Internet of Things Analytics

Advances in Data Science and Architecture

Under the guidance

Professor Srikanth Krishnamurthy

Team 8

- Ankita Suresh Shanbhag
- Vignesh Karthikeyan
- Sriniketan G.S.

Index

Abstract

We have the weather station data from Sensors (the sensor we have chosen is https://www.wunderground.com/), for this analysis we have chosen Boston, MA and weather data from the year 2008 to present. We are going to forecast the weather for the next few days. Our Web application will display the forecast for the future period and will get updated every day.

Most of the cities have a weather station at the Airport but many places like Universities and Government properties have their weather stations too, if they can forecast with their own stations they can feed a more accurate data into their building (services) controls.

We are forecasting the future weather using the past data so we have used AR + I + MA modeling (ARIMA) for the forecast as the future data depends only on the past trend. And we have tried to observe the trend and seasonal components out of our data and the trend represents the gradual increase in temperature over the year, which might be due to global warming.

Our data set has the following components.

Column	Type
Timestamp	Date
EST	character
Max.TemperatureF	numeric
Mean.TemperatureF	numeric
Min.TemperatureF	numeric
Max.Dew.PointF	numeric
MeanDew.PointF	numeric
Min.DewpointF	numeric
Max.Humidity	numeric
Mean.Humidity	numeric
Min.Humidity	numeric
Max.Sea.Level.PressureIn	numeric
Mean.Sea.Level.PressureIn	numeric
Min.Sea.Level.PressureIn	numeric
Max.VisibilityMiles	numeric
Mean.VisibilityMiles	numeric
Min.VisibilityMiles	numeric
Max.Wind.SpeedMPH	numeric
Mean.Wind.SpeedMPH	numeric

Max.Gust.SpeedMPH	numeric
PrecipitationIn	numeric
CloudCover	numeric
Events	character
WindDirDegrees.br...	character

Visualization and Analysis

To forecast the future weather data and understand how it behaves, so beginning with the temperature data over 9 years lets plot it against time.

We will do this in R

Packages you might need:

```
#install.packages("forecast")
#install.packages("xts")
#install.packages("reshape")
library(forecast)
library(xts)
library(reshape)
```

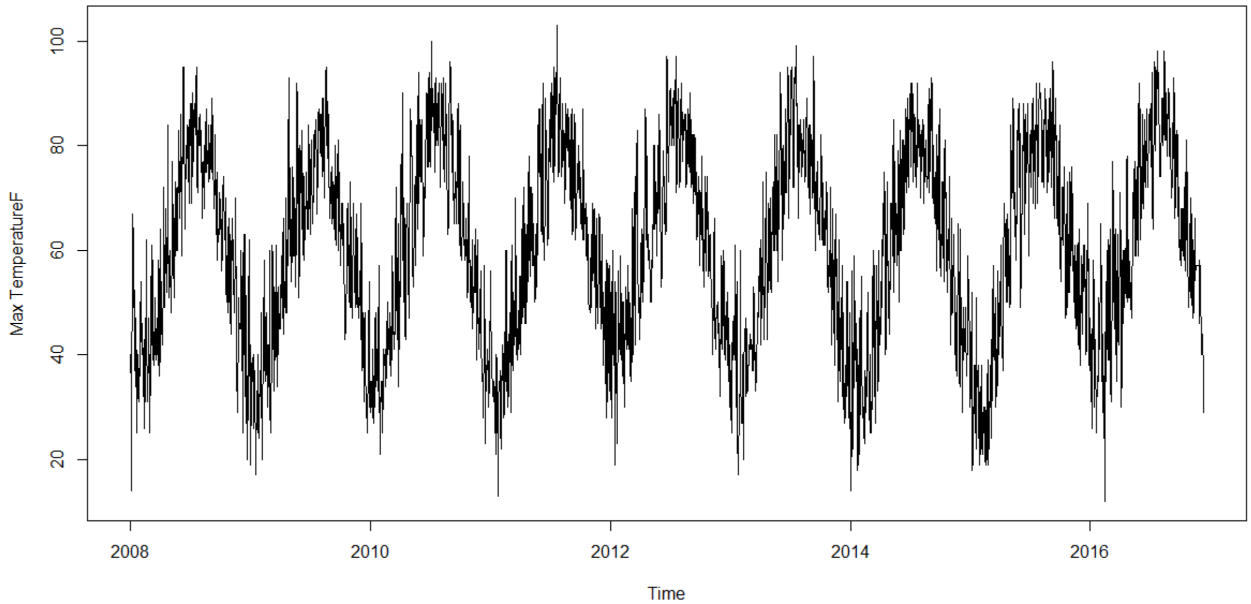
We have the weather data till date from Azure ML(Experiment.csv)

Choose the Max Temperature and create a Time series data set(ZOO)

```

data1<-read.csv("Experiment.csv")
myts<-data.frame(data2$timestamp,data2$`Max TemperatureF`)
ZOO <- zoo(myts$data2..Max.TemperatureF., order.by=myts$data2.timestamp)
```

2.1 Plot the time series dataset

We get the following plot



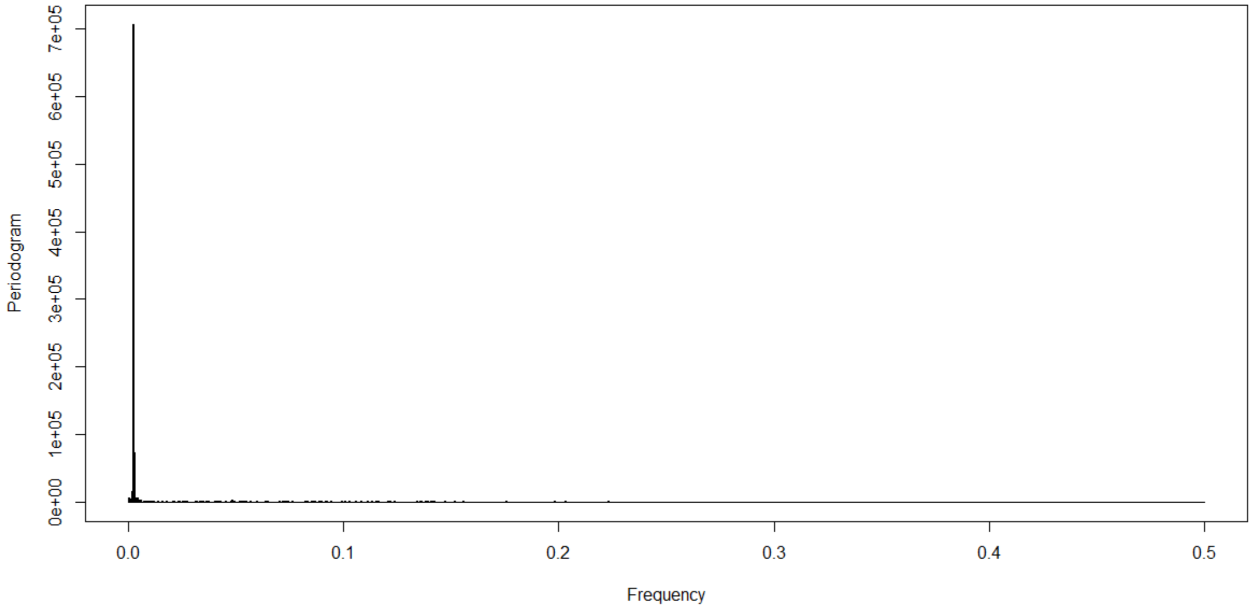
It can be observed that there is a seasonality component to data but the trend is not clearly visible so on decomposing we get.

2.2 Detect the seasonality using Fourier transform

For this data, you can observe that the seasonality is 365/366 days as the temperature varies with respect to the earth's rotation which is a cycle of 365/366 days.

But for rest of the data where it is not clear we can use Fourier transform to get the frequency.

```
# Install and import TSA package
install.packages("TSA")
library(TSA)

# read the data
raw = myts

# compute the Fourier Transform
p = periodogram(raw$data2..Max.TemperatureF.)

```



It can be observed that there is some trend around0.01 frequency

Get the seasonality

```
dd = data.frame(freq=p$freq, spec=p$spec)
order = dd[order(-dd$spec),]
top2 = head(order, 2)

# display the 2 highest "power" frequencies
top2

# convert frequency to time periods
time = 1/top2$f
time
```

