARIMA(2,0,0)(1,1,0)[24] with drift         : 23179.67
 ARIMA(2,0,0)(2,1,1)[24] with drift         : Inf
 ARIMA(2,0,0)(1,1,1)[24] with drift         : Inf
 ARIMA(3,0,0)(2,1,0)[24] with drift         : 22166.34
 ARIMA(3,0,0)(1,1,0)[24] with drift         : 23061.92
 ARIMA(3,0,0)(2,1,1)[24] with drift         : Inf
 ARIMA(3,0,0)(1,1,1)[24] with drift         : Inf
 ARIMA(4,0,0)(2,1,0)[24] with drift         : 22139.26
 ARIMA(4,0,0)(1,1,0)[24] with drift         : 23036.75
 ARIMA(4,0,0)(2,1,1)[24] with drift         : Inf
 ARIMA(4,0,0)(1,1,1)[24] with drift         : Inf
 ARIMA(5,0,0)(2,1,0)[24] with drift         : 22142.04
 ARIMA(4,0,1)(2,1,0)[24] with drift         : 22141.26
 ARIMA(3,0,1)(2,1,0)[24] with drift         : 22143.36
 ARIMA(5,0,1)(2,1,0)[24] with drift         : 22142.46
 ARIMA(4,0,0)(2,1,0)[24]                    : 22137.29
 ARIMA(4,0,0)(1,1,0)[24]                    : 23034.76
 ARIMA(4,0,0)(2,1,1)[24]                    : Inf
 ARIMA(4,0,0)(1,1,1)[24]                    : Inf
 ARIMA(3,0,0)(2,1,0)[24]                    : 22164.37
 ARIMA(5,0,0)(2,1,0)[24]                    : 22140.07
 ARIMA(4,0,1)(2,1,0)[24]                    : 22139.29
 ARIMA(3,0,1)(2,1,0)[24]                    : 22141.4
 ARIMA(5,0,1)(2,1,0)[24]                    : 22140.5

 Now re-fitting the best model(s) without approximations...

 ARIMA(4,0,0)(2,1,0)[24]                    : 22181.95

 Best model: ARIMA(4,0,0)(2,1,0)[24]
```

```
> weatherforecast = forecast(weathermodel, level=c(95), h=24)
> weatherforecast
         Point Forecast     Lo 95     Hi 95
17166.04     -1.1523189 -2.838037 0.5333991
17166.08     -1.4780330 -3.955125 0.9990587
17166.12     -1.8023575 -4.948138 1.3434231
17166.17     -2.0540950 -5.764204 1.6560142
17166.21     -2.1472727 -6.310504 2.0159586
17166.25     -2.0112465 -6.542625 2.5201314
17166.29     -1.6105423 -6.441048 3.2199633
17166.33     -0.9619246 -6.035937 4.1120882
17166.38     -0.1319451 -5.405014 5.1411236
17166.42      0.7694320 -4.666918 6.2057817
17166.46      1.6171292 -3.953613 7.1878717
17166.50      2.2905911 -3.391113 7.9722954
17166.54      2.6988876 -3.074683 8.4724586
17166.58      2.7997565 -3.050054 8.6495669
17166.62      2.6131741 -3.300036 8.5263844
17166.67      2.1986228 -3.767402 8.1646478
17166.71      1.6543712 -4.355715 7.6644578
17166.75      1.0839278 -4.962964 7.1308195
17166.79      0.5654995 -5.512168 6.6431670
17166.83      0.1464187 -5.957005 6.2498427
17166.88     -0.1633210 -6.288317 5.9616750
17166.92     -0.3987896 -6.541864 5.7442848
17166.96     -0.6087477 -6.766980 5.5494850
17167.00     -0.8427709 -7.013719 5.3281773
```




```
> accuracy(weathermodel)
                      ME      RMSE       MAE MPE MAPE      MASE         ACF1
Training set 0.001363075 0.8586022 0.5773539 NaN  Inf 0.2691455 0.0001351266
```

Inference

Case 1: Multi linear regression Model.

The columns Humidity and Apparent Temperature are correlated with the dependent variable Temperature with values 0.99 and -0.67. Hence these values are used for building the multi linear regression model. The p-value of the model is <2.2e-16 which is <0.05, this means we can reject the null hypothesis and accept the linear model for prediction.

Case 2: Forecasting / Time series data model

From the Augmented dickey-fuller test, the p value was found to be 0.01. We can conclude that the given data are stationary because it is less than 0.05. The AIC e valuates all of the models and selects the best one. The best model has been foun d to be ARIMA (4,0,0) (2,1,0) [24].

Program

Case 1: Multi linear regression Model.

```
weather<-read.csv("D:/6th Sem Works/A2- EDA/LAB/Lab3/weatherData.csv");
head(weather)
library(dplyr)
input<-
weather[,c("Formatted.Date","Temperature","ApparentTemperature","Humidity","WindSp
eed","WindBearing","Visibility")]
input=sample_n(input,200)
head(input)
#Correaltion between dependent and inpedent variable
cor(input$Temperature,input$ApparentTemperature)
cor(input$Temperature,input$Humidity)
cor(input$Temperature,input$WindSpeed) #remove from the model
cor(input$Temperature,input$WindBearing) #remove from the model
cor(input$Temperature,input$Visibility) #remove
model <- lm(Temperature~Humidity+ApparentTemperature, data =input)
print(model)
summary(model)
```

Case 2: Forecasting / Time series data model

```
rm(list = ls())
library(dplyr)
library(forecast)
library(tseries)
data <- read.csv('D:/6th Sem Works/A2- EDA/LAB/Lab3/weatherHistory2016.csv')
data <- ts(data$Temperature..C., start = as.Date("2016-01-01"), end = as.Date("2016-12-
31"), frequency = 24)
adf.test(data)
weathermodel = auto.arima(data, ic="aic", trace=TRUE)
weatherforecast = forecast(weathermodel, level=c(95), h=24)
weatherforecast
plot(data)
plot(weatherforecast)
accuracy(weathermodel)
```