Project Title : Freight Management (Predict the Cost Of Shipment)

Purpose : This project is to prdict about the cost of the shipment , so best price can be suggested to customer which will enable the provider to stop any revenue leak and maximize it's profit

""Quickfreight is the middle man between the customer and freight services. They will charge from the customer and pay to the freight services and during the transaction they will make their profit/revenue. The challenge is, given a request from a customer for shipment they would like to decide the estimated cost for the Carriers. This information will enable them to quote the best price for the customer and reduce the revenue leakages and increase their bottom line.""

Mentors : Mr Jagannadha Rao Basa & Mr Jayant Kumar Mulmoodi

Overview of the Data

After detail discussion with mentor's of the project i have started analyzing the data , and the first thing we did is to clean the data.

	ORDER_NBR	EQUIPMENT_TYPE	CUSTOMER_MILES	WEIGHT	ORDER_COST	FIRST_PICK_ZIP	FIRST_PICK_EARLY_APPT
1	167159	R	1832	41976	3450.00	67501	1/6/2016 0:00
2	167160	R	1136	41344	2800.00	67501	1/2/2016 0:00
3	167161	R	1832	42000	3900.00	67501	1/6/2016 0:00
4	167162	R	1832	42000	3700.00	67501	1/4/2016 0:00
5	167163	R	1744	41344	3650.00	67501	1/4/2016 0:00
6	167164	R	941	41000	1600.00	33309	1/4/2016 0:00
7	167164	R	941	41000	1600.00	33309	1/4/2016 0:00

FIRST_PICK_LATE_APPT	LAST_DELIVERY_ZIP	LAST_DELIVERY_EARLY_APPT	LAST_DELIVERY_LATE_APPT	IS_HAZARDOUS
1/6/2016 0:00	98372	1/11/2016 0:00	1/11/2016 0:00	N
1/3/2016 0:00	24153	1/5/2016 0:00	1/5/2016 0:00	N
1/6/2016 0:00	98372	1/8/2016 0:00	1/8/2016	N
1/4/2016 0:00	98372	1/8/2016 0:00	1/8/2016 0:00	N
1/4/2016 0:00	97015	1/8/2016 0:00	1/8/2016 0:00	N
1/4/2016 0:00	23224	1/6/2016 0:00	1/6/2016 0:00	N
1/4/2016 0:00	23111	1/6/2016 0:00	1/6/2016 0:00	N

If we see above this is the kind of data available in the beginning .

There are many features of the data for which there are spaces or some other values which does not make any sense. Like the order cost is 0 , or the customer miles is 0 or the weight

of the parecel is 0 , such things do not make any sense as we know for a simple order like documents also weigh in at leaset some grams and it will have some order cost and has to travel for some miles to be delivered.

So we have discarded those types of data.

Now the other type of data processing we have done is that for the EQUIPMENT_TYPE we have a look up table where the EQUIPMENT_TYPE code will match to a particular vehicle , like given below:

R=REFRIGERATED

V=DRY FREIGHT

VM=DRY FREIGHT

VR=DRY FREIGHT

LTL=DRY FREIGHT

So we have replaced these values for the equipement types.

Then another feature we have decided to have in stead of Source Zip Code and Destination region , it will help us going forward because if we consider the source and desination region zip code then the number of level of those features is way more than some thousands.

But when we replace them with the source and destination region (we get those values given to us in another look up region code dataset) the levels of those features came down drastically to 15 levels each.

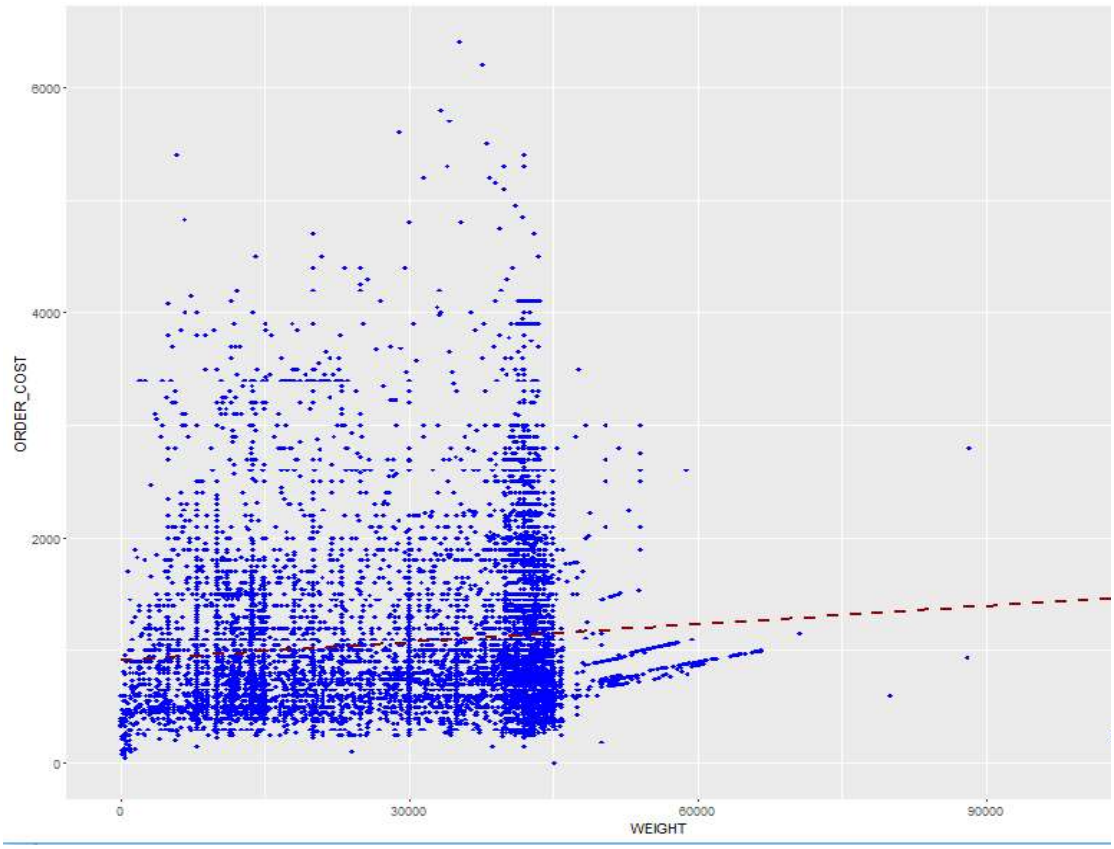
Also we have derived another feature which we thought that it is going to give us some more benefit is the day of the week the lte delivery date is falling on . The idea behind is this is that we want to find out if the cost of the order is going to increase or decrease if it i falling on week-end or any particular day of week.

THe most important feature we derived is the time (in minutes) the order should deliver. For that we have taken the first pick early appointment and last delivery late appointment.

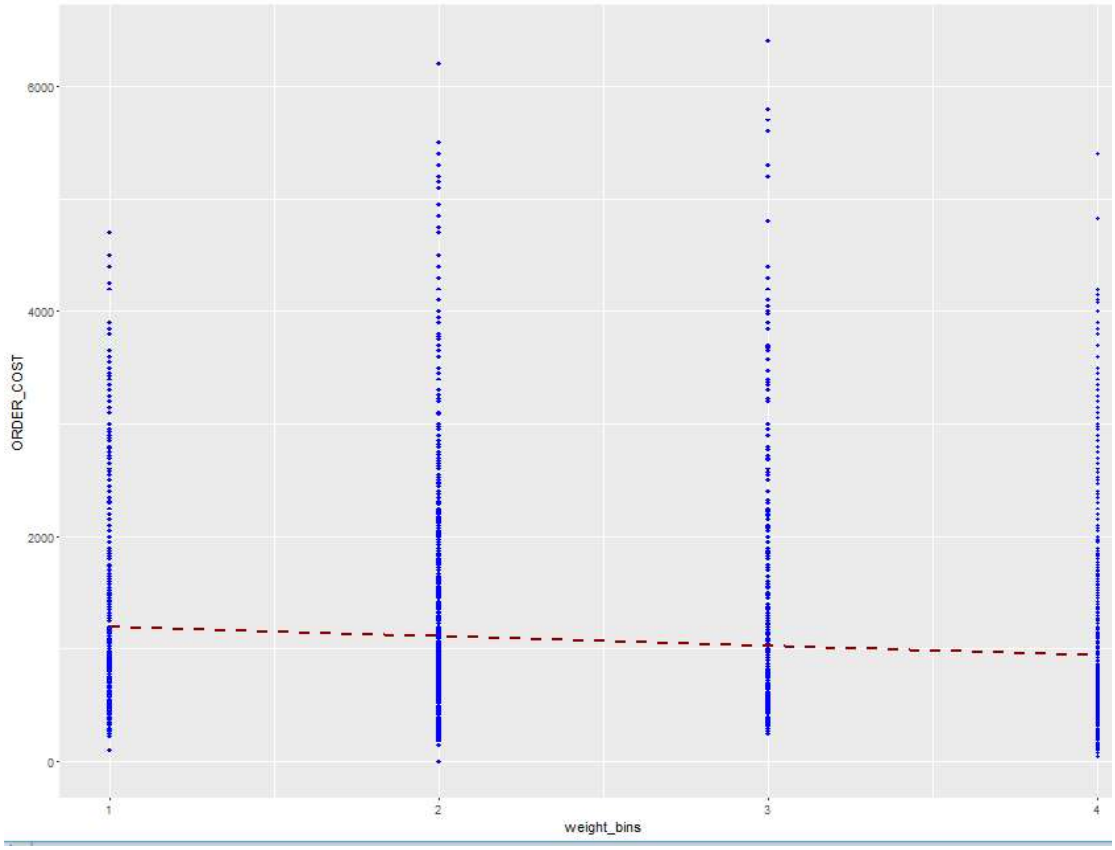
The idea behind is to calculate the maximum time avaiable for delivery and to include that in the feature list to predict the cost of the order.

Also there is another feature the weight of the parcel , we thought to directly consider this as part of the prediction of the cost but the way that data is distributed we thought to cluser that to yeild some better result.

This is the plot for the weight and order cost.

