Project 4 - Regression Analysis (Part 1)

- Gaurav Singh
- UID: 305353434

Library imports

In [140]:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler

import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.feature_selection import mutual_info_regression, f_regression,SelectKBe
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.model_selection import cross_validate

from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV

from statsmodels.regression.linear_model import OLS
from sklearn.preprocessing import PolynomialFeatures

from sklearn.neural_network import MLPRegressor
from itertools import combinations_with_replacement

from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import export_graphviz
import pydot
import errno
from IPython.display import Image
from skimage.io import imread, imshow

import lightgbm as lgb
from skopt import BayesSearchCV
from skopt.space import Real, Categorical, Integer
```

In [4]:

```
# Reading the dataset

diamondsData = pd.read_csv('diamonds.csv')
```

In [5]:

```
diamondsData = diamondsData.drop(columns=['Unnamed: 0'])
```

In [6]:

```
# Preparing gas emission data
gt2011 = pd.read_csv('./pp_gas_emission/gt_2011.csv')
gt2012 = pd.read_csv('./pp_gas_emission/gt_2012.csv')
gt2013 = pd.read_csv('./pp_gas_emission/gt_2013.csv')
gt2014 = pd.read_csv('./pp_gas_emission/gt_2014.csv')
gt2015 = pd.read_csv('./pp_gas_emission/gt_2015.csv')
```

Concatenating all gas emission dataset with year as a feature

In [7]:

```
gt2011['year'] = 2011
gt2012['year'] = 2012
gt2013['year'] = 2013
gt2014['year'] = 2014
gt2015['year'] = 2015

emissionData = pd.concat([gt2011, gt2012, gt2013, gt2014, gt2015], axis=0).reset_inc
```

In [8]:

```
# Dropping NOX gas emission
emissionData = emissionData.drop(columns=['NOX'])
```

In [9]:

```
# Previewing some data points

emissionData.sample(5)
```

Out[9]:

	AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	year
20773	24.774	1018.8	69.222	4.0459	31.246	1100.1	537.32	150.52	13.443	0.35307	2013
4906	25.336	1010.2	74.826	4.0052	25.980	1092.4	550.07	133.76	12.036	0.93695	2011
19922	26.454	1012.6	67.676	3.9059	20.266	1059.2	550.06	109.60	10.547	3.95000	2013
33660	31.171	1009.1	52.748	4.4000	30.050	1100.1	545.32	146.82	13.160	1.25450	2015
3921	20.797	1008.3	89.461	3.1164	20.392	1061.6	549.96	113.91	10.727	1.99450	2011

In [10]:

```
diamondsData.sample(5)
```

Out[10]:

	carat	cut	color	clarity	depth	table	price	x	y	z
26889	2.06	Premium	F	SI2	61.7	59.0	16859	8.21	8.13	5.04
5471	1.01	Premium	F	SI2	59.2	58.0	3839	6.50	6.47	0.00
39212	0.40	Very Good	E	VVS2	62.8	58.0	1070	4.68	4.71	2.95
38529	0.34	Ideal	E	VS1	61.2	57.0	1033	4.51	4.45	2.74
28410	0.35	Ideal	D	SI2	61.8	57.0	673	4.57	4.53	2.81

Preprocessing of the datasets

Question 1

In [11]:

```
# Standardization of emission data
scaler = StandardScaler()
contCols = ['AT', 'AP', 'AH', 'AFDP', 'GTEP', 'TIT', 'TAT', 'TEY', 'CDP', 'year']
contVarsEmission = emissionData[['AT', 'AP', 'AH', 'AFDP', 'GTEP', 'TIT', 'TAT', 'TE
scaledEmission = contVarsEmission.copy()
targetEmission = emissionData['CO']
scaledEmission[contCols] = scaler.fit_transform(contVarsEmission)
```

In [12]:

```
# Checking the rating based on cut
print("Fair Cut::::")
print(diamondsData[diamondsData['cut'] == 'Fair']['price'].describe())

print("Good Cut::::")
print(diamondsData[diamondsData['cut'] == 'Good']['price'].describe())

print("Very Good Cut::::")
print(diamondsData[diamondsData['cut'] == 'Very Good']['price'].describe())

print("Premium Cut::::")
print(diamondsData[diamondsData['cut'] == 'Premium']['price'].describe())

print("Ideal Cut::::")
print(diamondsData[diamondsData['cut'] == 'Ideal']['price'].describe())
```