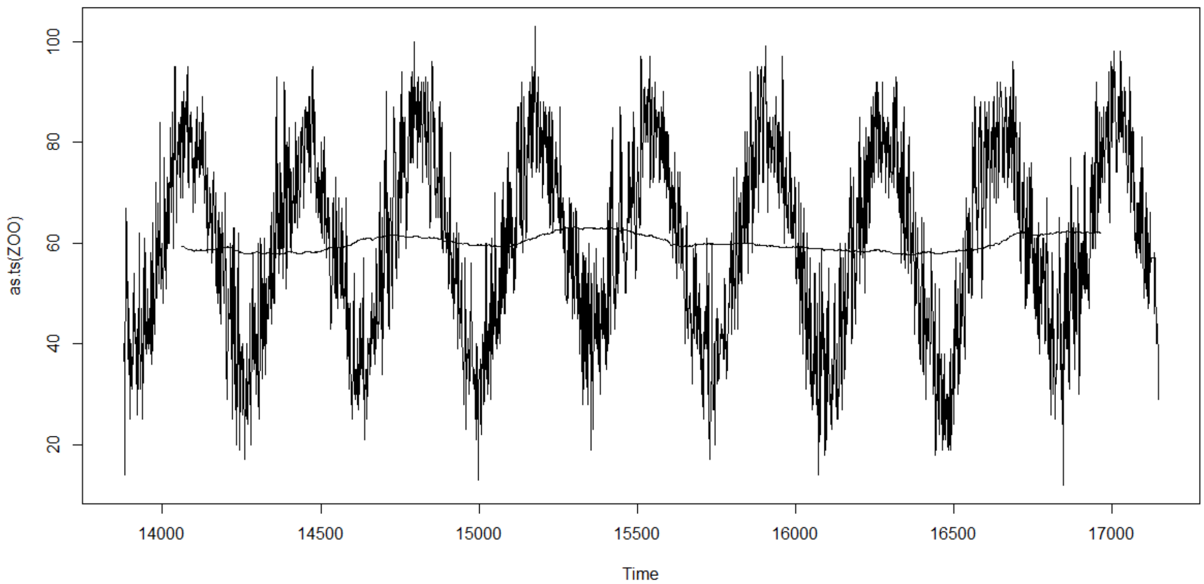
The seasonality is

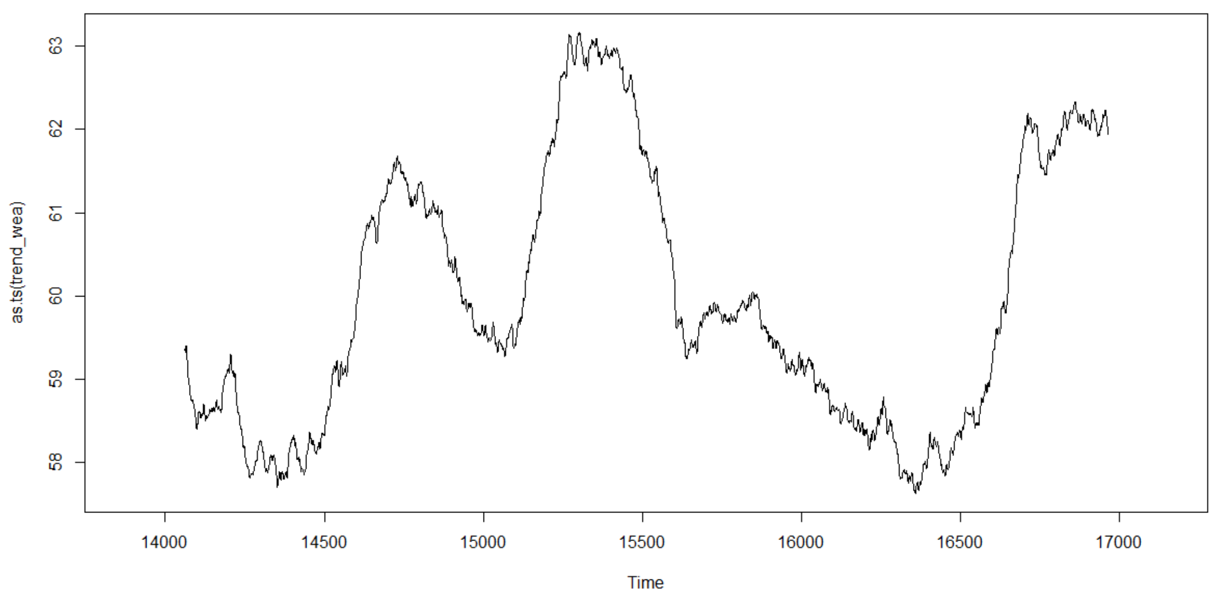
```
> dd = data.frame(freq=p$freq, spec=p$spec)
> order = dd[order(-dd$spec),]
> top2 = head(order, 2)
>
> # display the 2 highest "power" frequencies
> top2
          freq      spec
9  0.002666667 706687.16
10 0.002962963  73593.92
> # convert frequency to time periods
> time = 1/top2$f
> time
[1] 375.0 337.5
```

The average of which is 356.25, but we will use 365 as we know it for this data (which is reconfirmed)

Use moving average windows of 365 as the data is collected for every day for 9 years

```
#detect the trend
trend_wea = ma(zoo, order = 366, centre = T)
plot(as.ts(zoo))
lines(trend_wea)
plot(as.ts(trend_wea))
```





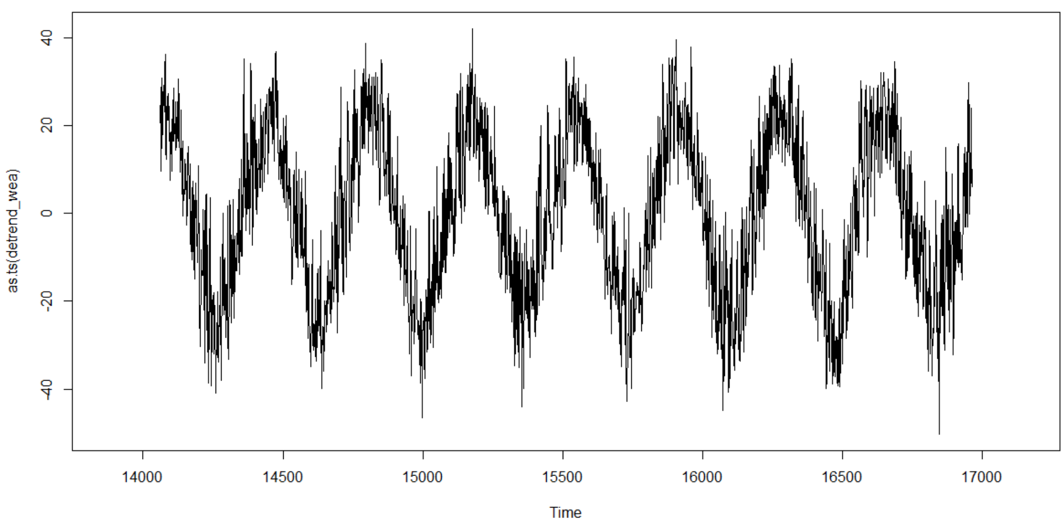
The trend component seems random, but on close inspection you can see the temperature it is varying between is 58F and 61F and only for a period of 9 years, which is not enough the observe a trend of increasing temperature.

We will still remove it from the seasonal data.

From the graph we can observe that the trend and seasonal components are additive.

2.3 DE trend the time series

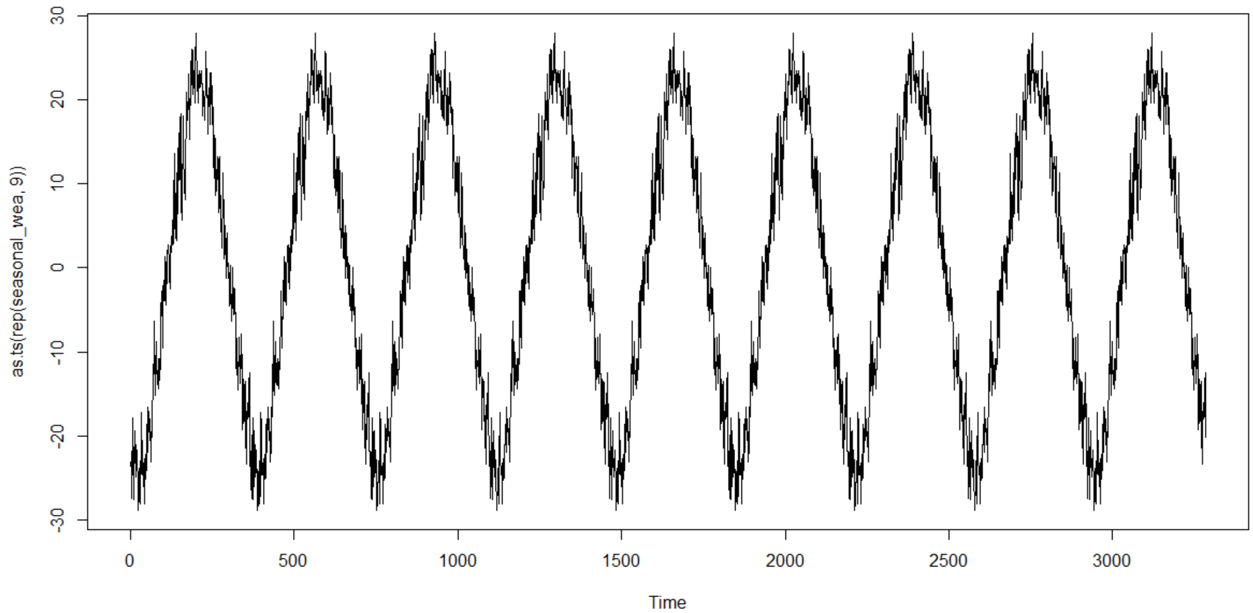
```
#detrend the time series
detrend_wea = zoo - trend_wea
plot(as.ts(detrend_wea))
```



The trend component has been removed from the time series

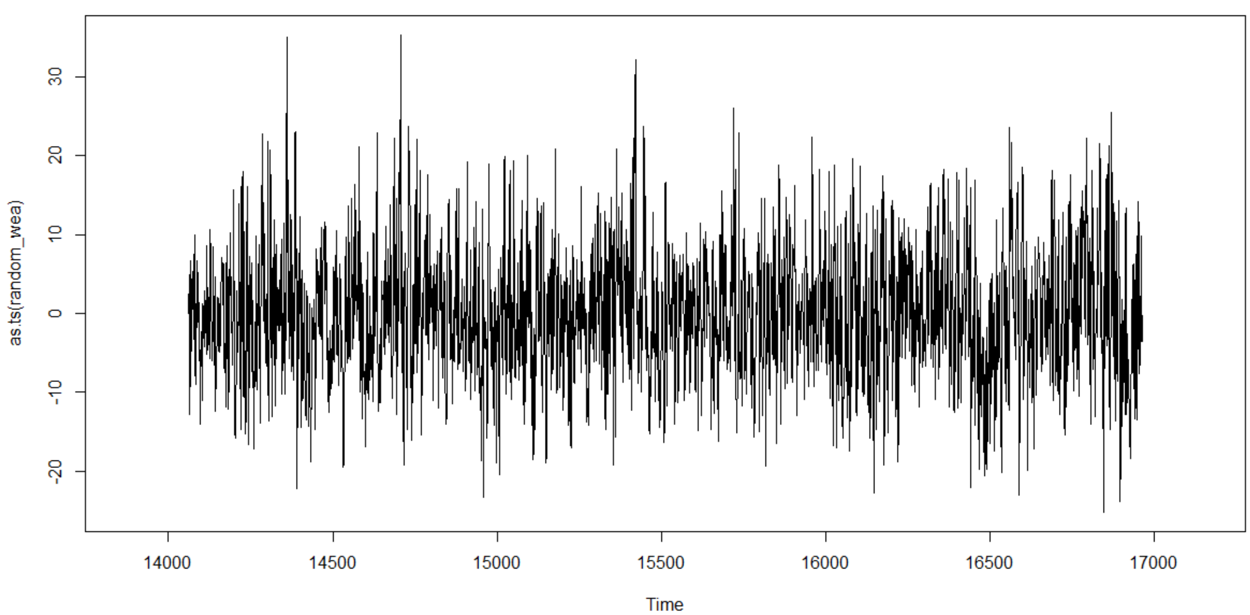
2.4 Average Seasonality

```
#average seasonality
m_wea = t(matrix(data = detrend_wea, nrow = 365))
seasonal_wea = colMeans(m_wea, na.rm = T)
plot(as.ts(rep(seasonal_wea,9)))
```