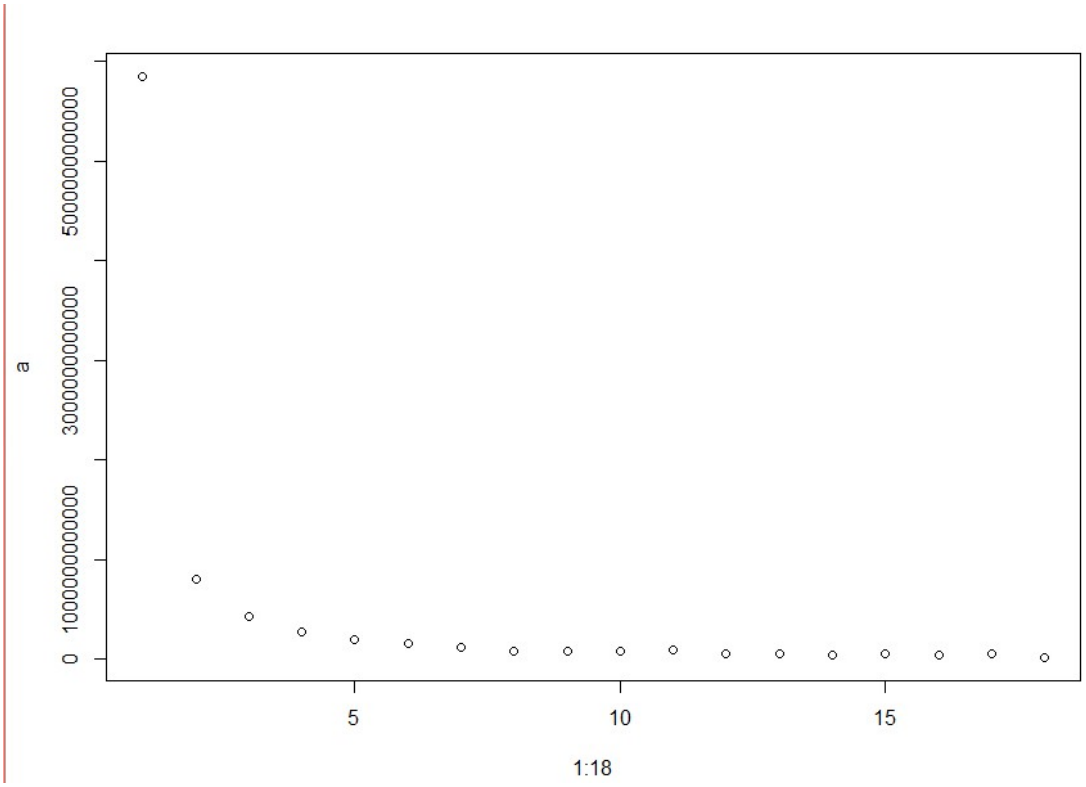


Below is the plot for the cluster of weight we have made (we have made 4 clusters for the weights.)



We can see the variability of the data is there in both the cases but the binned weight will be more easy to handle than the large amount of variable weights which was originally available in the dataset.

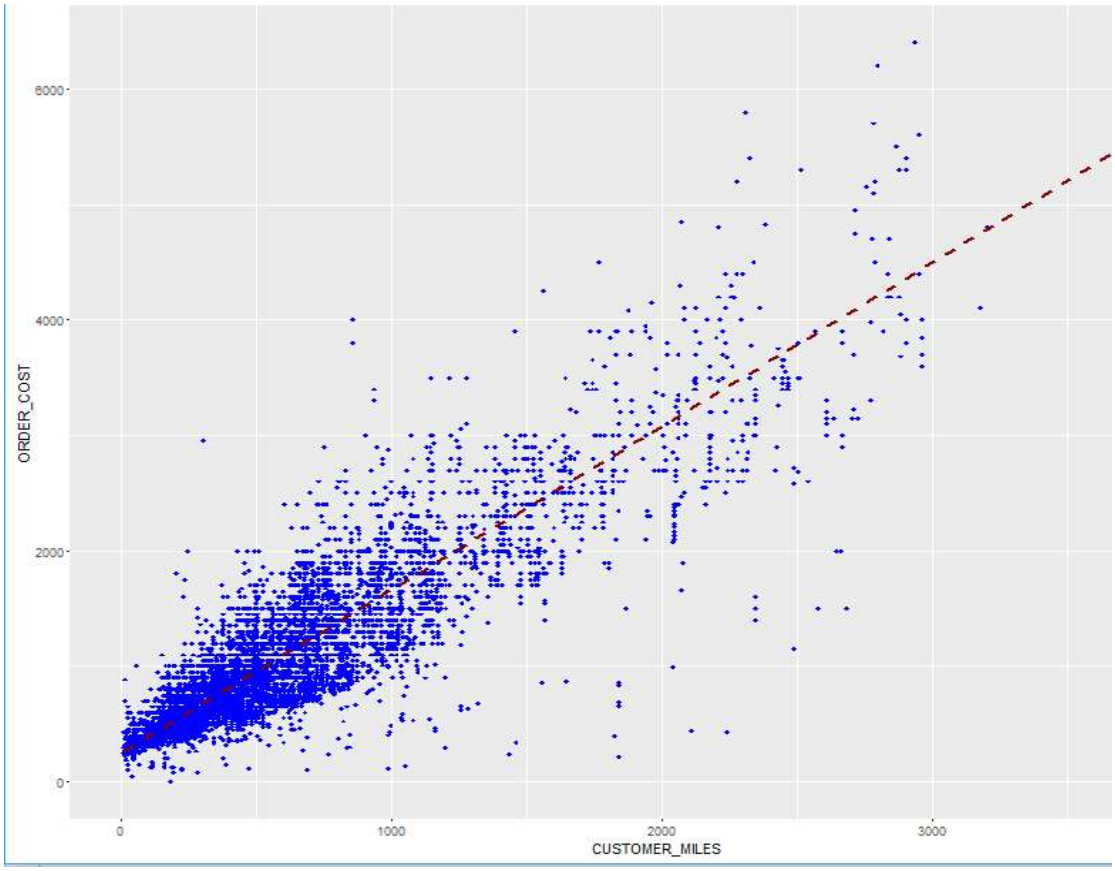
The way we have clsutered the data is by using k-means menthod with k value as four and i have given the plot for that below as to why we use 4 for the value of k.



If we can see the variability is not that much after the 4th observation for k value. So we have choose the value of k as 4.

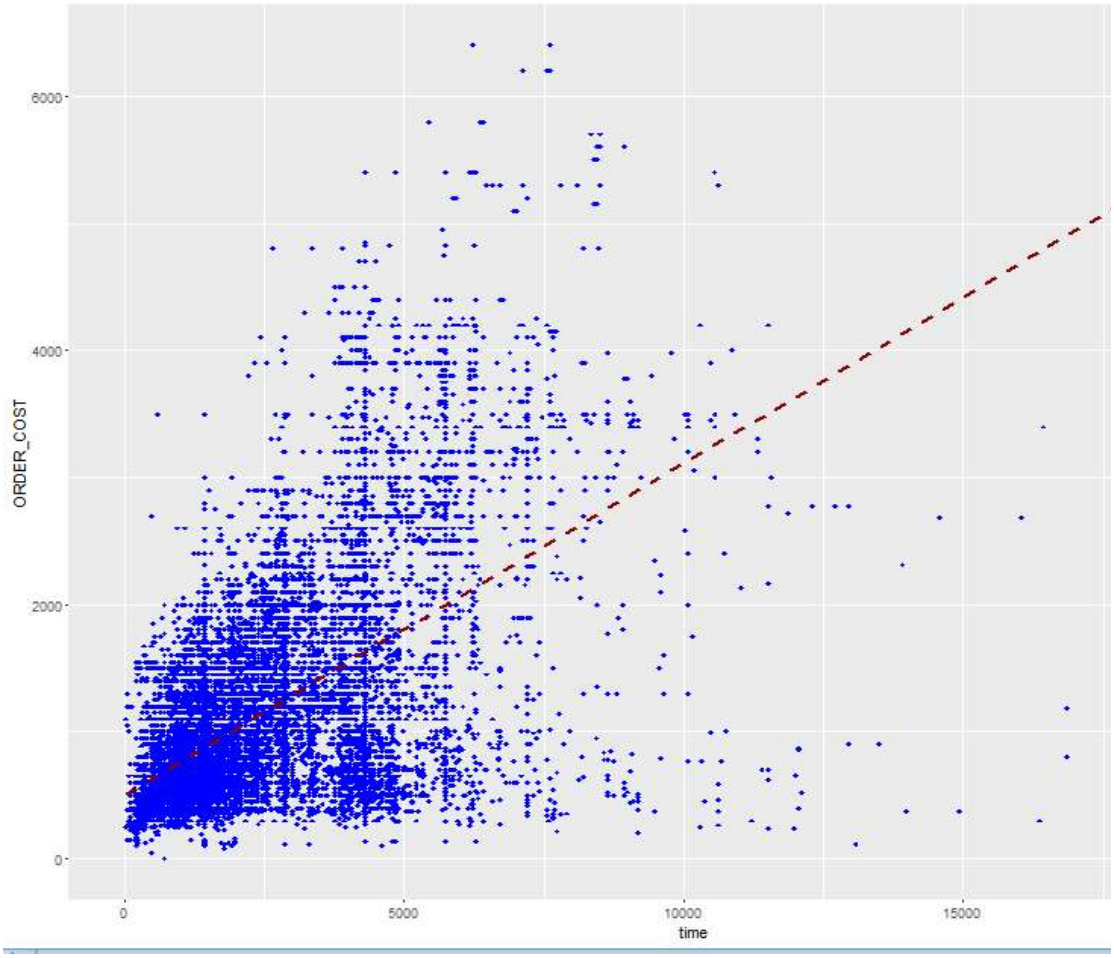
Below we have given some more visualziation of the features we have.

Plot for Customer miles with the order cost. CLearly we can see a pattern an increasing pattern that as the number of miles is increasing then the cost of the order is also increasing.



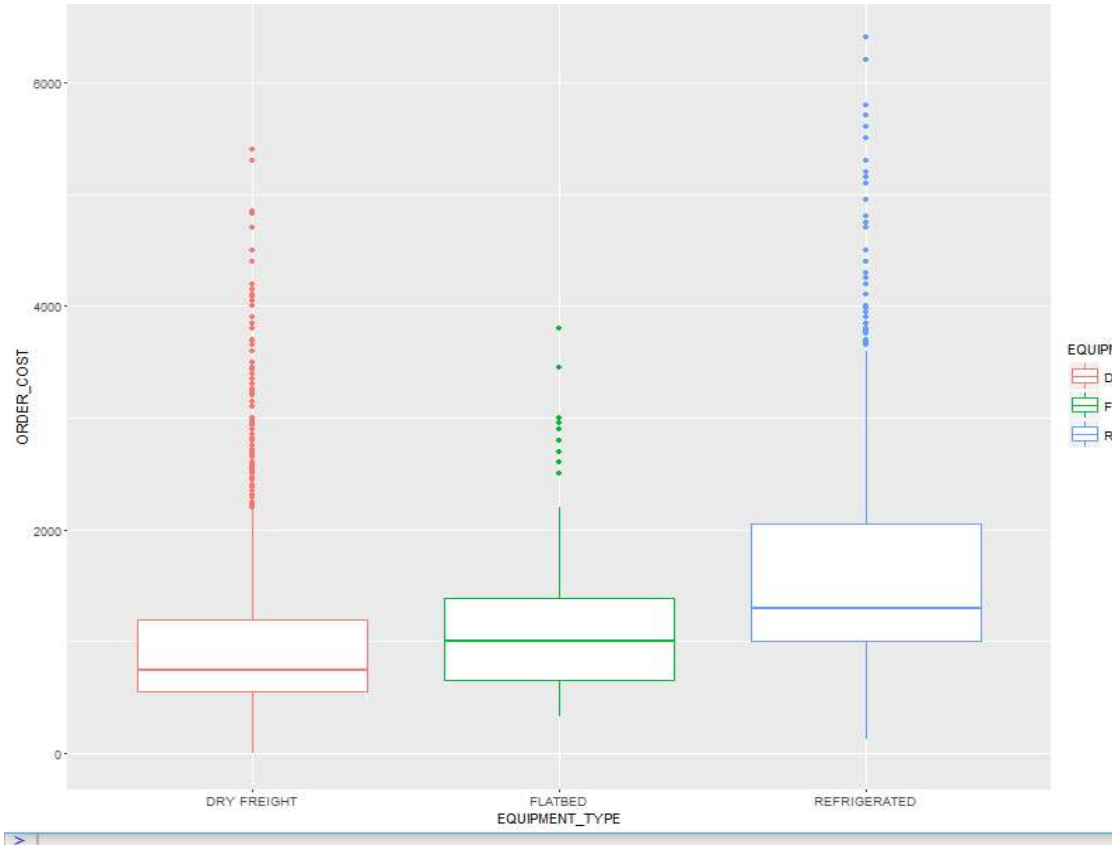
Below we have the plot for the derived feature of the time versus order cost. If we can see for some values the time for delivery is quite large but the order cost is not that much high , and if we see the plot below then most of the orders are in between 0 to 5000 minutes of delivery time and the cost of the order is also mixed , some what increasing trend though.