```
Fair Cut::::
count     1610.000000
mean      4360.767702
std       3560.422751
min        338.000000
25%       2053.000000
50%       3285.000000
75%       5208.500000
max      18574.000000
Name: price, dtype: float64
Good Cut::::
count     4906.000000
mean      3930.878924
std       3681.582370
min        328.000000
25%       1146.250000
50%       3054.000000
75%       5029.750000
max      18792.000000
Name: price, dtype: float64
Very Good Cut::::
count    12082.000000
mean      3983.770816
std       3935.864776
min        336.000000
25%        913.000000
50%       2648.500000
75%       5375.000000
max      18818.000000
Name: price, dtype: float64
Premium Cut::::
count    13791.000000
mean      4586.261402
std       4349.203184
min        327.000000
25%       1048.500000
50%       3188.000000
75%       6297.000000
max      18823.000000
Name: price, dtype: float64
Ideal Cut::::
count    21551.000000
mean      3459.534036
```

```
std       3808.405240
min        330.000000
25%        880.000000
50%       1812.000000
75%       4681.000000
max      18808.000000
Name: price, dtype: float64
```

In [13]:

```
# Encoding and standardization of diamonds data
cutDict = {
    'Fair': 4,
    'Good': 2,
    'Very Good': 3,
    'Premium': 5,
    'Ideal': 1
}

colorDict = {
    'D': 7,
    'E': 6,
    'F': 5,
    'G': 4,
    'H': 3,
    'I': 2,
    'J': 1
}

clarityDict = {
    'SI1': 3,
    'VS2': 4,
    'SI2': 2,
    'VS1': 5,
    'VVS2': 6,
    'VVS1': 7,
    'IF': 8,
    'I1': 1
}
```

In [14]:

```
diamondsScaled = diamondsData.copy()
diamondsScaled['cut'] = diamondsData['cut'].apply(lambda x: cutDict[x])
diamondsScaled['color'] = diamondsData['color'].apply(lambda x: colorDict[x])
diamondsScaled['clarity'] = diamondsData['clarity'].apply(lambda x: clarityDict[x])
diamondsScaled = diamondsScaled.drop(columns=['price'])

diamondsEncoded = diamondsScaled.copy()

contVarsDiamond = ['carat', 'cut', 'color', 'clarity', 'depth', 'table', 'x', 'y', '
targetDiamonds = diamondsData['price']
diamondScaler = StandardScaler()

diamondsScaled[contVarsDiamond] = diamondScaler.fit_transform(diamondsScaled)
```

Answer 1

Both the datasets - Diamonds and CO emission ones were Standardized for further question. StandardScaler is used.

$$R_X = \frac{X - \mu_X}{\sigma_X}$$

Benefits of standardisation:

- Reduces skewness.
- Faster convergence.
- Makes predictions less sensitive to outliers

Before standardisation "Categorical" features were mapped to numbers and encoding the relative stregths of these features using increasing order of numbers of representation.

In CO emission dataset there were no categrorical values except the year value which was standardised as well.

The target variables were not scaled and later we see that this is not making much impact. In diamonds dataset: Features [cut, color, clarity] were label encoded keeping relative strengths intact.

Data inspection for Diamonds dataset

In [15]:

```
print("No of rows in Diamonds data having NaN values: {}". format(len(diamondsData)
```

```
No of rows in Diamonds data having NaN values: 0
```

In [16]:

```
print("No of rows in Emission data having NaN values: {}". format(len(emissionData)
```