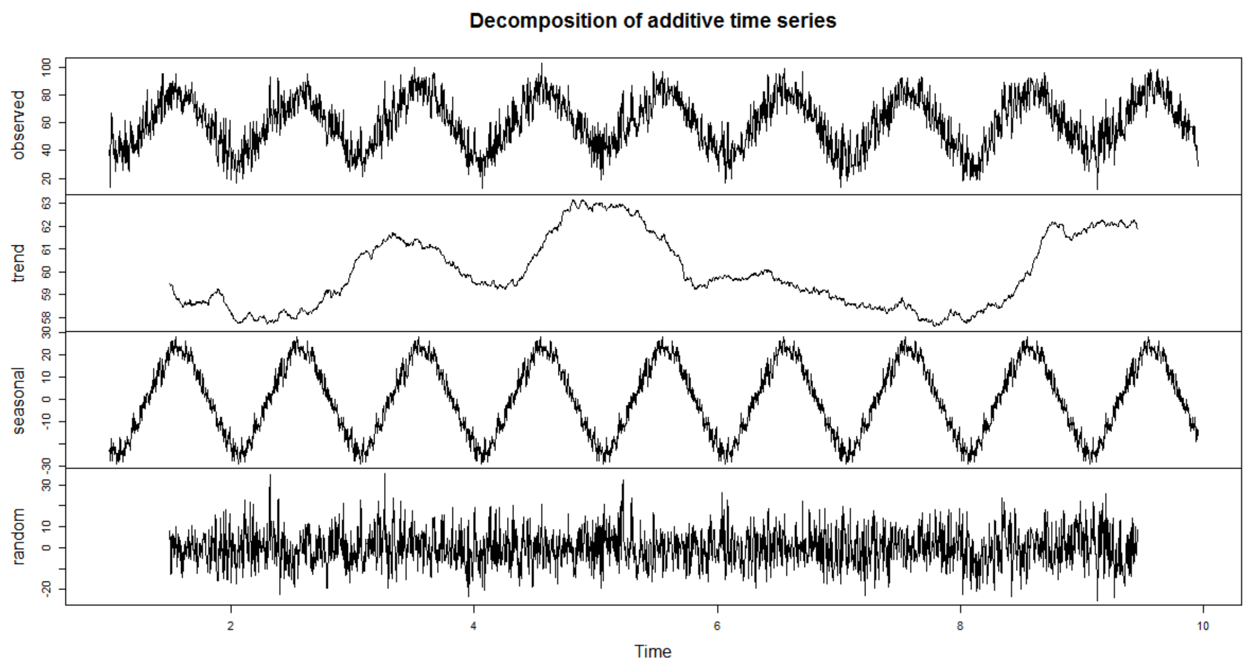
2.5 Random Noise extraction

```
#random noise left
random_wea = zoo - trend_wea - seasonal_wea
plot(as.ts(random_wea))
```



2.6 Decompose

```r

#DECOMPOSE Time series decomposition in R
ts_wea = ts(zoo, frequency = 365)
decompose_wea = decompose(ts_wea, "additive")

plot(as.ts(decompose_wea$seasonal))
plot(as.ts(decompose_wea$trend))
plot(as.ts(decompose_wea$random))
plot(decompose_wea)
```



ARIMA Forecast

Our data is Multivariate Time Series as we have multiple time series dependent variables.

For each Model we will have to convert it to Univariate Time Series for modeling.

```r

data1<-read.csv("Experiment.csv")
myts<-data.frame(data2$timestamp,data2$`Max TemperatureF`)
zoo <- zoo(myts$data2..Max.TemperatureF., order.by=myts$data2.timestamp)

```

Now this data just has Max Temperature.

For our Data we have observed that there is some trend observed and hence we will be using ARIMA model instead of ARMA model as it is not stationary Time Series.

3.1 Estimation and Forecasting using Box-Jenkins (B-J) methodology

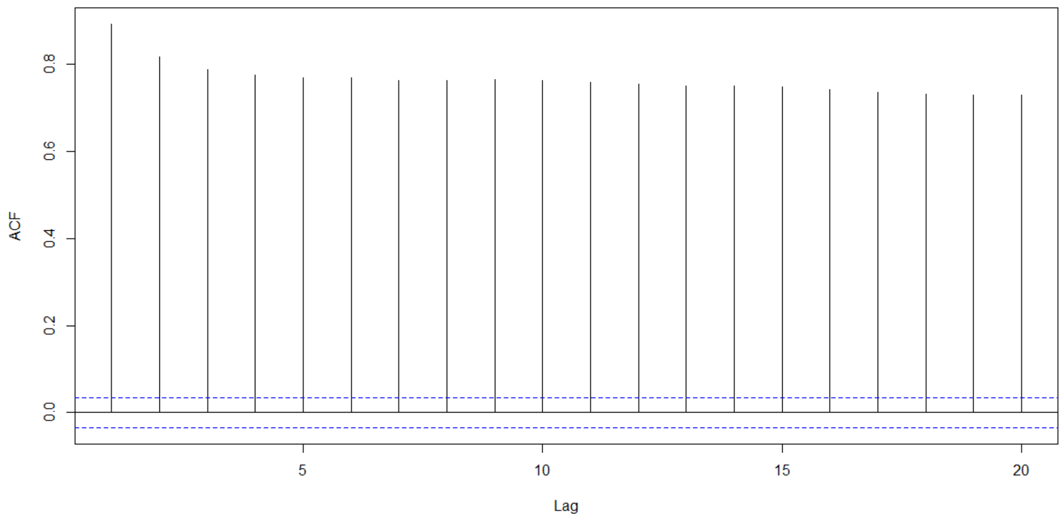
3.1.1 Identification

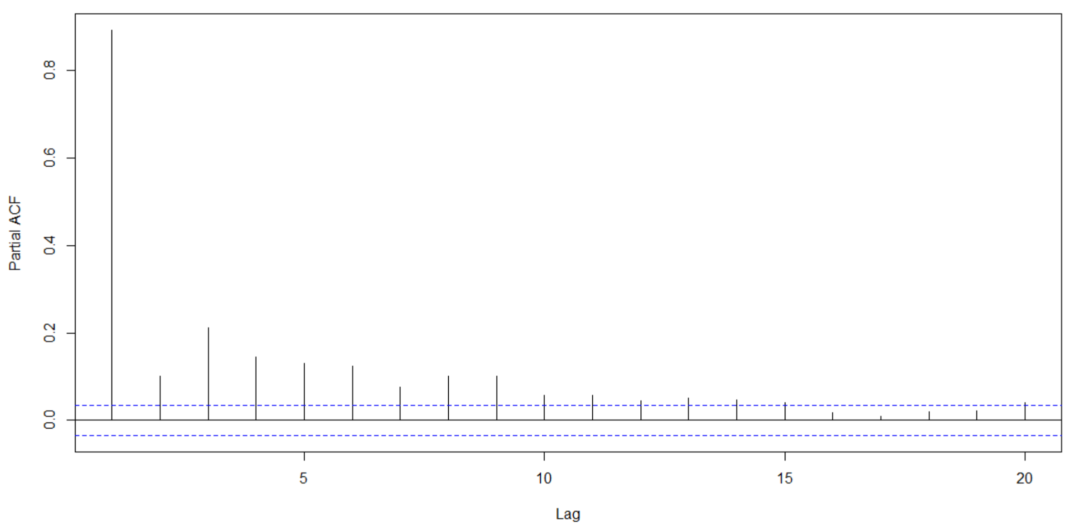
Autocorrelation Function(ACF)

Partial Autocorrelation Function(PACF)

```
#plot Time series
plot(zoo, ylab="Max TemperatureF",xlab="Time")
#plot ACF & PACF
acf<-acf(zoo, lag.max = 20)
pacf<-pacf(zoo, lag.max = 20)
```

We have seen the Time Series before so now we will see the ACF $ PACF Plots





To see how the present values are related to past value and how many past value.

And depending on that you pick ARIMA(p, q)

In R you can get it Automatically

```
library("forecast")
fit<-auto.arima(zoo)
tsdiag(fit)
```

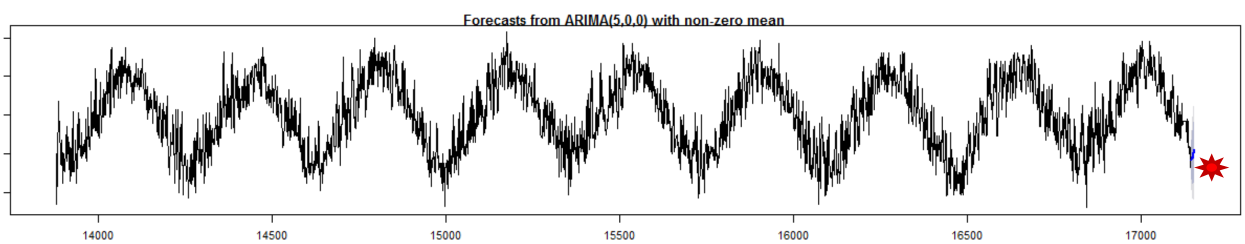
Which gives us the best result

```
Series: zoo
ARIMA(5,0,0) with non-zero mean

Coefficients:
         ar1      ar2     ar3     ar4     ar5  intercept
      0.7294  -0.0732  0.1101  0.0467  0.1316    60.1361
s.e.  0.0173   0.0215  0.0215  0.0215  0.0174     2.4455

sigma^2 estimated as 61.29:  log likelihood=-11360.02
AIC=22734.03   AICc=22734.07   BIC=22776.68
```

The best Model is ARIMA(5,0,0)

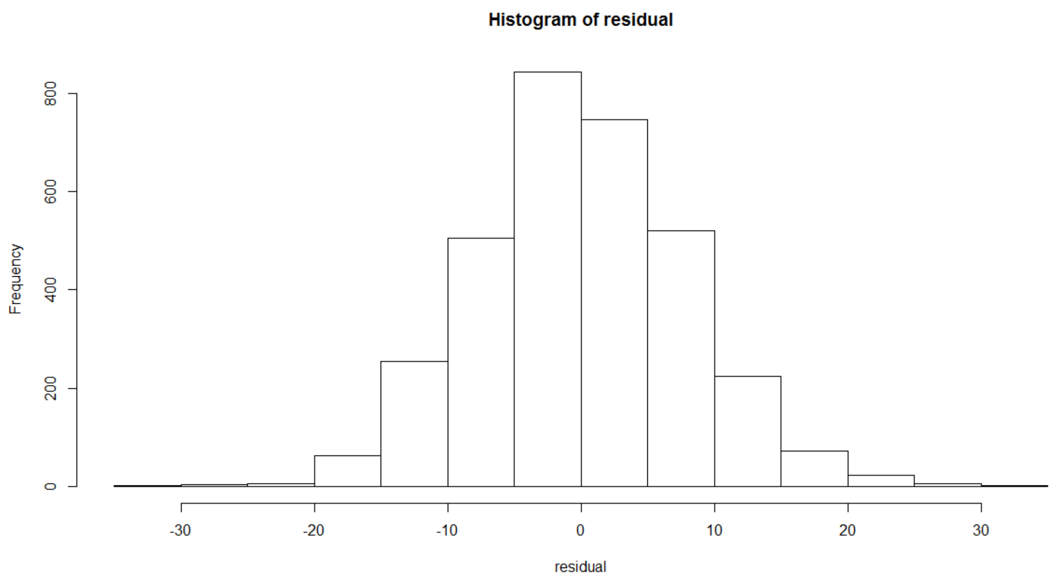
Forecast for next 10 days

```
fc<-forecast.Arima(fit,h=10)
plot.forecast(fc)
```