We can say that if the delivery time is beyond 7000 minutes then we do not have much of the data falling to that category and also the cost of the order is also in mixed state not have any kind of particular order of increasing or decreasing.

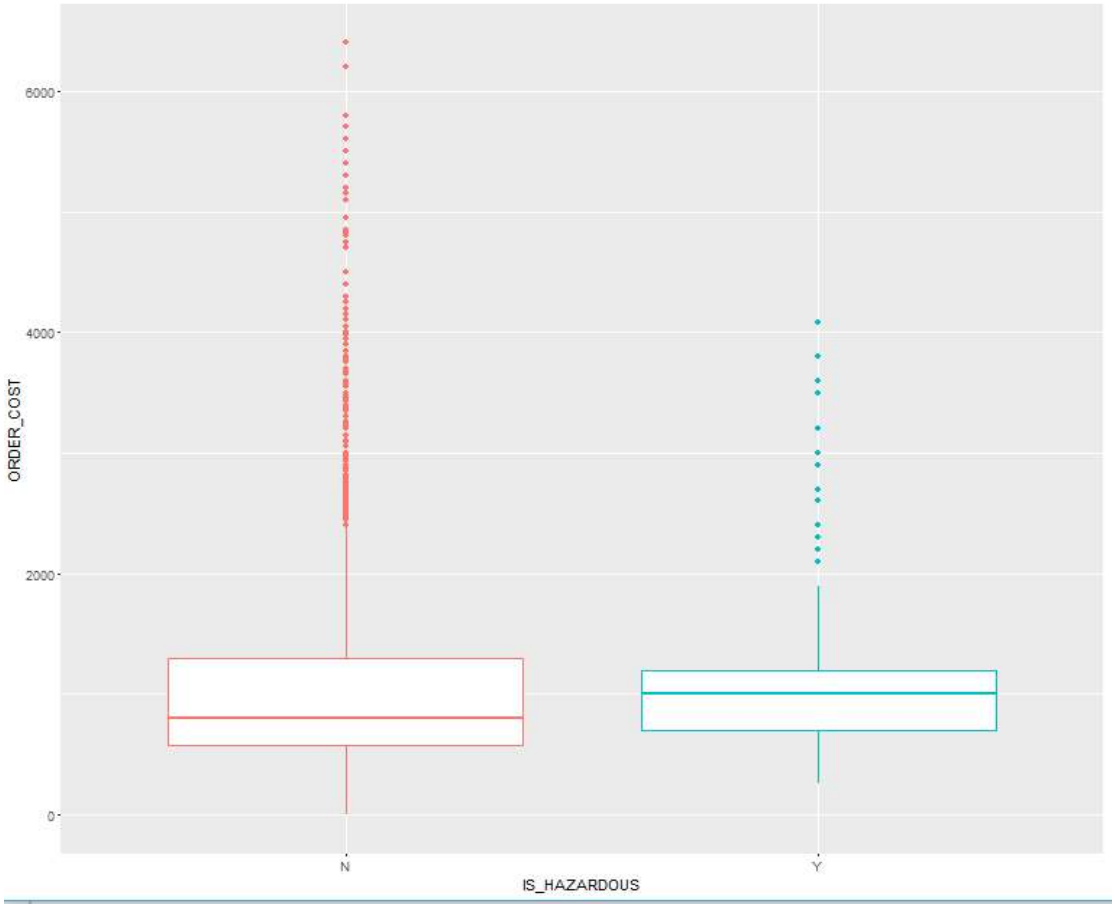


For the equipment type and order cost. We can clearly see that the order cost is increasing in the order of DRY FREIGHT ---> FLATBED ---> REFRIGERATED.

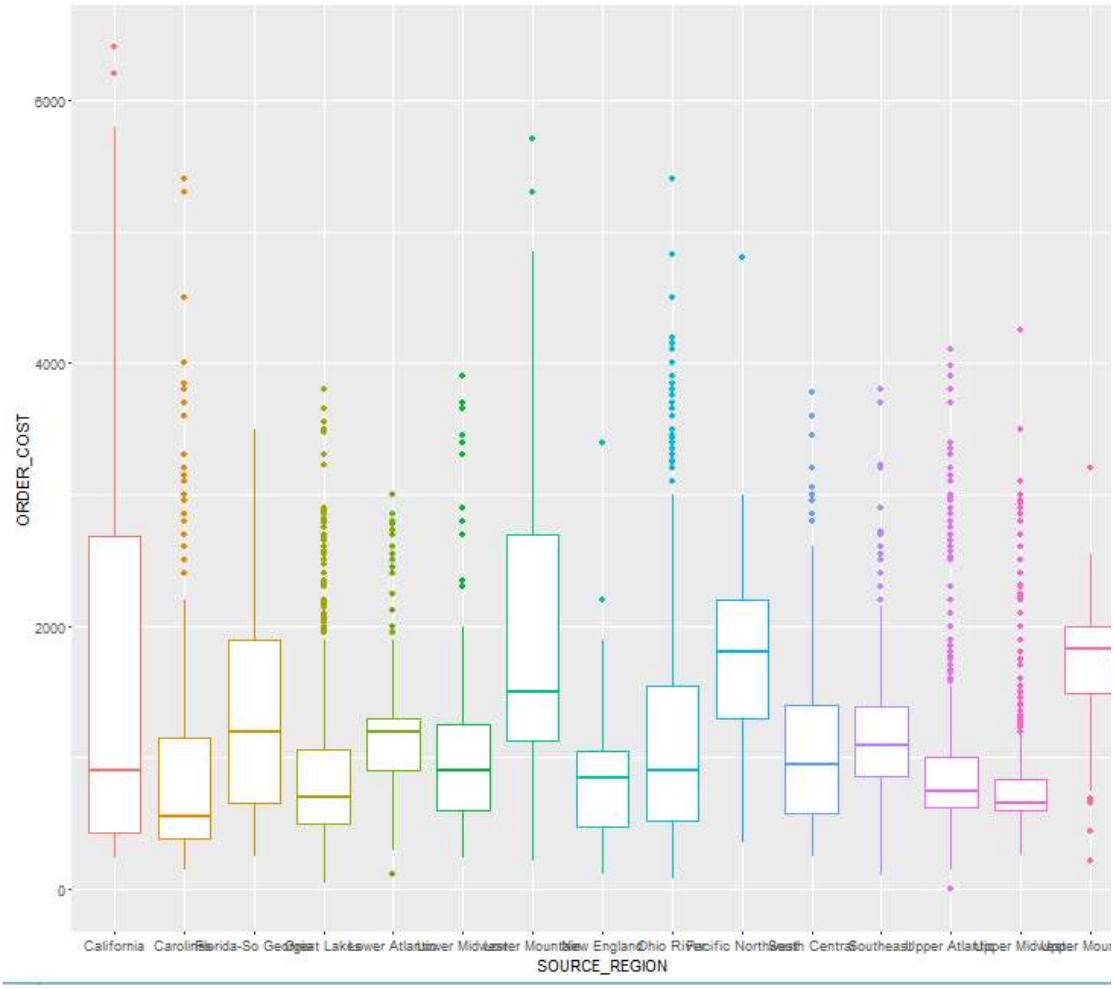
So if any order required refrigerated delivery equipment then cost of the order will be high.



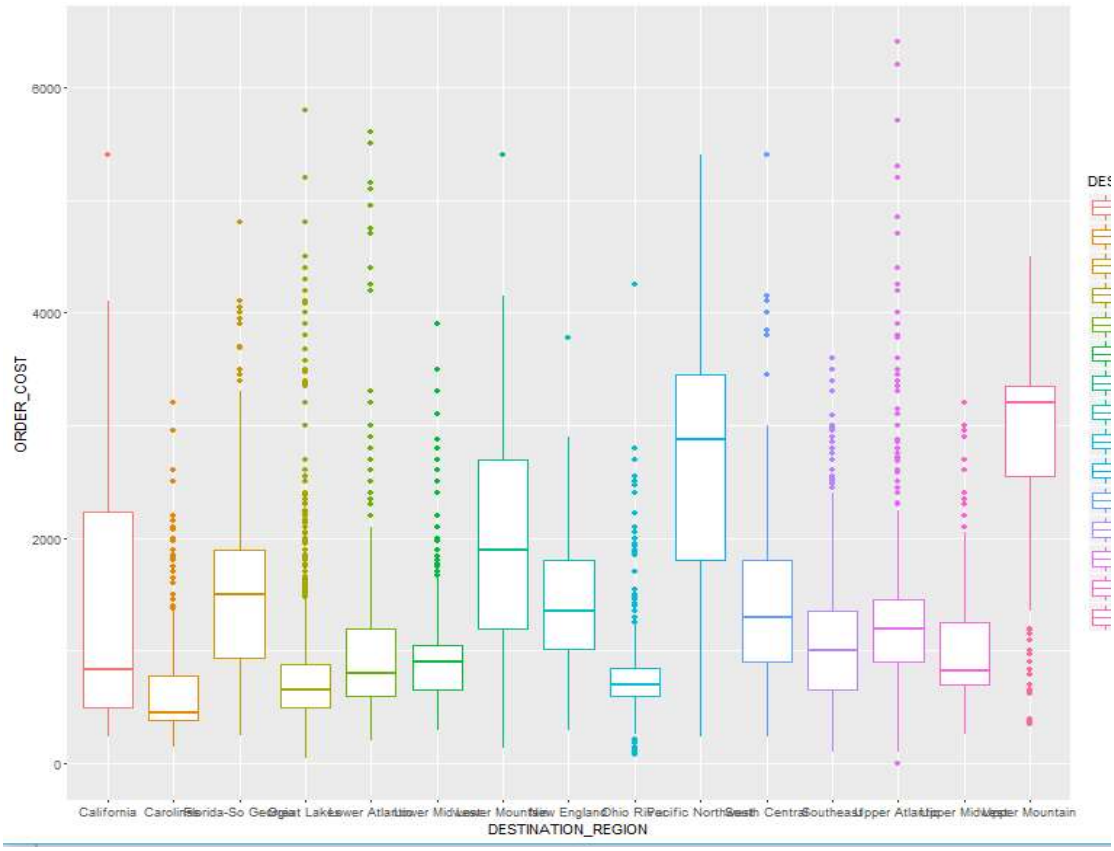
Also if the kind of item to deliver is hazardous then the cost of the order will be more.