```
No of rows in Emission data having NaN values: 0
```

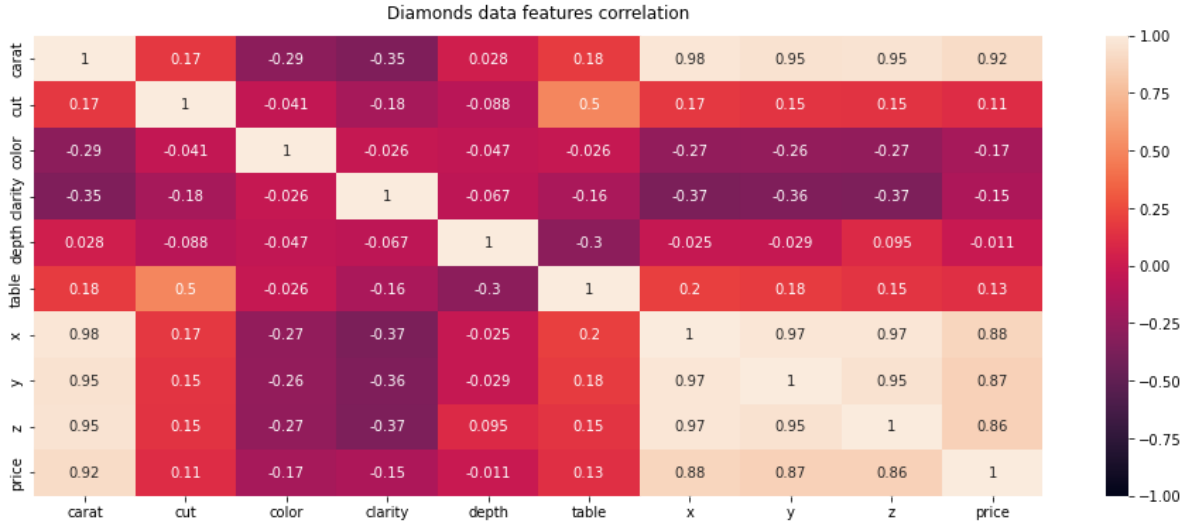
Question 2

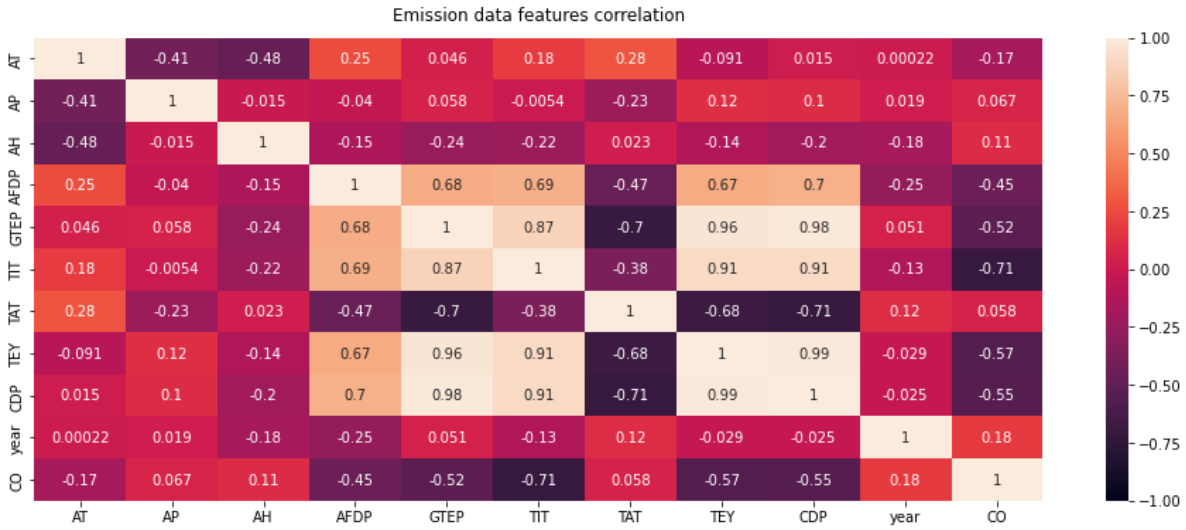
In [17]:

```
tempEmission = scaledEmission.copy()
tempEmission['CO'] = targetEmission

tempDiamonds = diamondsScaled.copy()
tempDiamonds['price'] = targetDiamonds

corrDiamonds = tempDiamonds.corr()
corrEmission = tempEmission.corr()
```

In [18]:

```
plt.figure(figsize=(16, 6))

heatmapDiamonds = sns.heatmap(corrDiamonds, vmin=-1, vmax=1, annot=True)
heatmapDiamonds.set_title('Diamonds data features correlation', fontdict={'fontsize'
plt.show()