The forecast for next 10 days is near the red star

```
plot.ts(fc$residuals)
residual<-resid(fit)
hist(residual)
par("mar")
par(mar=c(1,1,1,1))
```

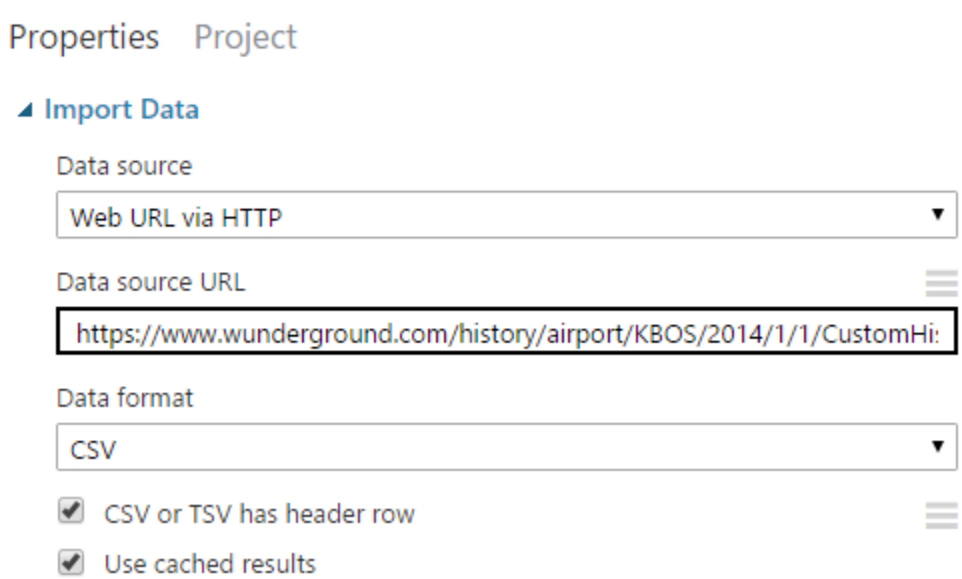


Azure Experiment

Now that we Know how we can forecast the data for next 10 days we will create an azure experiment which gets the forecast.

4.1 Data collection

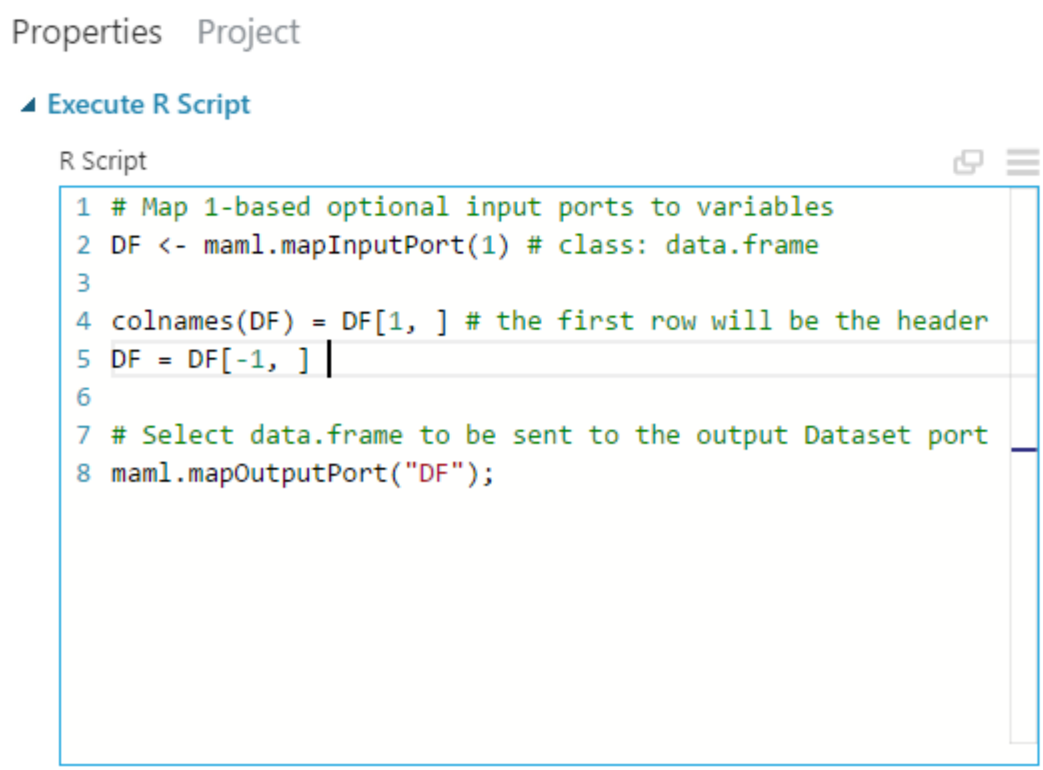
The data is collected in Azure from the URL, the past year data is stored in cache and the new data is not stored in cache and hence every time we call the model we get the latest data.



4.2 Data Cleaning

4.2.1 Add all 9 years of data using ADD ROW module

4.2.2 Make the first column the header of data



4.2.3 Run Rscript to complete the time series & remove NA

```
# Map 1-based optional input ports to variables

df1 <- maml.mapInputPort(1) # class: data.frame

library(dplyr)

library(stringi)

library(zoo)

df1$EST<-as.Date(df1$EST)

a<-Sys.Date()

ts <- seq.POSIXt(as.POSIXct("2008-01-01",'%m/%d/%y'), as.POSIXct(a,'%m/%d/%y'), by="day")
```

```
ts <- seq.POSIXt(as.POSIXlt("2008-01-01"), as.POSIXlt(a),by="d")
ts <- format.POSIXct(ts)#,'%y-%m-%d')


df <- data.frame(timestamp=ts)
df$timestamp<-as.Date(df$timestamp)
df1$timestamp<-df1$EST
data<- merge(df,df1,by="timestamp",all.x = T)
data1<-na.locf(data)
# extract the numbers:
nums <- stri_extract_all_regex(data1$WindDirDegrees.br..., "[0-9]+")
# Make vector and get unique numbers:
nums <- unlist(nums)
data1$WindDirDegrees.br...<-nums
# Select data.frame to be sent to the output Dataset port
maml.mapOutputPort("data1");
```

4.3 Forecast the Temperature for next 10 days

Properties Project

Execute R Script

R Script

```r
# Map 1-based optional input ports to variables
myts<- maml.mapInputPort(1) # class: data.frame
library("zoo")
ZOO <- zoo(myts[,2], order.by=myts[,1])
library("forecast")
fitt <- Arima(ZOO, order=c(9,0,0))
fc<-forecast.Arima(fitt,h=10)
plot.forecast(fc)
a<-Sys.Date()
b<-seq(as.Date(a), by = "day", length.out = 10)
fc<-as.data.frame(fc)
fc$Date<-as.character(b)
# Select data.frame to be sent to the output Dataset port
maml.mapOutputPort("fc");
```

When checked for AIC ARIMA(9,0,0) has smaller AIC than ARIMA(5,0,0) so we have modelled it for ARIMA(9,0,0)

The forecast Data we get is:- (snap shot taken on 12 Dec 2016)

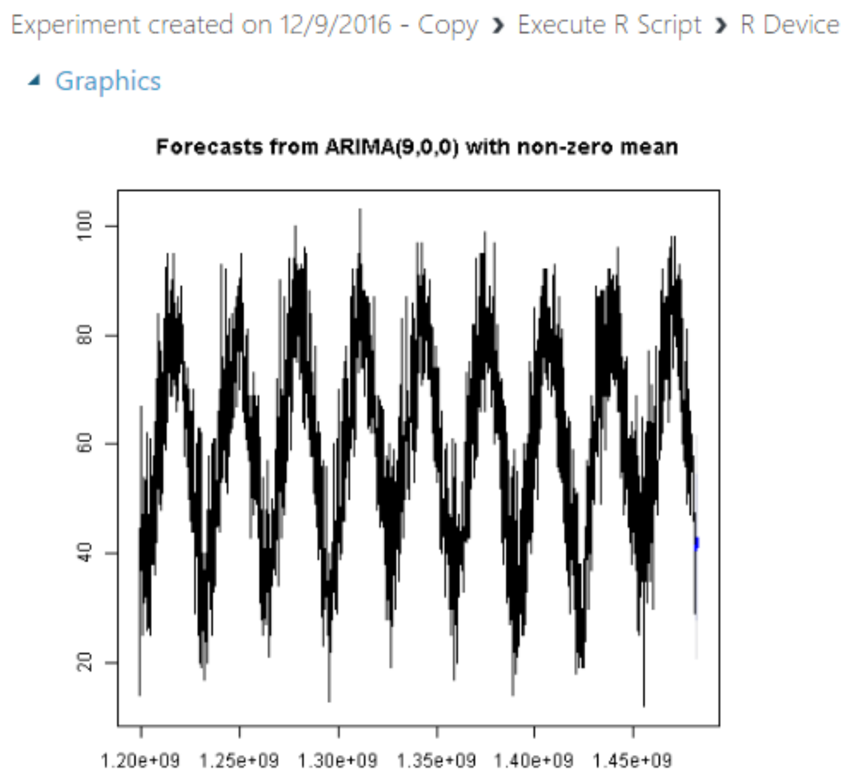
Experiment created on 12/9/2016 - Copy › Add Columns › Combined dataset