Also we can see the cost of the order is flctuating for the source region from where the order is booked , we can see that for some region the order cost is a bit low and for some rgions the cost of the order is more than the other regions.

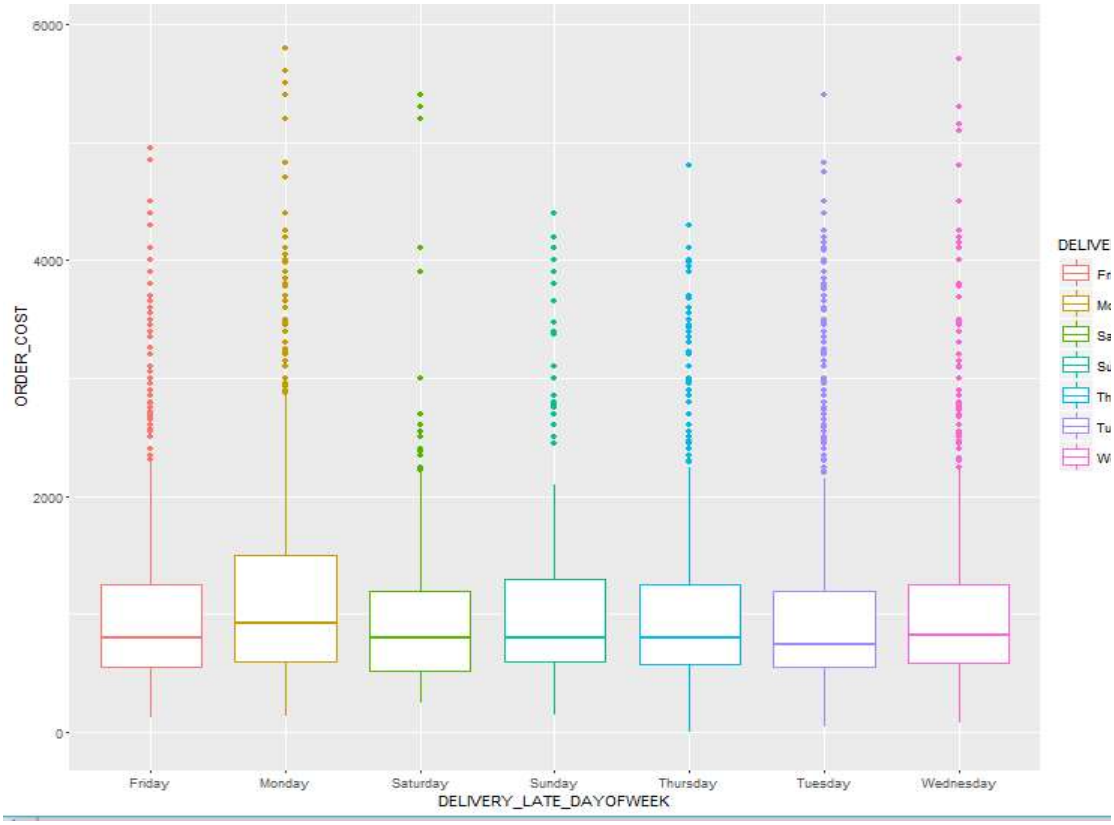


Also the destination region also impacts the cost of the order .



Also we can see the derived feature we have generated for the last day delivery late appointment (the day on which it falls).

Here we can see that if the last delivery late day is falling on Monday then the cost of the order is a bit high than for the other days.



Below given is a sample of data after all the pre processing and addition of new features.

	ORDER_NBR	EQUIPMENT_TYPE	CUSTOMER_MILES	WEIGHT	ORDER_COST	FIRST_PICK_ZIP	SOURCE_REGION
1	167159	REFRIGERATED	1832	41976	3450.00	67501	Lower Midwest
2	167160	REFRIGERATED	1136	41344	2800.00	67501	Lower Midwest
3	167161	REFRIGERATED	1832	42000	3900.00	67501	Lower Midwest
4	167162	REFRIGERATED	1832	42000	3700.00	67501	Lower Midwest
5	167163	REFRIGERATED	1744	41344	3650.00	67501	Lower Midwest
6	167164	REFRIGERATED	941	41000	1600.00	33309	Florida-So Georgia
7	167164	REFRIGERATED	941	41000	1600.00	33309	Florida-So Georgia
8	167165	DRY FREIGHT	748	21950	2400.00	61822	Great Lakes
10	167167	DRY FREIGHT	268	5000	550.00	27012	Carolinas
11	167167	DRY FREIGHT	268	5000	550.00	27012	Carolinas

FIRST_PICK_EARLY_APPT	FIRST_PICK_LATE_APPT	LAST_DELIVERY_ZIP	DESTINATION_REGION	LAST_DELIVERY_EARLY_APPT
1/6/2016 0:00	1/6/2016 0:00	98372	Pacific Northwest	1/11/2016 0:00
1/2/2016 0:00	1/3/2016 0:00	24153	Lower Atlantic	1/5/2016 0:00
1/6/2016 0:00	1/6/2016 0:00	98372	Pacific Northwest	1/8/2016 0:00
1/4/2016 0:00	1/4/2016 0:00	98372	Pacific Northwest	1/8/2016 0:00
1/4/2016 0:00	1/4/2016 0:00	97015	Pacific Northwest	1/8/2016 0:00
1/4/2016 0:00	1/4/2016 0:00	23224	Lower Atlantic	1/6/2016 0:00
1/4/2016 0:00	1/4/2016 0:00	23111	Lower Atlantic	1/6/2016 0:00
1/6/2016 0:00	1/6/2016 0:00	19522	Upper Atlantic	1/7/2016 0:00
1/4/2016 0:00	1/4/2016 0:00	22901	Lower Atlantic	1/5/2016 0:00
1/4/2016 0:00	1/4/2016 0:00	24073	Lower Atlantic	1/5/2016 0:00

LAST_DELIVERY_LATE_APPT	IS_HAZARDOUS	CREATED_DATE	time	weight_bins	DELIVERY_LATE_DAYOFWEEK
1/11/2016 0:00	N	1/1/2016 0:00	7200	1	Monday
1/5/2016 0:00	N	1/1/2016 0:00	4320	1	Tuesday
1/8/2016	N	1/1/2016 0:00	2880	1	Friday
1/8/2016 0:00	N	1/1/2016 0:00	5760	1	Friday
1/8/2016 0:00	N	1/1/2016 0:00	5760	1	Friday
1/6/2016 0:00	N	1/2/2016 0:00	2880	1	Wednesday
1/6/2016 0:00	N	1/2/2016 0:00	2880	1	Wednesday
1/7/2016 0:00	N	1/2/2016 0:00	1440	2	Thursday
1/5/2016 0:00	N	1/2/2016 0:00	1440	4	Tuesday
1/5/2016 0:00	N	1/2/2016 0:00	1440	4	Tuesday

After cleaning and pre processing of the data we have tried to build many models and see how the models are working in terms of predicting the order cost.

We have considered RMSE as the standard error factor for all our models , so we will compare the results of the models based upon that factor.

Models

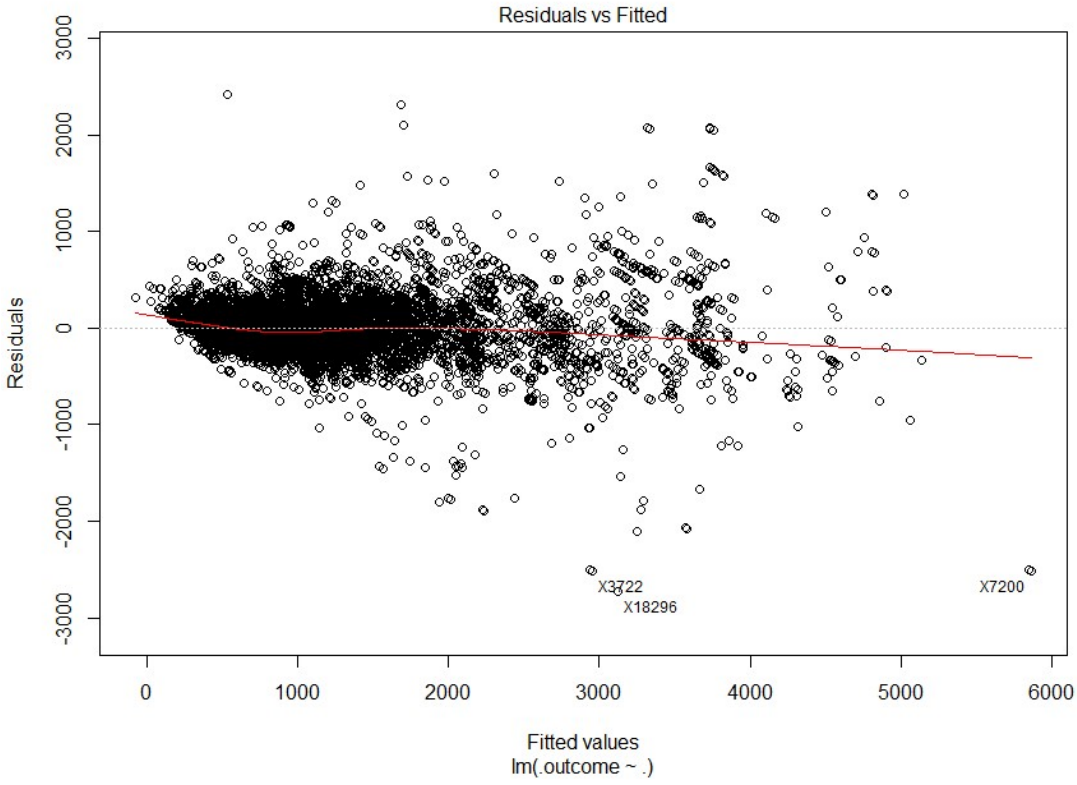
Initially we have tried to apply many kind of linear models as they are simple and fast to run and also many times very good results.