plt.figure(figsize=(16, 6))
heatmapEmission = sns.heatmap(corrEmission, vmin=-1, vmax=1, annot=True)
heatmapEmission.set_title('Emission data features correlation', fontdict={'fontsize'
plt.show()
```





For Diamonds dataset:

Carat, x(length), y(width), z(depth) has the highest absolute correlation in (decreasing order of values).

Also the correlation is positive which means as the value of features increases, the price also increases and vice versa. This makes sense since, bigger diamonds have more price generally.

I can also see that features [x, y, z, carat] are strongly correlated which is expected as carat is a function of diamond shape.

For the emission dataset:

TIT, TEY, CDP, GTEP, AFDP has the highest absolute correlation in (decreasing order of values).

All of above mentioned sensor values have negative correlation with the CO emissions which indicate as the value of these sensor readings go up, the value of CO emission goes down and vice versa.

Maybe these sensors measure the purity of clean air and when purity decreases that means pollution (CO) is increasing.

Pearson's correlation coefficient is a measure of linear correlation between two variables which takes values in between [-1, 1]. Negative values means variables are negatively correlated and vice versa. 0 means they are not correlated at all.

$$\rho_{X,Y} = \frac{\mathbb{E}[(X-\mu_X)(Y-\mu_Y)]}{\rho_X \rho_Y}$$

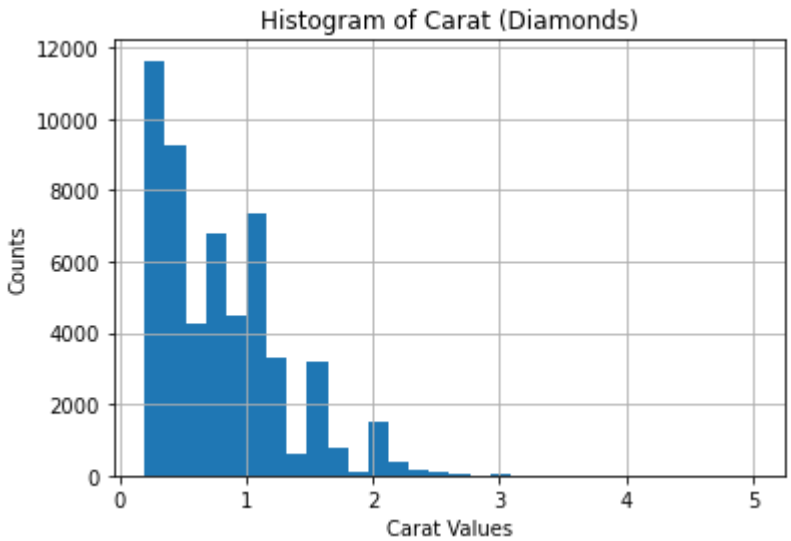
X and Y are features, and μ and ρ are respective mean and standard deviations of that feature.

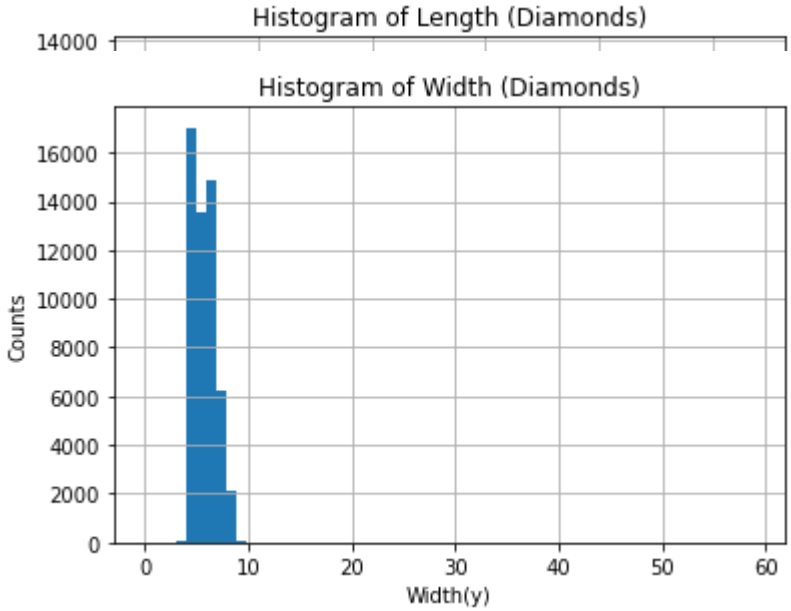
Question 3

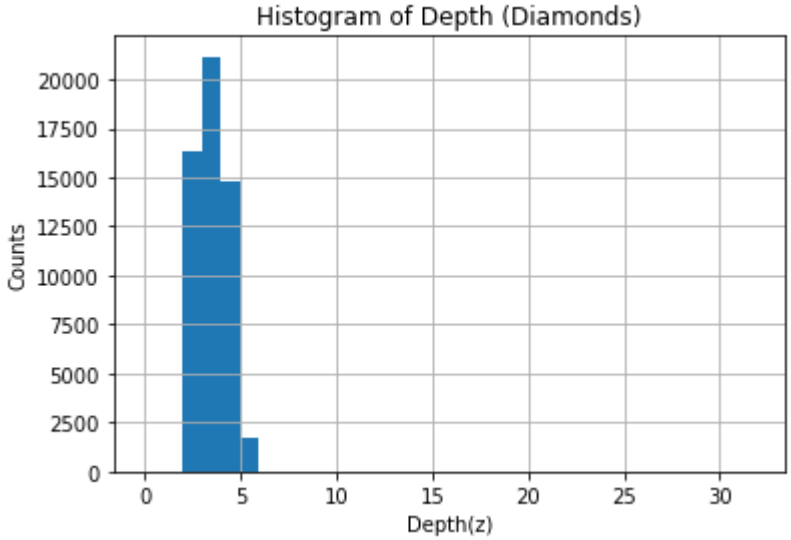
Histograms for numerical features of Diamonds dataset:

In [19]:

```
diamondsData['carat'].hist(bins = 30)
plt.title("Histogram of Carat (Diamonds)")
plt.xlabel("Carat Values")
plt.ylabel("Counts")
plt.show()


diamondsData['x'].hist(bins = 20)
plt.title("Histogram of Length (Diamonds)")
plt.xlabel("Length(x)")
plt.ylabel("Counts")
plt.show()


diamondsData['y'].hist(bins = 60)
plt.title("Histogram of Width (Diamonds)")
plt.xlabel("Width(y)")
plt.ylabel("Counts")
plt.show()


diamondsData['z'].hist(bins = 32)
plt.title("Histogram of Depth (Diamonds)")
plt.xlabel("Depth(z)")
plt.ylabel("Counts")
plt.show()
```

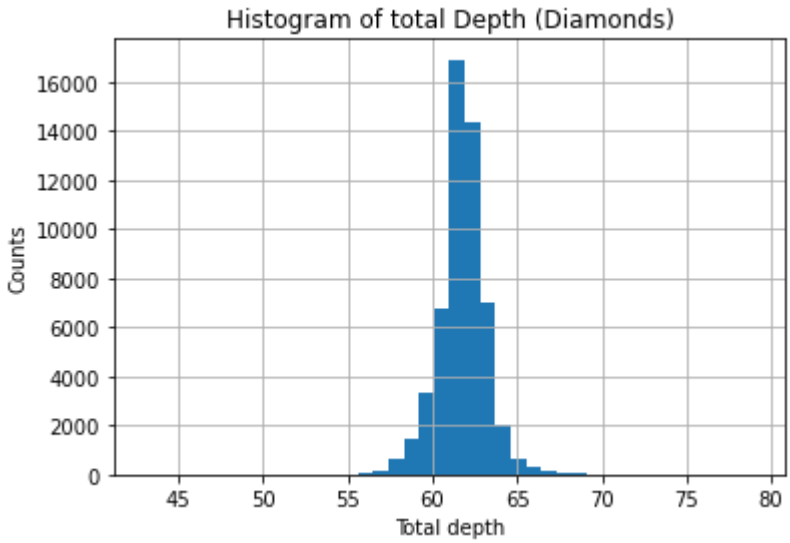


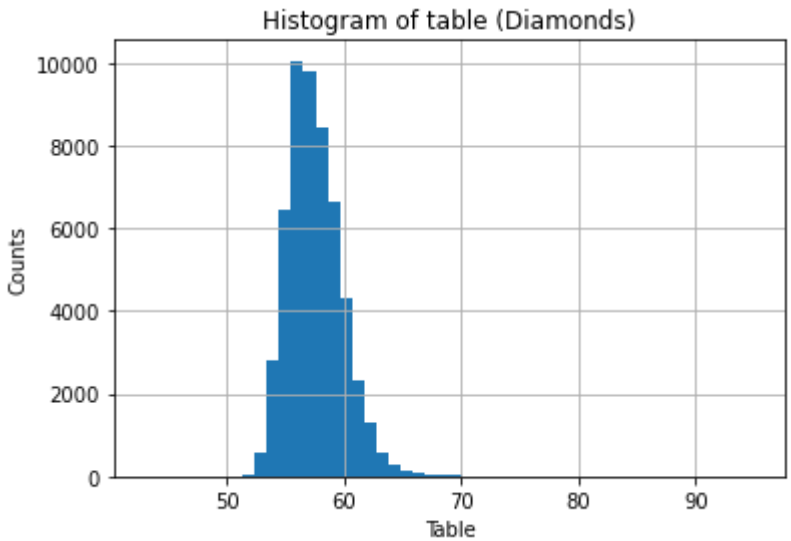

Histogram of Length (Diamonds)
14000
Histogram of Width (Diamonds)
16000
14000
12000
10000
8000
6000
4000
2000
0
Counts
0
10
20
30
40
50
60
Width(y)

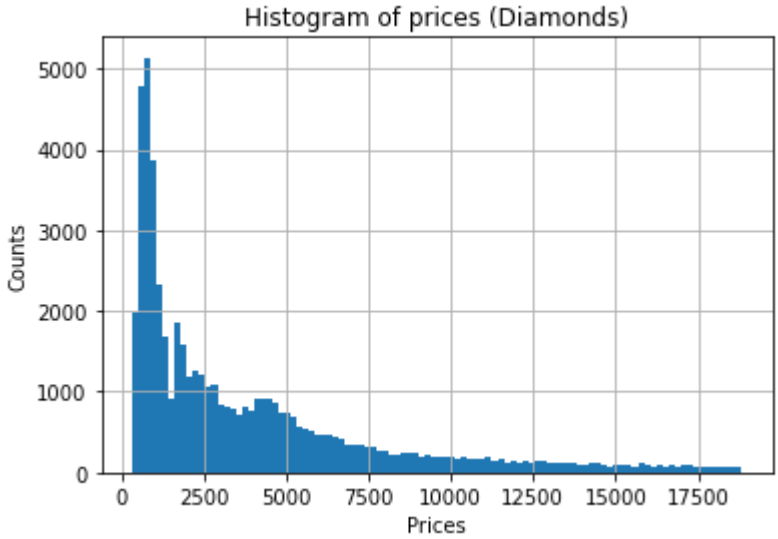

Histogram of Depth (Diamonds)
20000
17500
15000
12500
10000
7500
5000
2500
0
Counts
0
5
10
15
20
25
30
Depth(z)

In [20]:

```
diamondsData['depth'].hist(bins = 40)
plt.title("Histogram of total Depth (Diamonds)")
plt.xlabel("Total depth")
plt.ylabel("Counts")
plt.show()

diamondsData['table'].hist(bins = 50)
plt.title("Histogram of table (Diamonds)")
plt.xlabel("Table")
plt.ylabel("Counts")
plt.show()


diamondsData['price'].hist(bins = 100)
plt.title("Histogram of prices (Diamonds)")
plt.xlabel("Prices")
plt.ylabel("Counts")
plt.show()
```



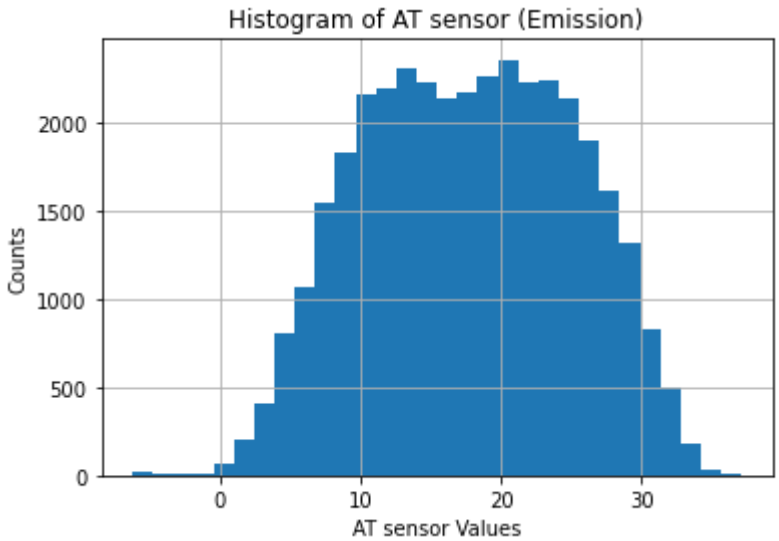


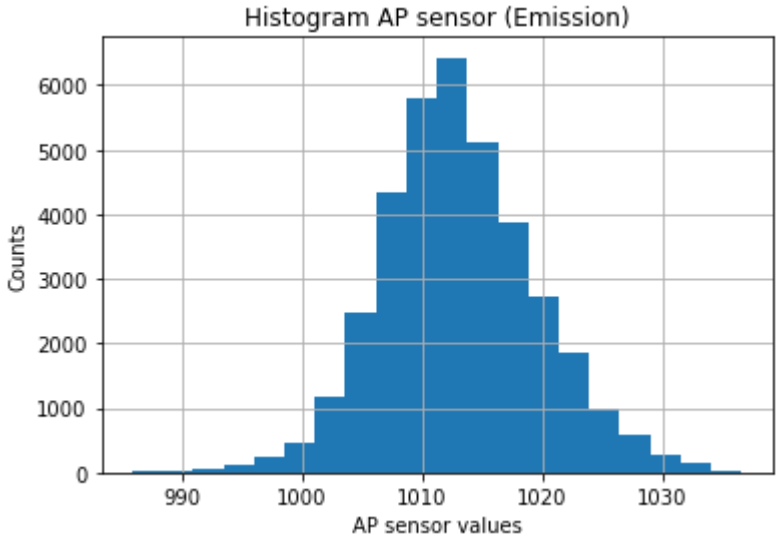


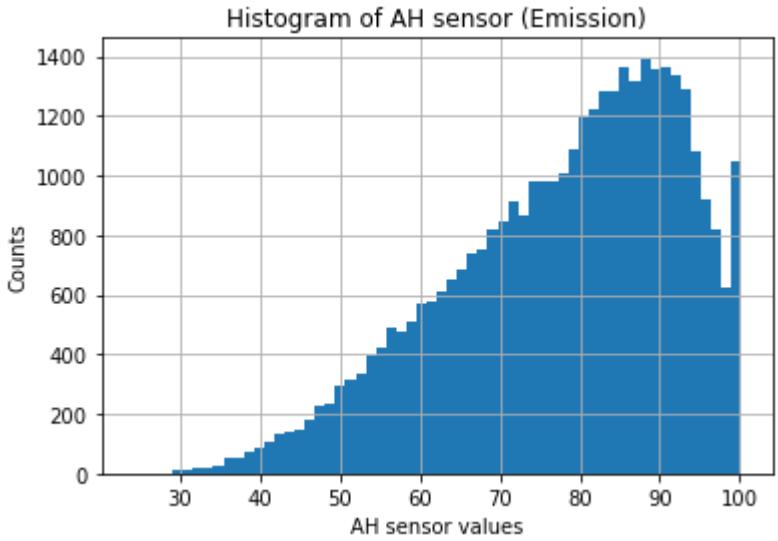
Histograms for numerical features of Emission dataset:

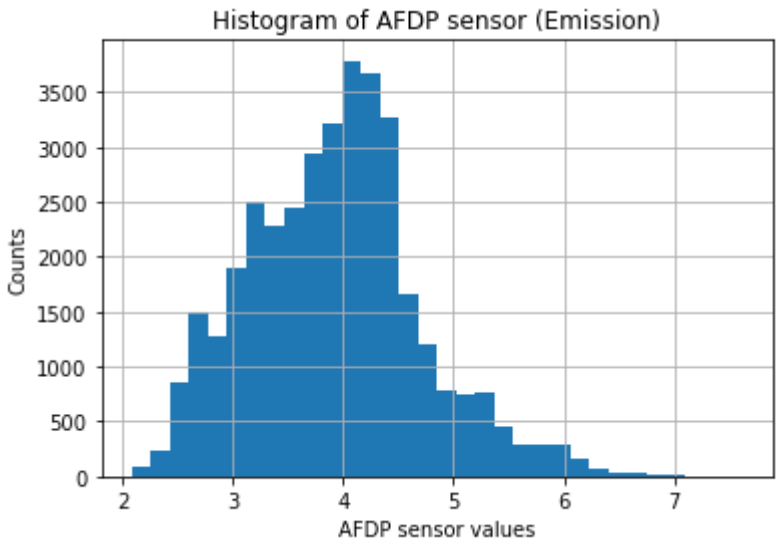
In [21]:

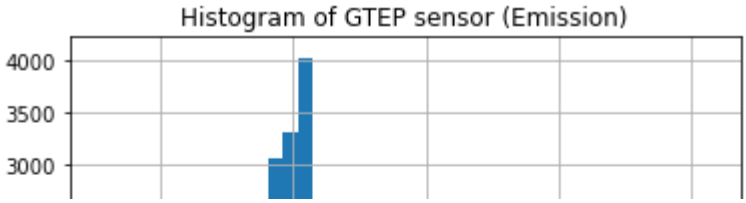
```
emissionData['AT'].hist(bins = 30)
plt.title("Histogram of AT sensor (Emission)")
plt.xlabel("AT sensor Values")
plt.ylabel("Counts")
plt.show()