rows 10 columns 7

view as

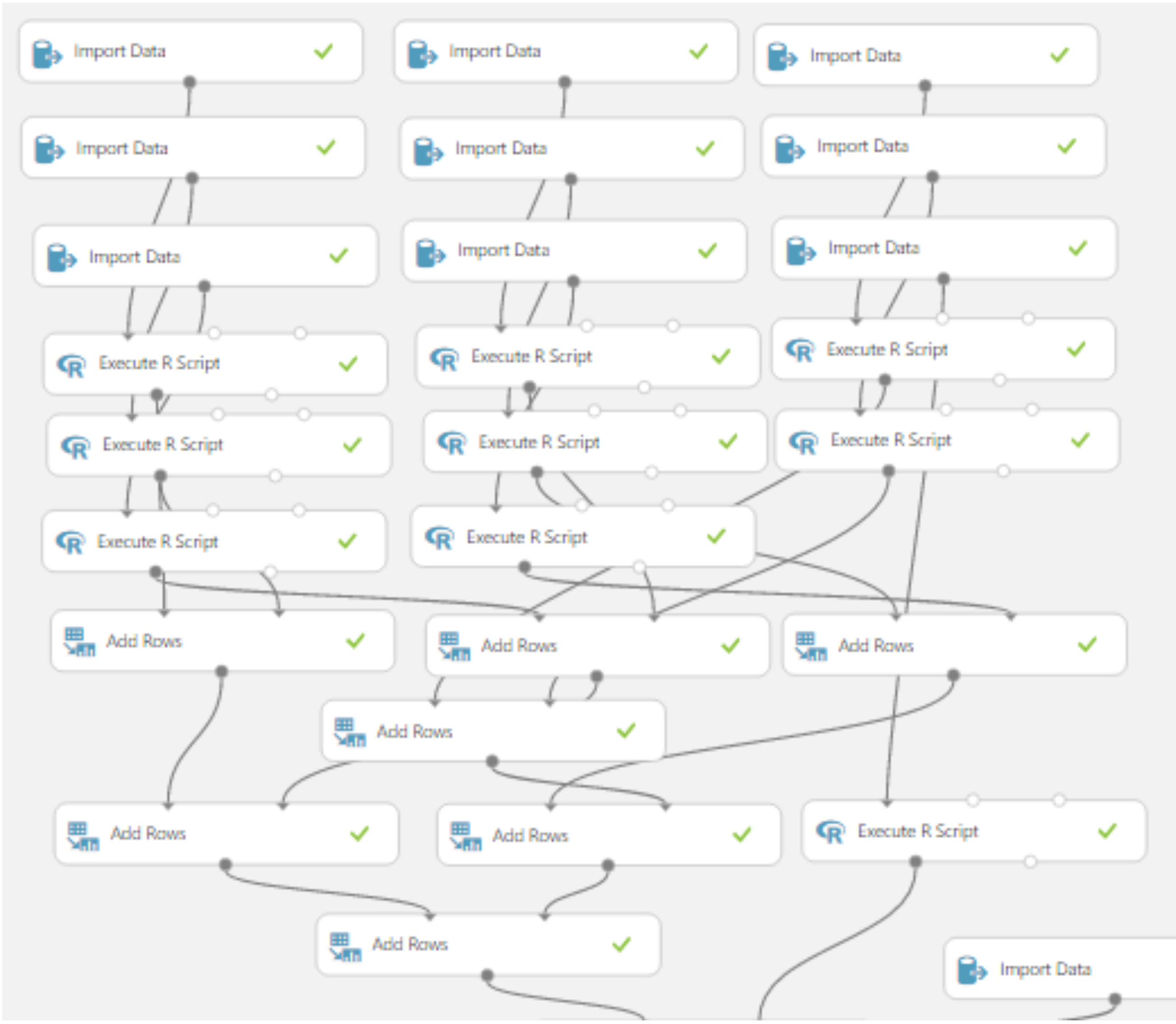
Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95	Date
41.170795	31.359409	50.98218	26.165575	56.176014	2016-12-13
41.410144	29.516466	53.303823	23.220333	59.599956	2016-12-14
41.223294	28.760138	53.686449	22.162542	60.284045	2016-12-15
41.438204	28.680966	54.195442	21.927692	60.948716	2016-12-16
41.380202	28.389684	54.370719	21.51292	61.247484	2016-12-17
40.65607	27.460847	53.851293	20.475718	60.836422	2016-12-18
40.986045	27.530648	54.441443	20.40779	61.5643	2016-12-19
42.131898	28.421241	55.842555	21.163258	63.100538	2016-12-20

And the forecast graph on Azure is

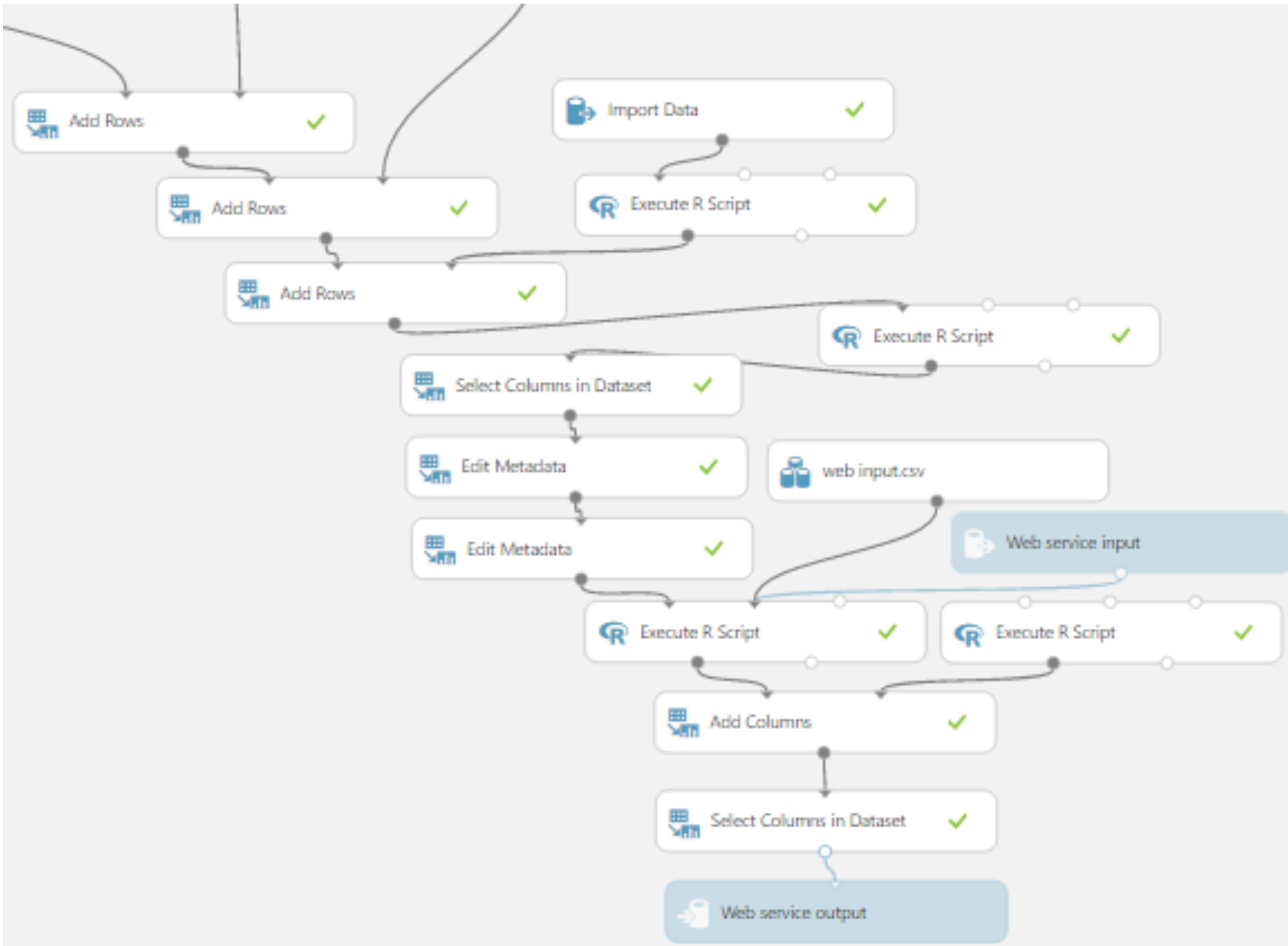


The total Azure Experiment is below

Part 1


Import Data
Import Data
Import Data
Import Data
Import Data
Import Data
Import Data
Import Data
Import Data
Execute R Script
Execute R Script
Execute R Script
Execute R Script
Execute R Script
Execute R Script
Execute R Script
Execute R Script
Add Rows
Add Rows
Add Rows
Add Rows
Add Rows
Add Rows
Add Rows
Execute R Script
Import Data

Part 2



4.3 Add other weather components

Calculate the ARIMA(p,d,q) for rest of the weather components

```
library("forecast")
fit<-auto.arima(zoo)
tsdiag(fit)
```

Weather component	(p,d,q)
Mean.TemperatureF	(9,0,0)
MeanDew.PointF	(1,0,1)
Mean.Humidity	(2,0,2)
Mean.VisibilityMiles	(0,1,2)
PrecipitationIn	(0,0,1)
CloudCover	(0,0,1)
Events	Cannot be computed

Except Events add the above to the experiment

And like before with the Max. TemperatureF we forecast the weather for next 10 days.

Experiment created on 12/9/2016 - Copy › Select Columns in Dataset › Results dataset

rows 10 columns 7

view as

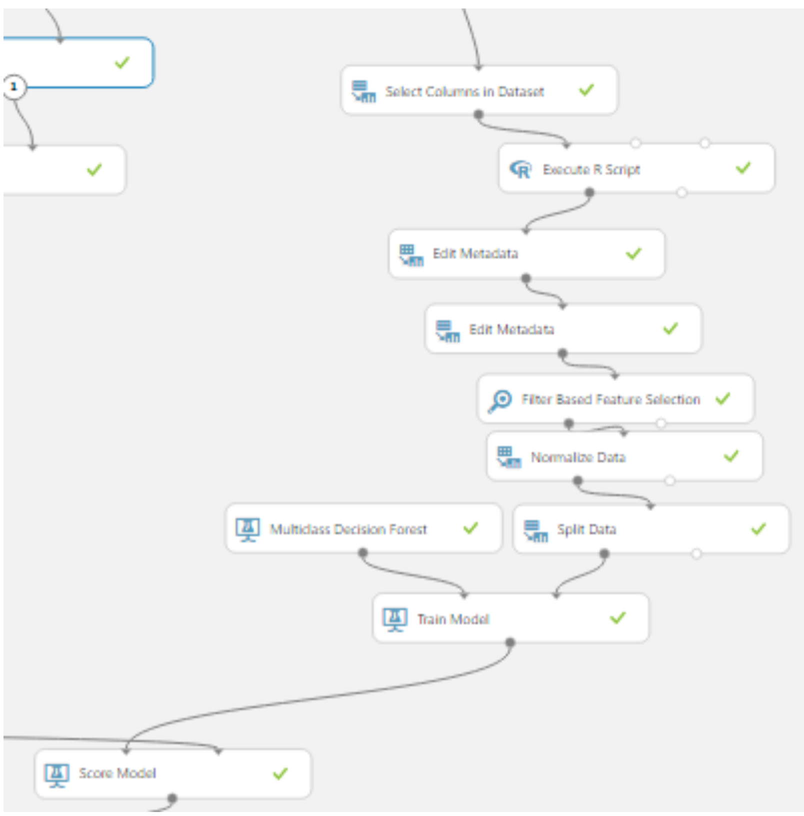
PrecipitationIn	Date	CloudCover	Mean VisibilityMiles	Mean Humidity	Temperature	Dew Point
0.104154	2016-12-16	3.410632	9.468701	49.023469	25.219758	-8.92302
0.116214	2016-12-17	4.732635	9.31412	62.462681	31.661309	-3.350474
0.116214	2016-12-18	4.732635	9.31412	64.919279	33.028406	1.589046
0.116214	2016-12-19	4.732635	9.31412	65.366482	34.032802	5.96745
0.116214	2016-12-20	4.732635	9.31412	65.446085	34.997861	9.84848
0.116214	2016-12-21	4.732635	9.31412	65.45848	34.872078	13.288636
0.116214	2016-12-22	4.732635	9.31412	65.458627	35.126492	16.337999
0.116214	2016-12-23	4.732635	9.31412	65.456582	34.406569	19.040964