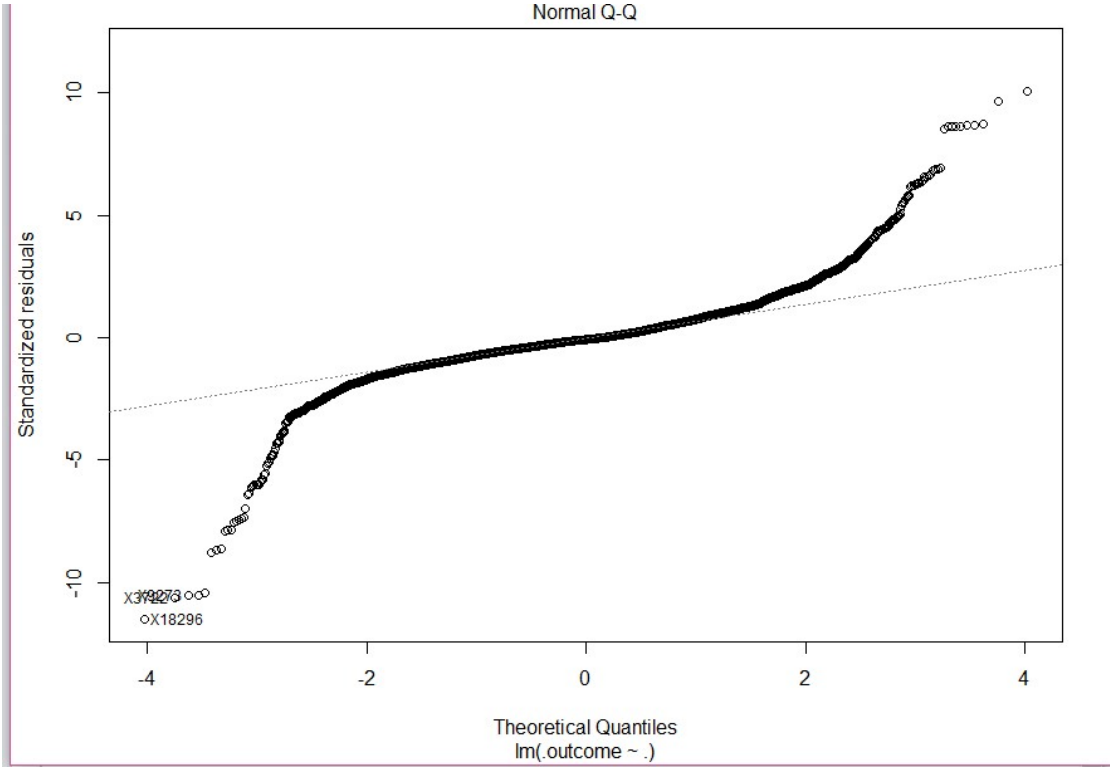
Linear Regression :

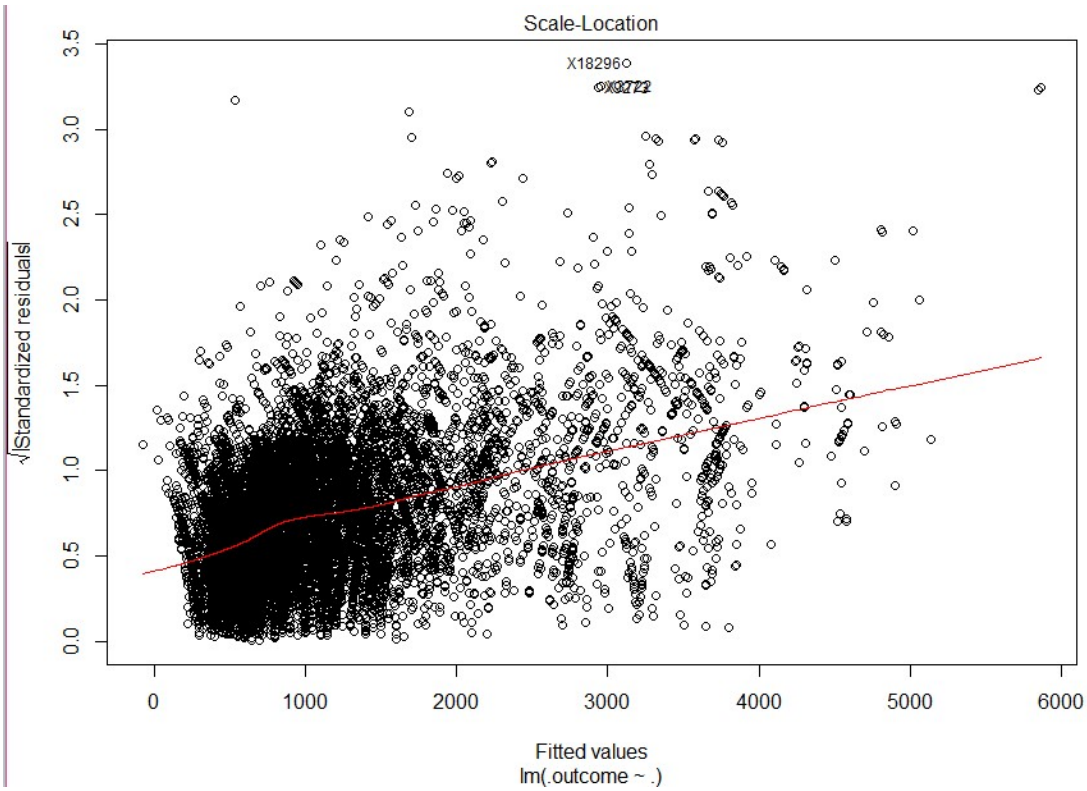
```
Linear Regression

17082 samples
   50 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 15372, 15374, 15374, 15375, 15373, 15374, ...
Resampling results:

  RMSE      Rsquared
  241.0325  0.8940527

Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
Residual standard error: 240.3 on 17038 degrees of freedom
Multiple R-squared:  0.8949,    Adjusted R-squared:  0.8946
F-statistic:  3373 on 43 and 17038 DF,  p-value: < 0.00000000000000
```

Here we see that with linear regression we are able to address 89% of variation in the data. It will give us more insight if we will plot the model and see.







If we see the Normal Q-Q plot we can see that approximately 60-70 % of the data points is falling into a linear pattern but only portion of the data not the entire , if the data is linear in

nature all the residuals will have fall across the line in the Normal Q-Q plot.

We can see there are outliers in the data also , i tried to remove some of the outiers and do the linear model again but it did not give that much of improvement to the model , the inital model was giving a RMSE of 241 with the ouliers removal it was coming around 239 not beyond that.

Let us see how the other linear models perform with the data.

```
> smp.modl_lnr[[2]]
Generalized Linear Model

17082 samples
   50 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 15372, 15374, 15374, 15375, 15373, 15374, ...
Resampling results:

  RMSE      Rsquared
  241.0325  0.8940527


The lasso

17082 samples
   50 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 15372, 15374, 15374, 15375, 15373, 15374, ...
Resampling results across tuning parameters:

  fraction  RMSE      Rsquared
  0.1       413.9289  0.8509702
  0.5       256.0333  0.8811613
  0.9       241.3579  0.8937826


Ridge Regression

17082 samples
   50 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 15372, 15374, 15374, 15375, 15373, 15374, ...
Resampling results across tuning parameters:

  lambda  RMSE      Rsquared
  0e+00   241.0325  0.8940527
  1e-04   241.0325  0.8940527
  1e-01   250.9100  0.8854593

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was lambda = 1e-04.
```

```
Elasticnet

17082 samples
   50 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 15372, 15374, 15374, 15375, 15373, 15374, .
Resampling results across tuning parameters:

  lambda  fraction  RMSE      Rsquared
  0e+00   0.050     469.1281  0.8485749
  0e+00   0.525     254.0017  0.8829740
  0e+00   1.000     241.0325  0.8940527
  1e-04   0.050     662.2362  0.8188956
  1e-04   0.525     270.7115  0.8689397
  1e-04   1.000     241.0325  0.8940527
  1e-01   0.050     665.5240  0.8188956
  1e-01   0.525     276.2166  0.8638023
  1e-01   1.000     250.9100  0.8854593

RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were fraction = 1 and lambda = 1
```

```
glmnet

17082 samples
   50 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 15372, 15374, 15374, 15375, 15373, 15374, ...
Resampling results across tuning parameters:

  alpha  lambda      RMSE      Rsquared
  0.10     1.339212  241.0613  0.8940364
  0.10    13.392123  242.1919  0.8933682
  0.10   133.921227  281.0426  0.8733880
  0.55     1.339212  241.0818  0.8940175
  0.55    13.392123  246.4375  0.8899374
  0.55   133.921227  325.1931  0.8349067
  1.00     1.339212  241.2056  0.8939174
  1.00    13.392123  253.2529  0.8839927
  1.00   133.921227  342.3870  0.8188956

RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were alpha = 0.1 and lambda = 1.33
```

I have also tried a generalized linear model with stepwise feature selection , let us see how it performs.

```
Generalized Linear Model with Stepwise Feature Selection

17082 samples
   50 predictor

No pre-processing
Resampling: Bootstrapped (5 reps)
Summary of sample sizes: 17082, 17082, 17082, 17082, 17082
Resampling results:

  RMSE      Rsquared
  243.8584  0.892205
```

Almost same as linear regression. We can also see the significant features from the data for this model.

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	636.061	28.968	21.957	< 0.000000000000000
CUSTOMER_MILES	5715.052	24.467	233.585	< 0.000000000000000
time	-568.340	34.436	-16.504	< 0.000000000000000
IS_HAZARDOUS.N	-227.845	18.397	-12.385	< 0.000000000000000
`EQUIPMENT_TYPE.DRY FREIGHT`	-221.164	6.120	-36.137	< 0.000000000000000
SOURCE_REGION.California	630.346	23.257	27.103	< 0.000000000000000
SOURCE_REGION.Carolinas	391.991	21.478	18.250	< 0.000000000000000
`SOURCE_REGION.Florida-So Georgia`	269.549	24.000	11.231	< 0.000000000000000
`SOURCE_REGION.Great Lakes`	483.476	20.445	23.647	< 0.000000000000000
`SOURCE_REGION.Lower Atlantic`	402.983	20.816	19.359	< 0.000000000000000
`SOURCE_REGION.Lower Midwest`	462.138	22.673	20.382	< 0.000000000000000
`SOURCE_REGION.Lower Mountain`	366.826	22.732	16.137	< 0.000000000000000
`SOURCE_REGION.Ohio River`	577.537	20.659	27.956	< 0.000000000000000
`SOURCE_REGION.Pacific Northwest`	446.002	29.478	15.130	< 0.000000000000000
`SOURCE_REGION.South Central`	215.019	22.204	9.684	< 0.000000000000000
SOURCE_REGION.Southeast	333.988	21.777	15.336	< 0.000000000000000
`SOURCE_REGION.Upper Atlantic`	223.559	20.509	10.900	< 0.000000000000000
`SOURCE_REGION.Upper Midwest`	495.279	21.322	23.228	< 0.000000000000000
DESTINATION_REGION.California	-582.138	12.956	-44.932	< 0.000000000000000
DESTINATION_REGION.Carolinas	-323.676	10.100	-32.046	< 0.000000000000000
`DESTINATION_REGION.Florida-So Georgia`	-37.078	11.553	-3.209	0.00133
`DESTINATION_REGION.Great Lakes`	-430.703	6.703	-64.254	< 0.000000000000000
`DESTINATION_REGION.Lower Atlantic`	-201.691	9.063	-22.254	< 0.000000000000000
`DESTINATION_REGION.Lower Midwest`	-407.506	10.886	-37.433	< 0.000000000000000
`DESTINATION_REGION.Lower Mountain`	-149.545	14.221	-10.516	< 0.000000000000000
`DESTINATION_REGION.New England`	175.403	18.344	9.562	< 0.000000000000000
`DESTINATION_REGION.Ohio River`	-331.434	7.921	-41.842	< 0.000000000000000
`DESTINATION_REGION.Pacific Northwest`	-320.905	19.393	-16.547	< 0.000000000000000
`DESTINATION_REGION.South Central`	-231.047	9.254	-24.966	< 0.000000000000000
DESTINATION_REGION.Southeast	-304.196	10.464	-29.070	< 0.000000000000000
`DESTINATION_REGION.Upper Midwest`	-412.277	10.780	-38.244	< 0.000000000000000
DELIVERY_LATE_DAYOFWEEK.Friday	-18.384	5.207	-3.530	0.00041

```
DELIVERY_LATE_DAYOFWEEK.Monday      21.409     5.592   3.828          0.00013(
DELIVERY_LATE_DAYOFWEEK.Sunday      19.515    10.440   1.869          0.06160!
DELIVERY_LATE_DAYOFWEEK.Thursday   -14.661     5.183  -2.829          0.00467(
SRC_DST_MTCH.0                     -35.752     5.184  -6.896  0.0000000000055!
weight_bins.2                      -13.690     4.327  -3.164          0.00155!
weight_bins.3                      -15.668     6.197  -2.528          0.01147(
---
```