4.3 Compute the Events

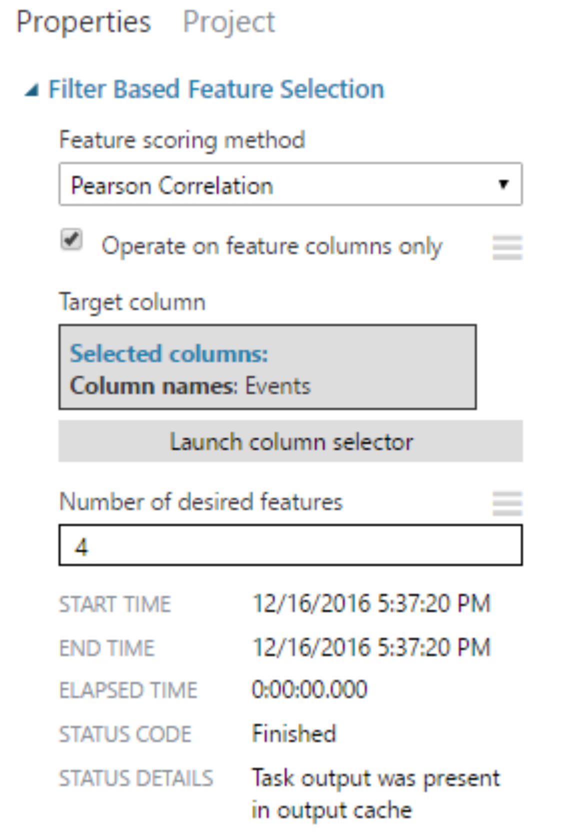
Events is the parameter of the weather which cannot be computed by using past year data. So we are going to perform multi-class classification for Events using dependent variables mentioned above

Weather component	(p,d,q)	Variable Type
Mean.TemperatureF	(9,0,0)	Independent
MeanDew.PointF	(1,0,1)	Independent
Mean.Humidity	(2,0,2)	Independent
Mean.VisibilityMiles	(0,1,2)	Independent
PrecipitationIn	(0,0,1)	Independent
CloudCover	(0,0,1)	Independent
Events	Perform Regression	Dependent

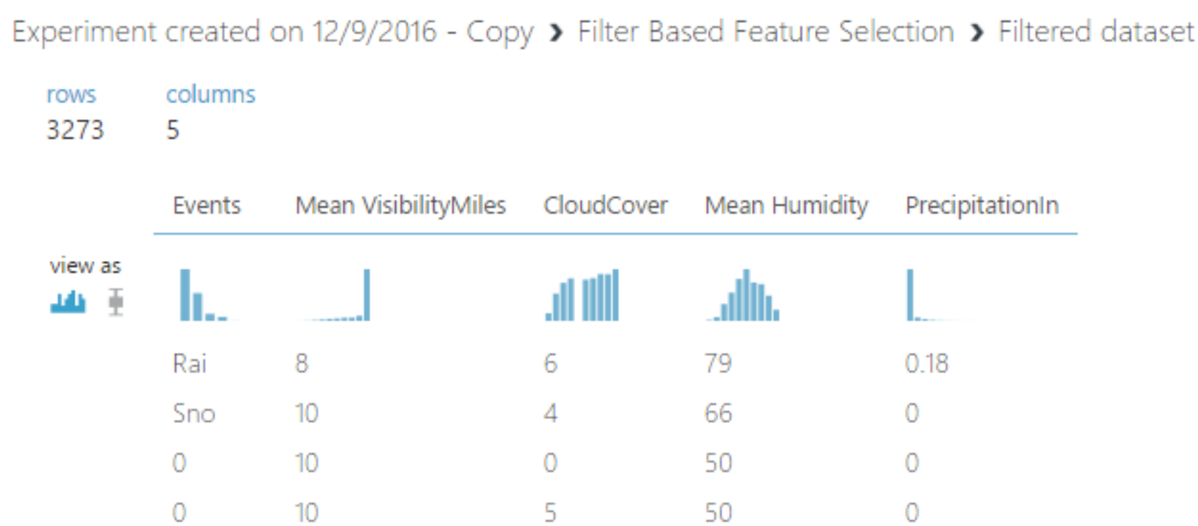
Now get the data from the Rscript and filter out only the above components. And do a multi-class regression.



Perform Filter Based Feature Selection.



We get the following dependent variables which impact the Events


Experiment created on 12/9/2016 - Copy › Filter Based Feature Selection › Filtered dataset

rows 3273 columns 5

Events	Mean VisibilityMiles	CloudCover	Mean Humidity	PrecipitationIn
Rai	8	6	79	0.18
Sno	10	4	66	0
0	10	0	50	0
0	10	5	50	0

The Evaluated model has the following result.

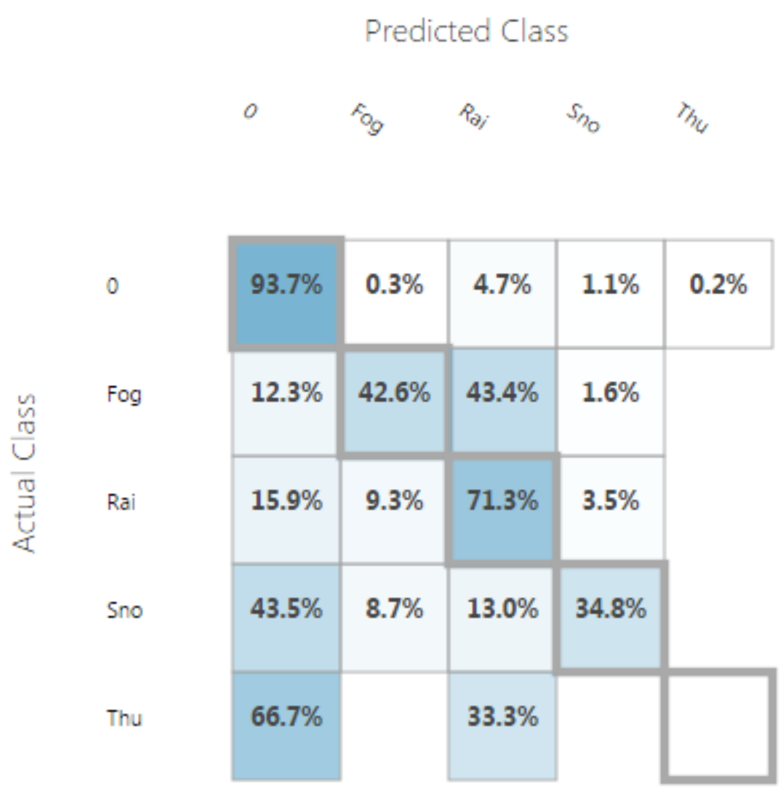
Experiment created on 12/9/2016 - Copy › Evaluate Model › Evaluation results

Metrics

Overall accuracy	0.802567
Average accuracy	0.921027
Micro-averaged precision	0.802567
Macro-averaged precision	0.514033
Micro-averaged recall	0.802567
Macro-averaged recall	0.484697

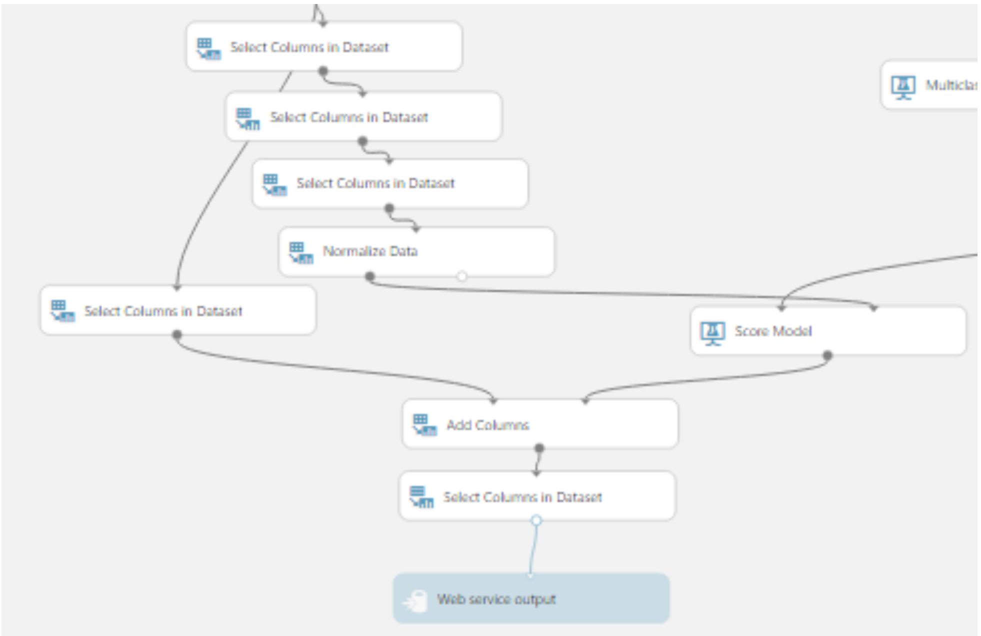
And a confusion matrix

Confusion Matrix



Once the confusion matrix is ready score the model.

4.4 Predict the 'Events' using dependent variable mentioned above.



To the score model we will add the forecasted data and predict the Events.

We get the following results.