When i compare the model with the linear regression i saw that the step wise feature selection model has dropped all the in-significant features.

For example for the weight_bins feature if we see the linear model has :

```
weight_bins.1                        1.359     6.026   0.226          0.8215!
weight_bins.2                      -13.086     4.658  -2.810          0.0049!
weight_bins.3                      -14.794     6.466  -2.288          0.0221!
weight_bins.4                           NA        NA      NA              NA
---
```

And we can see weight_bins.1 and weight_bins.4 got dropped from the step wise feature selection model.

Apart from these also i have tried many more models , details of which are given below.

```
k-Nearest Neighbors

17082 samples
   50 predictor

No pre-processing
Resampling: Bootstrapped (10 reps)
Summary of sample sizes: 17082, 17082, 17082, 17082, 17082, 17082,
Resampling results across tuning parameters:

  k  RMSE      Rsquared
  5  222.3520  0.9087999
  7  234.0489  0.8999089
  9  243.2267  0.8924604

RMSE was used to select the optimal model using  the smallest value
The final value used for the model was k = 5.
```

```
k-Nearest Neighbors

17082 samples
   50 predictor

No pre-processing
Resampling: Bootstrapped (10 reps)
Summary of sample sizes: 17082, 17082, 17082, 17082, 17082, 17082, ...
Resampling results across tuning parameters:

  kmax  RMSE      Rsquared
  5     181.4234  0.9304426
  7     181.4234  0.9304426
  9     181.4234  0.9304426

Tuning parameter 'distance' was held constant at a value of 2
Tuning parameter 'kernel' was
 held constant at a value of optimal
RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were kmax = 9, distance = 2 and kerne
```

SVM Models

```
Support Vector Machines with Linear Kernel

17082 samples
   50 predictor

No pre-processing
Resampling: Bootstrapped (3 reps)
Summary of sample sizes: 17082, 17082, 17082
Resampling results across tuning parameters:

  cost  RMSE      Rsquared
  0.25  246.7092  0.8900623
  0.50  246.7057  0.8900647
  1.00  246.6993  0.8900669

RMSE was used to select the optimal model using  the smallest va
The final value used for the model was cost = 1.


 Parameters:
    SVM-Type:  eps-regression
  SVM-Kernel:  linear
        cost:  1
       gamma:  0.02
     epsilon:  0.1


 Number of Support Vectors:  10396
```

```
Support Vector Machines with Radial Basis Function Kernel

17082 samples
   50 predictor

No pre-processing
Resampling: Bootstrapped (3 reps)
Summary of sample sizes: 17082, 17082, 17082
Resampling results across tuning parameters:

  C     RMSE      Rsquared
  0.25  210.6325  0.9199417
  0.50  198.5872  0.9282783
  1.00  189.8712  0.9340911

Tuning parameter 'sigma' was held constant at a value of 0.0137906
RMSE was used to select the optimal model using  the smallest valu
The final values used for the model were sigma = 0.01379063 and C
```

CART Models

One thing i have observed that when we apply CART models to data which is standaridzed , which normally is a requirement for neural net or svm or models like lasso or ridge or elastic net , in that type of data CART models dont work dont perform that much better.

When we apply the models to normal data which is not standardized they yield good result , we will see that when i will finally present all the models and its performance.

```
CART

17424 samples
    9 predictor

No pre-processing
Resampling: Cross-Validated (12 fold)
Summary of sample sizes: 15972, 15972, 15972, 15972, 15971, 15973, .
Resampling results:

  RMSE      Rsquared
  298.9107  0.8342482

Tuning parameter 'maxdepth' was held constant at a value of 9
```

```
n= 17424

node), split, n, deviance, yval
      * denotes terminal node

 1) root 17424 9396308000 1051.6180
   2) CUSTOMER_MILES< 1044 15577 2792754000  865.9620
     4) CUSTOMER_MILES< 489.5 9702  566572200  639.4684
       8) CUSTOMER_MILES< 296.5 5150  120675800  503.6246 *
       9) CUSTOMER_MILES>=296.5 4552  243340100  793.1580 *
     5) CUSTOMER_MILES>=489.5 5875  906560400 1239.9950
      10) DESTINATION_REGION=California,Carolinas,Great Lakes,Lower Midwest,Ohio River,So
dwest,Upper Mountain 3342  317751100 1064.5940 *
      11) DESTINATION_REGION=Florida-So Georgia,Lower Atlantic,Lower Mountain,New England
st,South Central,Upper Atlantic 2533  350334400 1471.4160
        22) CUSTOMER_MILES< 661 1148   56066980 1233.3140 *
        23) CUSTOMER_MILES>=661 1385  175238800 1668.7730 *
   3) CUSTOMER_MILES>=1044 1847 1538508000 2617.3820
     6) CUSTOMER_MILES< 1712.5 1148  296368500 2113.1900 *
     7) CUSTOMER_MILES>=1712.5 699  471016600 3445.4400
      14) SOURCE_REGION=Florida-So Georgia,Great Lakes,Lower Atlantic,Pacific Northwest,S
theast,Upper Atlantic,Upper Midwest,Upper Mountain 240   97248440 2851.0770 *
      15) SOURCE_REGION=California,Carolinas,Lower Midwest,Lower Mountain,New England,Ohi
652600 3756.2170 *
> |
```

Random Forest

```
Random Forest

17424 samples
    9 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 17424, 17424, 17424, 17424, 17424, 17424,
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared
  2     188.5285  0.9380094
  5     156.3990  0.9549429
  9     160.1004  0.9528572

RMSE was used to select the optimal model using  the smallest valu
The final value used for the model was mtry = 5.
```

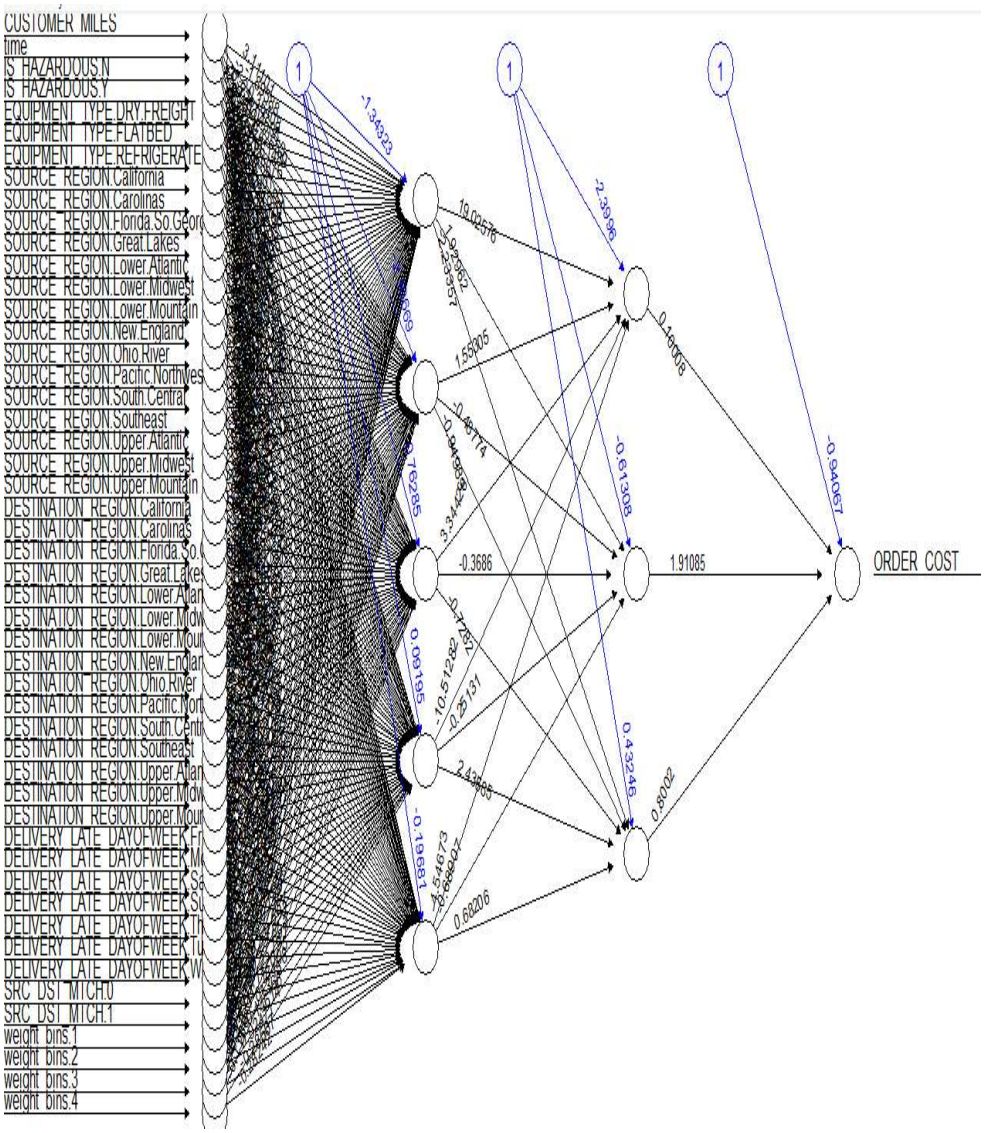
```
Type:                             Regression
Number of trees:                  500
Sample size:                      17424
Number of independent variables:  9
Mtry:                             5
Target node size:                 5
Variable importance mode:         none
OOB prediction error:             20947.28
R squared:                        0.9611587
```

Cubist

The tree-based Cubist model can be easily used to develop an ensemble classifier with a scheme called “committees”. The concept of “committees” is similar to the one of “boosting” by developing a series of trees sequentially with adjusted weights. However, the final prediction is the simple average of predictions from all “committee” members, an idea more close to **“bagging”.**

```
Cubist

17424 samples
    9 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 15683, 15683, 15680, 15683, 15682, 15682, ...
Resampling results across tuning parameters:

  committees  neighbors  RMSE      Rsquared
   1          0          198.2241  0.9273815
   1          5          173.0850  0.9446081
   1          9          176.1724  0.9425747
  10          0          181.8927  0.9390381
  10          5          159.6804  0.9526754
  10          9          162.4157  0.9510284
  20          0          181.2580  0.9395297
  20          5          158.8638  0.9531811
  20          9          161.4789  0.9516194

RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were committees = 20 and neighbors =
```

Finally i have tried a neural net model also with 2 hidden layers with 5 and 3 nodes in the hidden layers.



Another thing my mentor's want me to try to see if we can have some kind of time series model apply to the data.

The data available to us do not fall to time series data pattern in any way , most of the data we have is from 2016 Jan to 2016 May.

Also i was not able to find any pattern in the data which can be break to weekly data or monthly data , or quarterly , yearly data is not possible in any way.

For source region Carolinas and Destination region Lower Atlantic i got 338 observations in the data set.

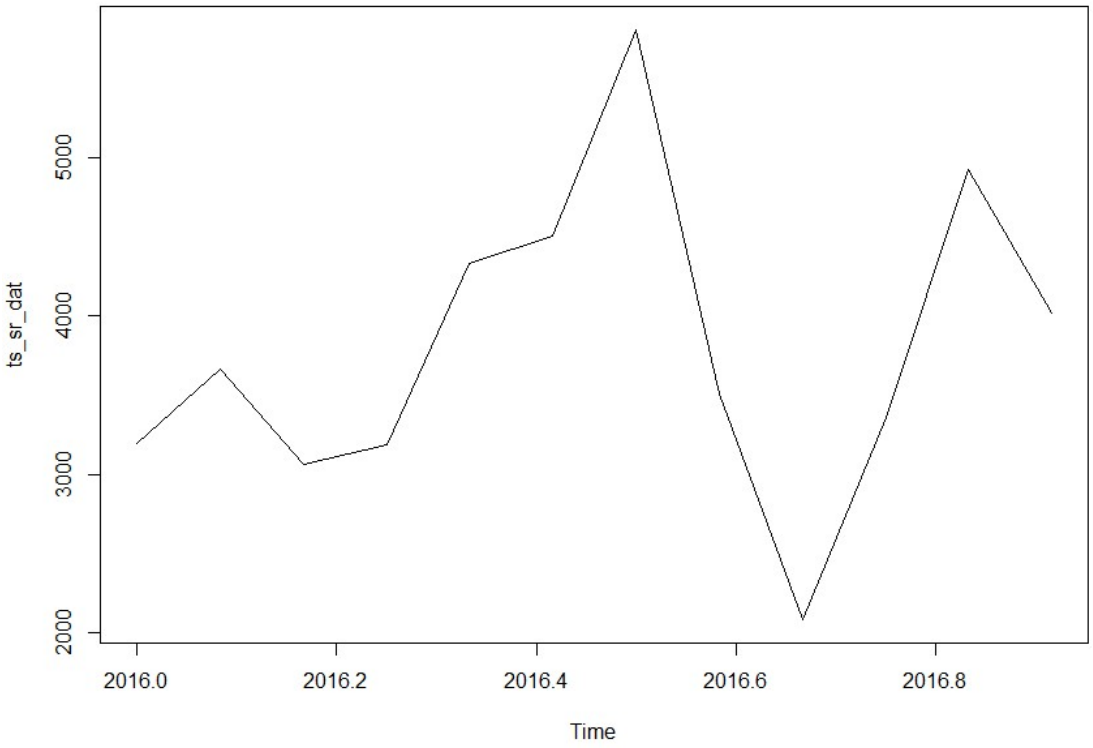
Then i have converted the data avaiable based upon the created date. So if on Jan 1st there are 4 orders then we took the mean of the data and arrived to another dataset where each record belongs to one particular date.

Then we convert that data to weekly basis and we got 21 weeks of data.

	mn4	unqwks
2	3194.127	01
3	3666.667	02
4	3066.667	03
5	3188.889	04
6	4333.333	05
7	4500.000	06
8	5800.000	07
9	3509.091	08
10	2090.000	09
11	3360.000	10
12	4925.000	11
13	4014.286	13
14	3380.000	14
15	3297.619	15
16	3946.667	16
17	2957.333	17
18	4033.333	18
19	4100.000	19
20	4326.667	20
21	4061.905	21

Now as the data is not falling to weekly data also , we can not create a time series data from this data set , as we need at least 52 or 53 week of data to create the weekly time series object.

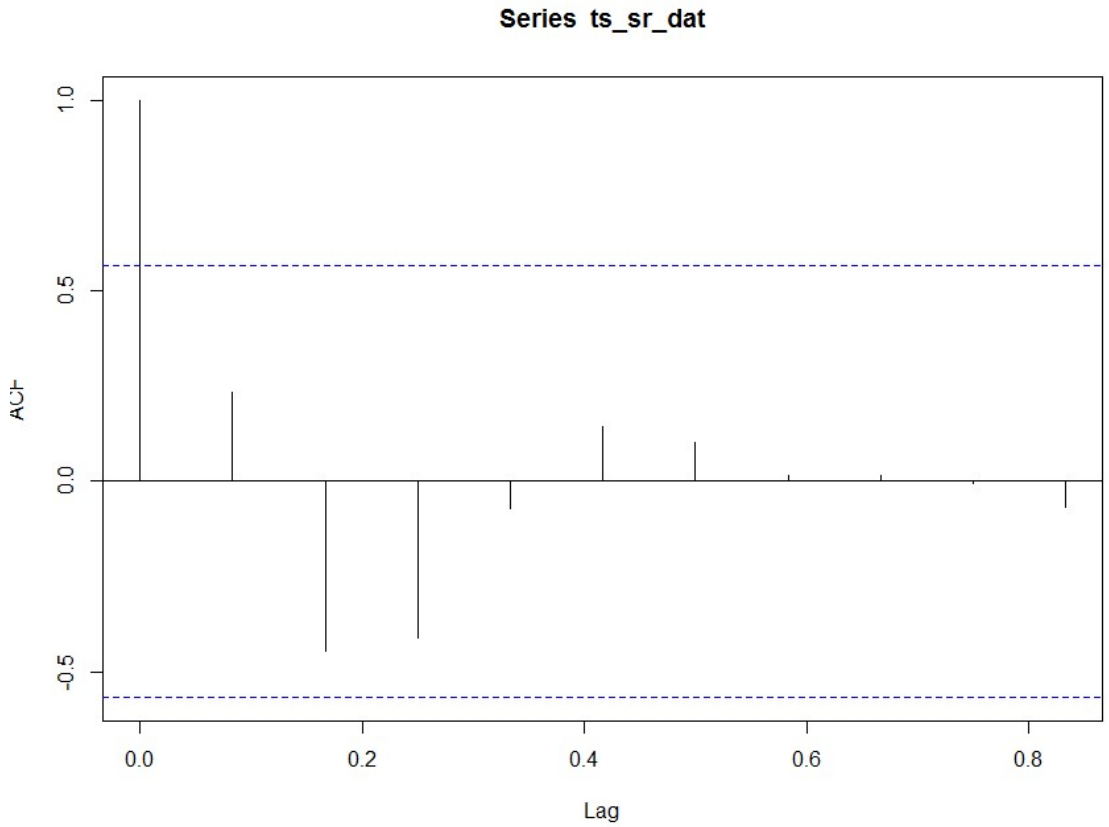
Here i have tried to assume that lets say these are monthly data (by multiplying the weeky order cost with 4) and take the first 12 data point as the 12 months of data of 2016 and then try to create a time series object.

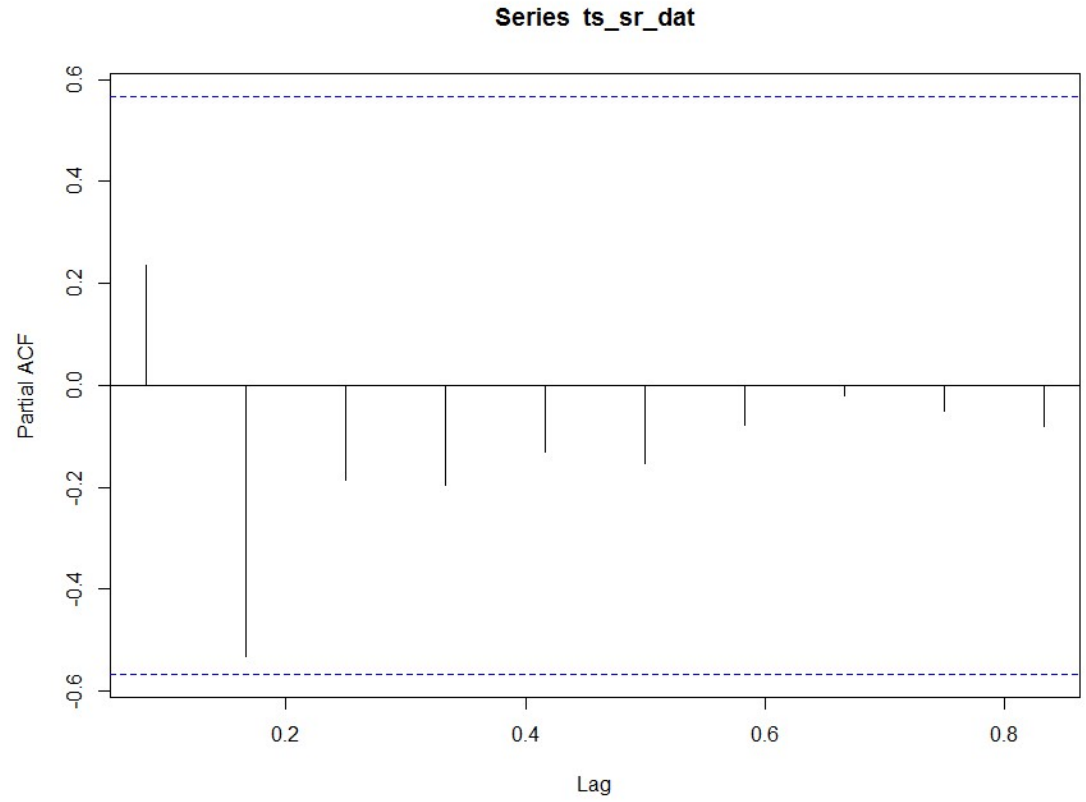


Here we can see that the data actually do not have any kind of trend , or seasonlatiy in it , basically not a good time series object , but still we go ahead with our analysis.