Experiment created on 12/9/2016 - Copy › Select Columns in Dataset › Results dataset

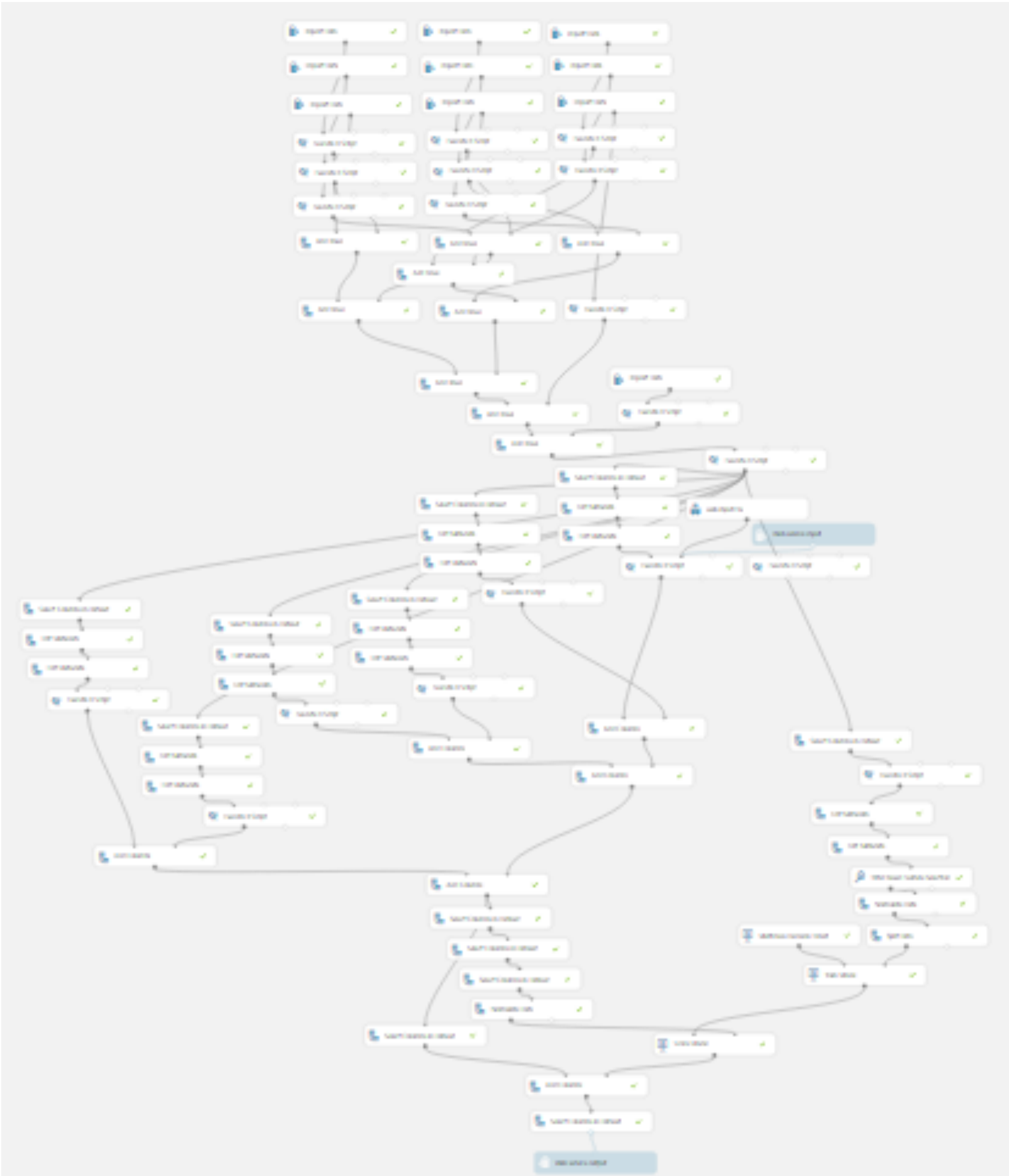
rows 10

columns 8

view as

Date	Temperature	Dew Point	PrecipitationIn	CloudCover	Mean VisibilityMiles	Mean Humidity	Scored Labels
2016-12-16	25.219758	-8.92302	0	0	1	0	0
2016-12-17	31.661309	-3.350474	1	1	0	0.817711	Fog
2016-12-18	33.028406	1.589046	1	1	0	0.967183	Fog
2016-12-19	34.032802	5.96745	1	1	0	0.994393	Fog
2016-12-20	34.997861	9.84848	1	1	0	0.999237	Fog
2016-12-21	34.872078	13.288636	1	1	0	0.999991	Fog
2016-12-22	35.126492	16.337999	1	1	0	1	Fog

4.5 Create a Web Service for the above model



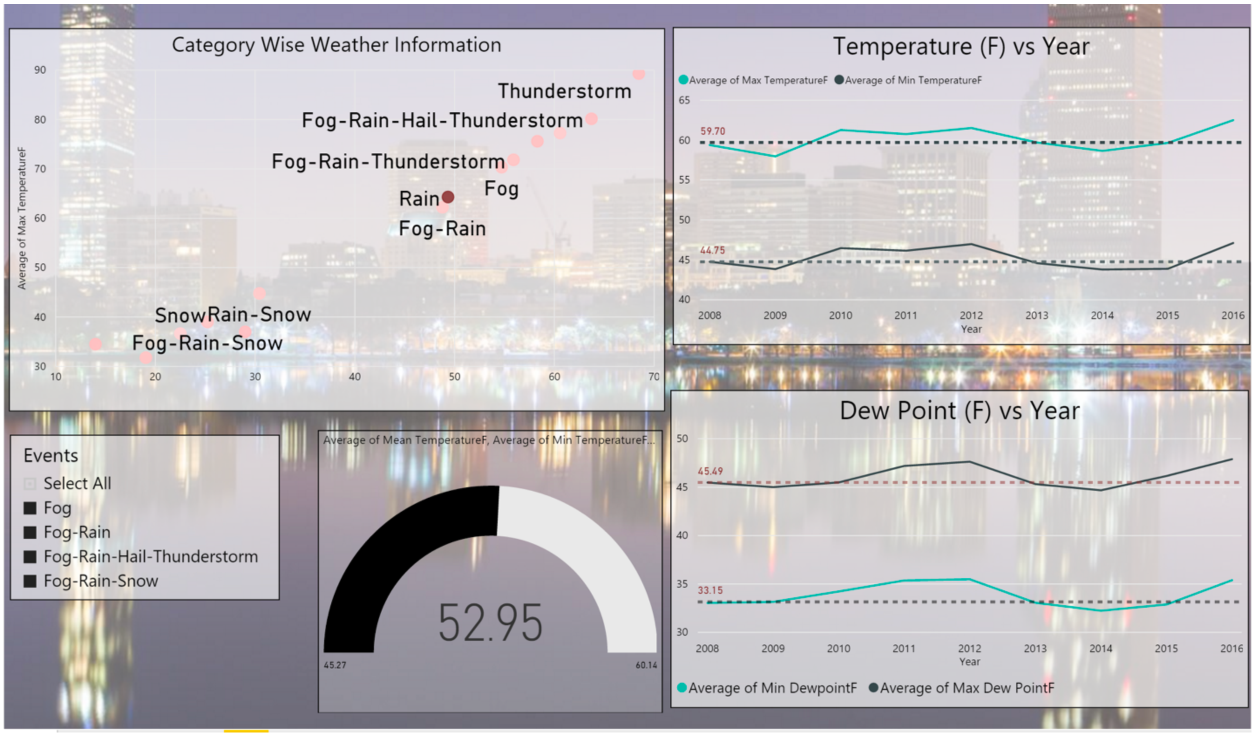
For reference login into azure as the pic is not clear

Azure login

Login Id : teameight2017@gmail.com

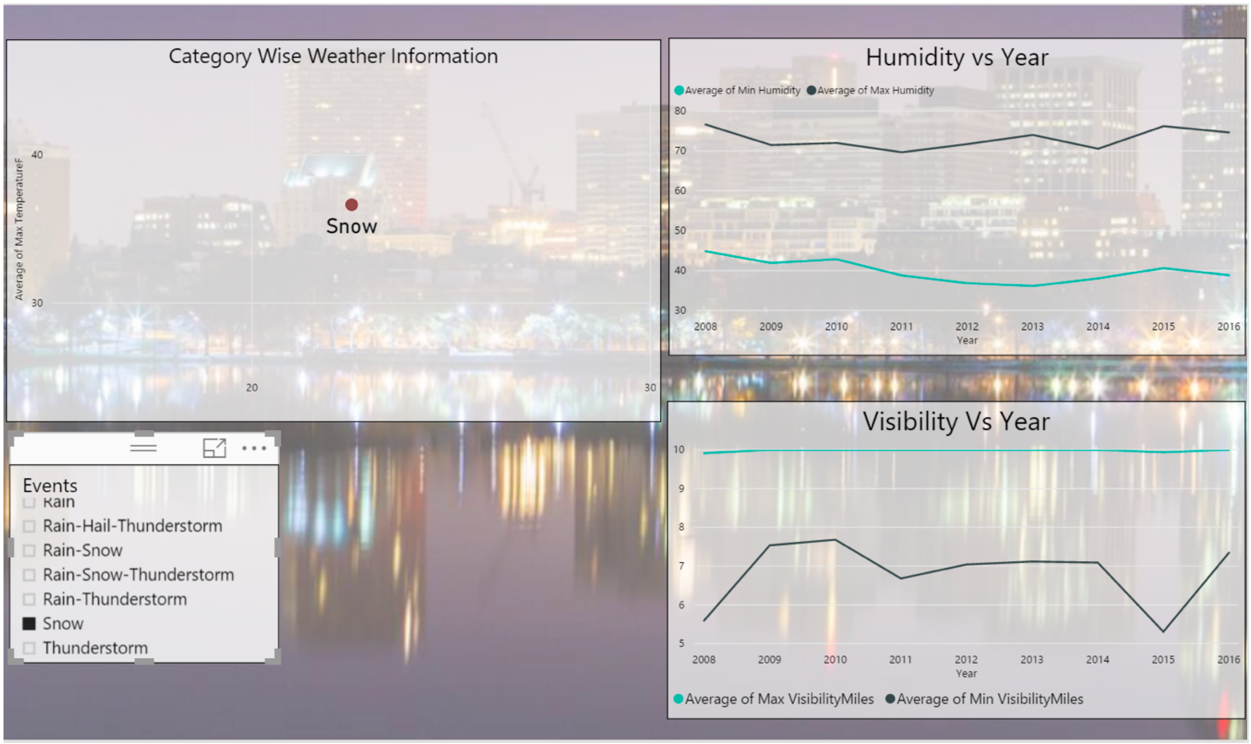
Password : team8ads

POWER BI:



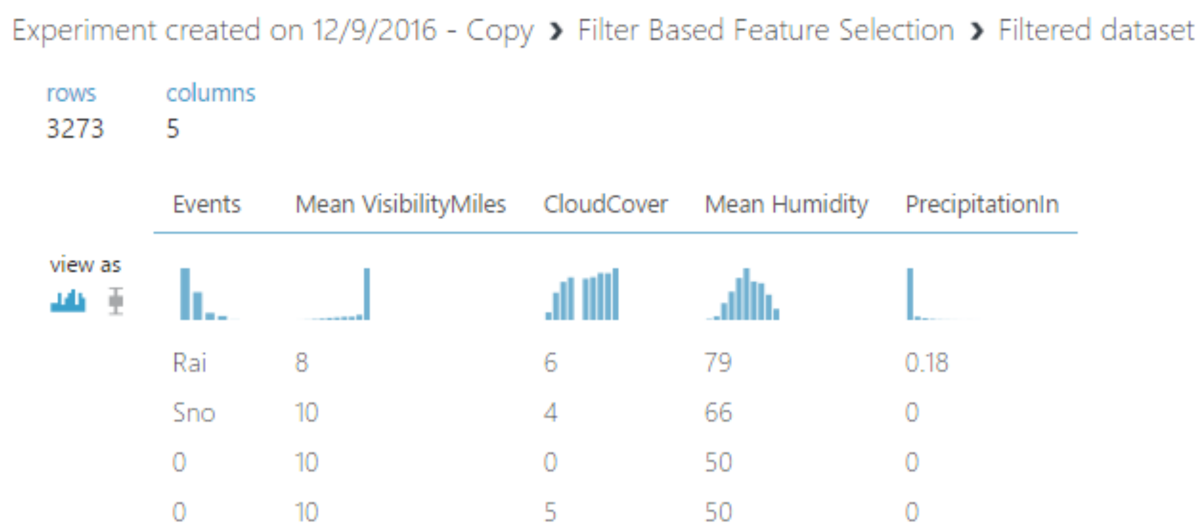
- The Above dashboard provides information about weather events occurring over span of 8 years and, trend of max.

and min temperature, dew point, visibility recorded over the years.



- We can see that the visibility had dropped drastically in the year 2015 during the snow in Boston.

Using which we have decided that we can compute the "Events" using the other weather data as they had correlation between them.


Experiment created on 12/9/2016 - Copy › Filter Based Feature Selection › Filtered dataset

rows	columns
3273	5

view as

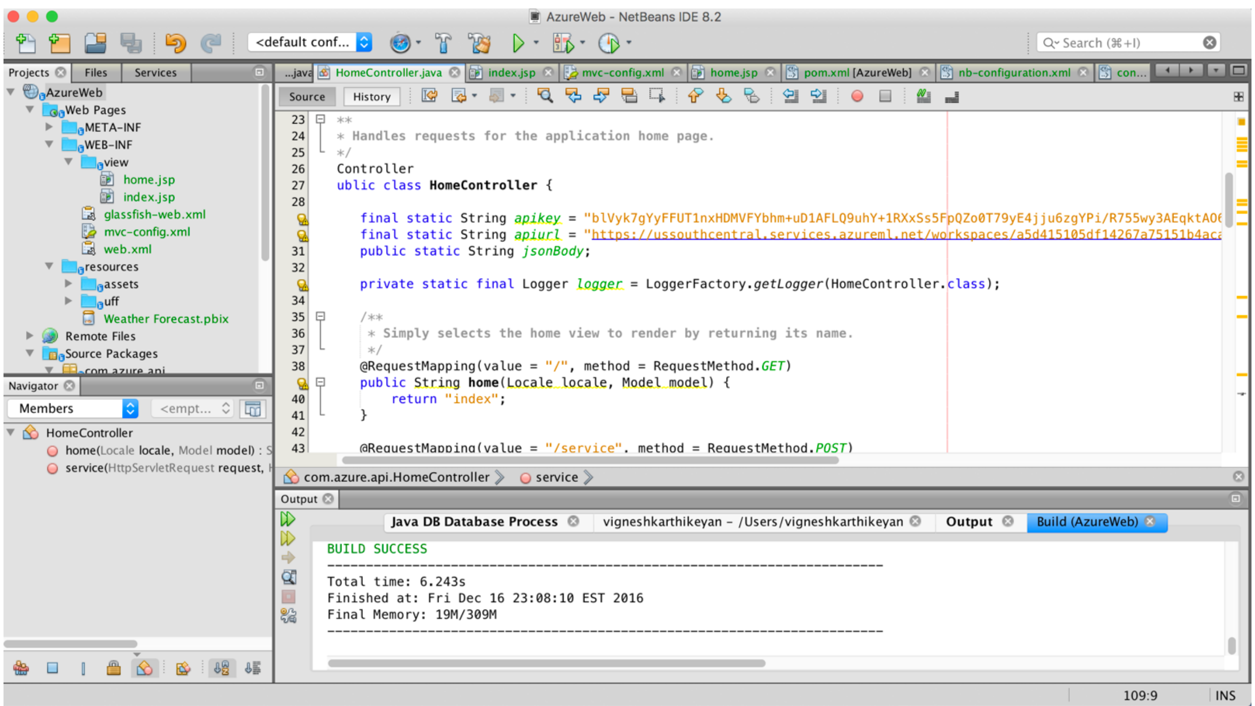
Events	Mean VisibilityMiles	CloudCover	Mean Humidity	PrecipitationIn
Rai	8	6	79	0.18
Sno	10	4	66	0
0	10	0	50	0
0	10	5	50	0

Web Application:

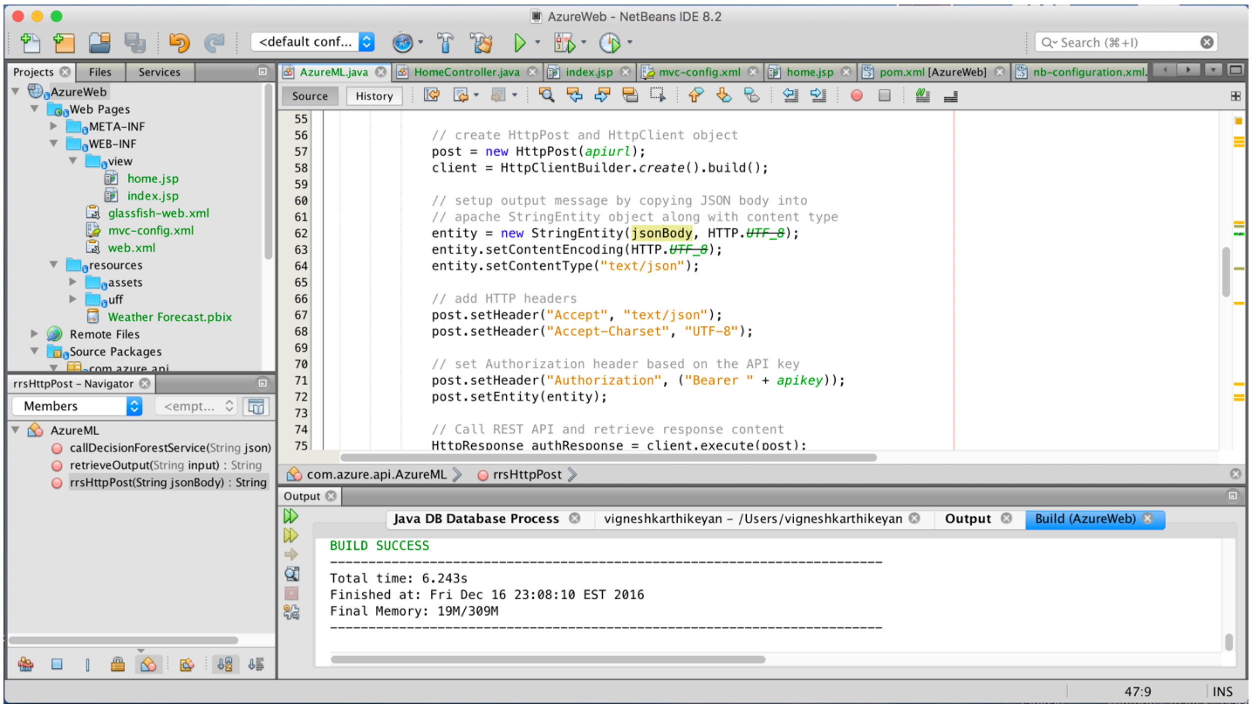
AWS Link: http://sample-env-1.qmntzmwe52.us-west-2.elasticbeanstalk.com

We created a MVC Java application on NetBeans and deployed the application on Amazon AWS.

Sample Code:



The Home Controller - where we make the API call


AzureWeb - NetBeans IDE 8.2
Projects
Files
Services
AzureWeb
Web Pages
META-INF
WEB-INF
view
home.jsp
index.jsp
glassfish-web.xml
mvc-config.xml
web.xml
resources
assets
uff
Weather Forecast.pbix
Remote Files
Source Packages
rrsHttpPost - Navigator
Members
AzureML
callDecisionForestService(String json)
retrieveOutput(String input) : String
rrsHttpPost(String jsonBody) : String
AzureML.java
HomeController.java
index.jsp
mvc-config.xml
home.jsp
pom.xml [AzureWeb]
nb-configuration.xml
Source
History
// create HttpPost and HttpClient object
post = new HttpPost(apiurl);
client = HttpClientBuilder.create().build();
// setup output message by copying JSON body into
// apache StringEntity object along with content type
entity = new StringEntity(jsonBody, HTTP.UTF_8);
entity.setContentEncoding(HTTP.UTF_8);
entity.setContentType("text/json");
// add HTTP headers
post.setHeader("Accept", "text/json");
post.setHeader("Accept-Charset", "UTF-8");
// set Authorization header based on the API key
post.setHeader("Authorization", ("Bearer " + apikey));
post.setEntity(entity);
// Call REST API and retrieve response content
HttpResponse authResponse = client.execute(post);
com.azure.api.AzureML
rrsHttpPost
Output
Java DB Database Process
vigneshkarthikeyan - /Users/vigneshkarthikeyan
Output
Build (AzureWeb)
BUILD SUCCESS
Total time: 6.243s
Finished at: Fri Dec 16 23:08:10 EST 2016
Final Memory: 19M/309M
47:9
INS

Reference

- Extract Seasonal & Trend: using decomposition in R: https://anomaly.io/seasonal-trend-decomposition-in-r/
- Time Series Forecasting Theory | AR, MA, ARMA, ARIMA: https://www.youtube.com/watch?v=Aw77aMLj9uM
- Using R for Time Series Analysis: http://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html
- https://www.otexts.org/fpp/8/8
- Detect Seasonality using Fourier Transform in R: https://anomaly.io/detect-seasonality-using-fourier-transform-r/
- Data set : https://www.wunderground.com/