When i tried to decompose the data i got the error like below.

Error in decompose(ts_sr_dat) : time series has no or less than 2 periods


Series ts_sr_dat
ACF
1.0
0.5
0.0
-0.5
0.0
0.2
0.4
0.6
0.8
Lag


Series ts_sr_dat
Partial ACF
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
0.2
0.4
0.6
0.8
Lag

Created one ARIMA model with the order of (2,3,2)

Then try to forecast for 2 periods.

```
         Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
Jan 2017      -648.3222 -3311.597 2014.953 -4721.450 3424.805
Feb 2017       192.3630 -3898.134 4282.860 -6063.512 6448.238
> 
```

So if we see the 13th and 14th data point order cost value in our dataset we see the values as

3380.000

3297.619

Surprisingly if we see the high value for 95% confodence the value is 3424 comparing to the actual data of 3380 , the second prediction was not good .

We have just tried to create the time series model for an experiment purpose , so the approach we have taken here is not perfect , but just with some approximation.

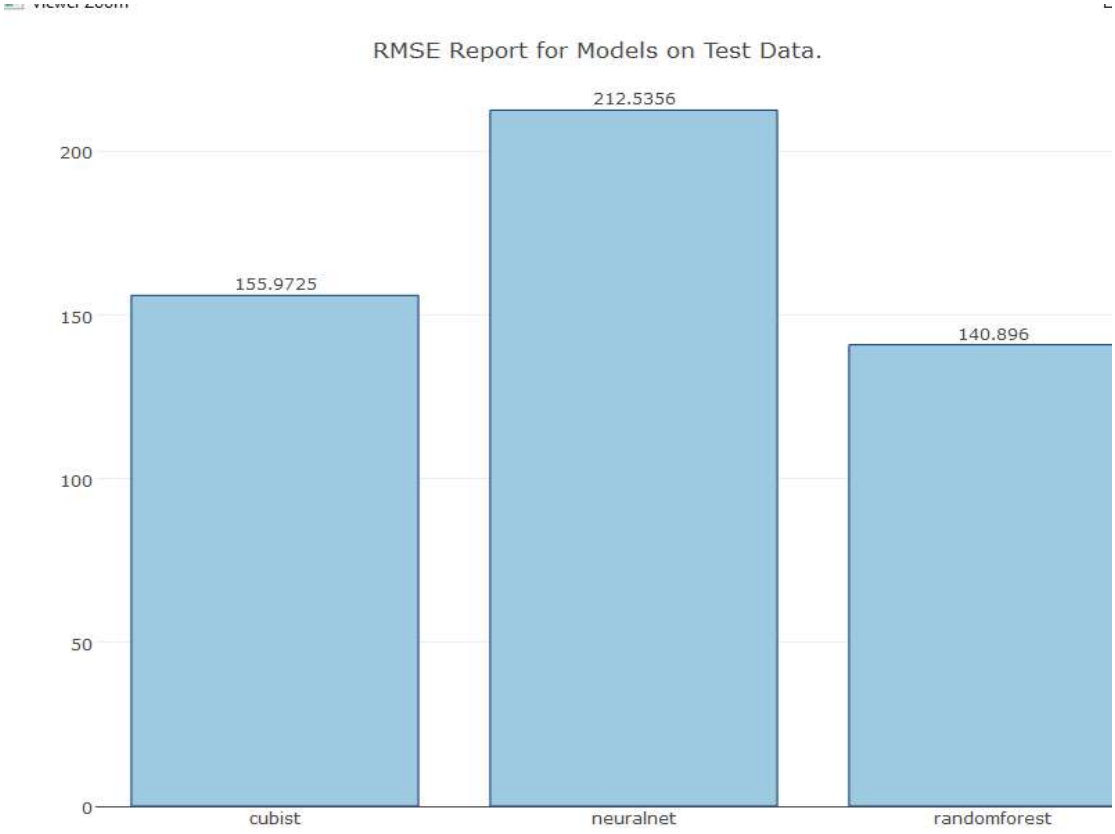
Now lets see the results of all the models.

	Model Name	Long Model Name	R Square	RMSE	Time Taken for the model to build	Test Data RMSE	Evaluation Data RMSE
1	lm	Linear Regression	0.8940527	241.0325	29.31	337.0747	235.9607
2	glm	Generalized Linear Model	0.8940527	241.0325	46.15	337.0747	236.1105
3	lasso	The lasso	0.8937826	241.3579	279.24	336.3950	236.6273
4	enet	Elasticnet	0.8940527	241.0325	719.36	337.0701	236.1044
5	ridge	Ridge Regression	0.8940527	241.0325	260.61	337.0701	236.1044
6	glmnet	glmnet	0.8940364	241.0613	19.77	336.9175	235.8942
7	glmStepAIC	Generalized Linear Model with Stepwise Feature Selection	0.8922050	243.8584	720.79	336.9175	235.9607
8	knn	k-Nearest Neighbors	0.9087999	222.3520	711.73	256.5351	225.9766
9	kknn	k-Nearest Neighbors	0.9304426	181.4234	951.68	261.0177	229.4603
10	svmLinear2	Support Vector Machines with Linear Kernel	0.8900669	246.6993	3888.44	322.2033	236.5036
11	svmRadial	Support Vector Machines with Radial Basis Function Ker...	0.9340911	189.8712	1103.21	264.3083	182.0877
12	rpart	CART(Standardized data)	0.7212036	389.8925	21.31	487.9340	398.1529
13	rpart2	CART(Standardized data)	0.7998123	331.2089	11.70	412.4964	330.6460
14	ranger	Random Forest(Standardized Data)	0.9502543	165.6458	140.33	313.7742	158.4808
15	rpart2	CART	0.8342482	298.9107	1.44	295.3180	318.8220
16	cubist	Cubist	0.9526700	159.6804	331.64	155.9725	164.0781
17	ranger	Random Forest	0.9549429	156.3990	1370.63	140.8960	152.6670
18	neuralnet	Neural Net	NA	184.2912	NA	212.5356	188.0220

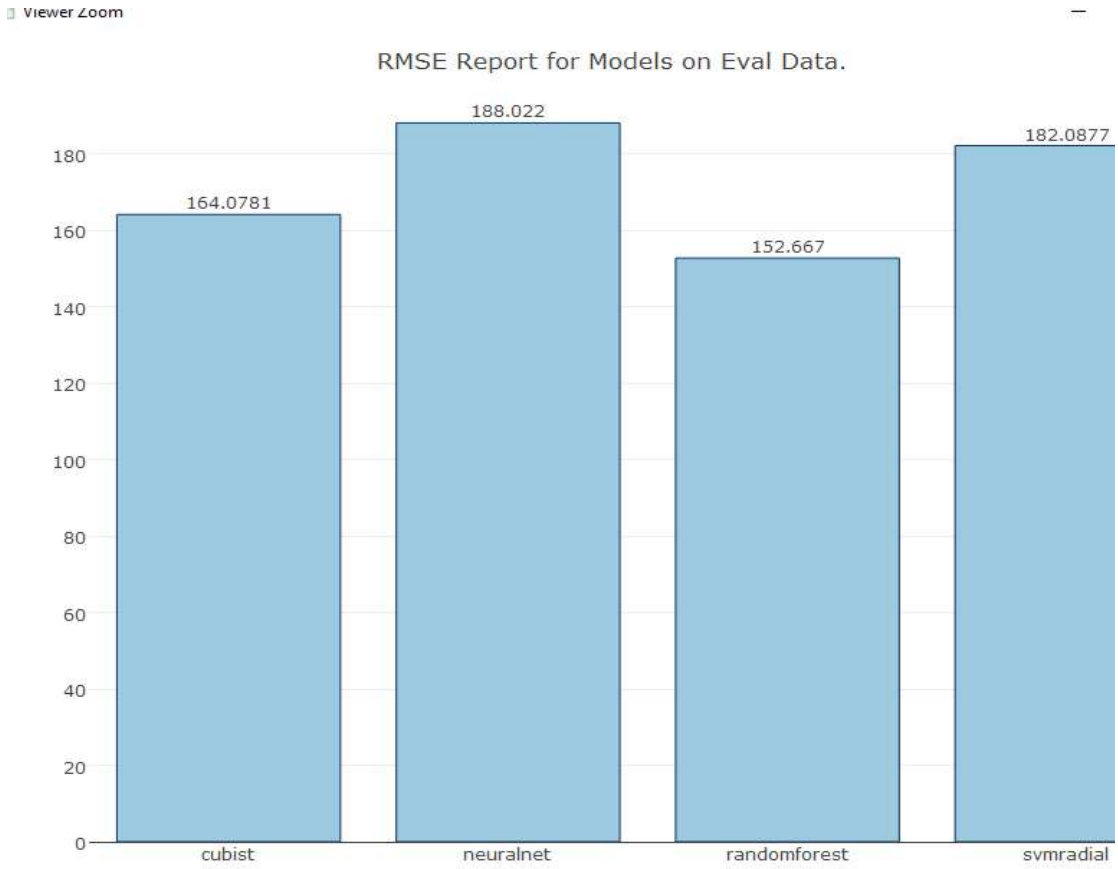
Below i have given the test results for different models on training , testing and evaluation data. Here we can see the for the Cubist and random forest model(for the model with no standardized data) the evaluation data set RMSE is slightly greater than the test dataset RMSE , all other models evaluation RMSE is lower than the test dataset RMSE.

	Model	Train.RMSE	Test.RMSE	Eval.RMSE
1	lm	239.9650	337.0747	235.9607
2	glm	239.9660	337.0747	236.1105
3	lasso	240.5270	336.3950	236.6273
4	enet	239.9660	337.0701	236.1044
5	ridge	239.9660	337.0701	236.1044
6	glmStepAIC	239.9990	336.9170	235.9607
7	knn	178.9501	256.5350	225.9766
8	kknn	78.3570	261.0177	229.4603
9	svmlinear	243.9202	322.2033	236.5036
10	svmradial	171.8322	264.3080	182.0877
11	rpart(std)	417.0812	487.9340	398.1520
12	rpart2(std)	328.9266	412.4964	330.6461
13	randomforest(std)	84.6870	313.7740	158.4808
14	rpart2	298.3870	295.3158	318.8211
15	cubist	114.8475	155.9725	164.0781
16	randomforest	79.1250	140.8490	152.6620
17	neuralnet	184.2912	212.5356	188.0220

Out of all these models the top three models for the test data evaluation.



The top 4 models on evaluation data.



So we can say that Neural net , Cubist , Random Forest and even K-Nearest Neighbors(kknn) , SVM with a radia basis Kernel given us good result , random forest and cubist are the optimum ones.

There are obvious scope of improvements also present , where we can try to stack different models and see how the combination of models perform.

Also we can think about new features of the data with more domain knowldge to see if we can come up with any new derived attributes which can help us in better prediction.