emissionData['AP'].hist(bins = 20)
plt.title("Histogram AP sensor (Emission)")
plt.xlabel("AP sensor values")
plt.ylabel("Counts")
plt.show()


emissionData['AH'].hist(bins = 60)
plt.title("Histogram of AH sensor (Emission)")
plt.xlabel("AH sensor values")
plt.ylabel("Counts")
plt.show()


emissionData['AFDP'].hist(bins = 32)
plt.title("Histogram of AFDP sensor (Emission)")
plt.xlabel("AFDP sensor values")
plt.ylabel("Counts")
plt.show()

emissionData['GTEP'].hist(bins = 40)
plt.title("Histogram of GTEP sensor (Emission)")
plt.xlabel("GTEP sensor values")
plt.ylabel("Counts")
plt.show()
```




Histogram AP sensor (Emission)
Counts
AP sensor values
0
1000
2000
3000
4000
5000
6000
990
1000
1010
1020
1030

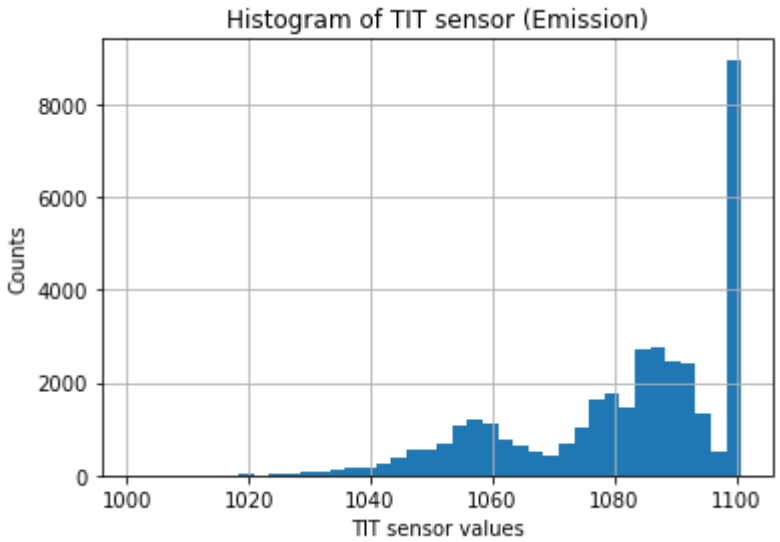

Histogram of AH sensor (Emission)
Counts
AH sensor values
0
200
400
600
800
1000
1200
1400
30
40
50
60
70
80
90
100

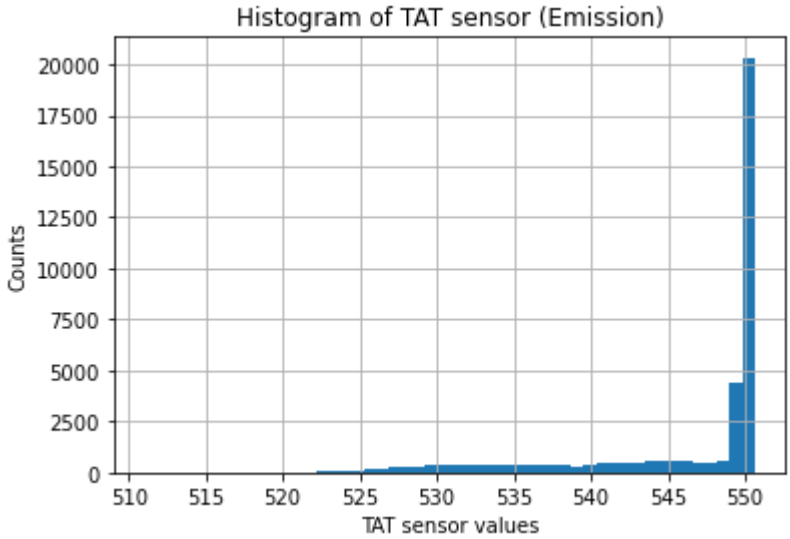

Histogram of AFDP sensor (Emission)
Counts
AFDP sensor values
0
500
1000
1500
2000
2500
3000
3500
2
3
4
5
6
7

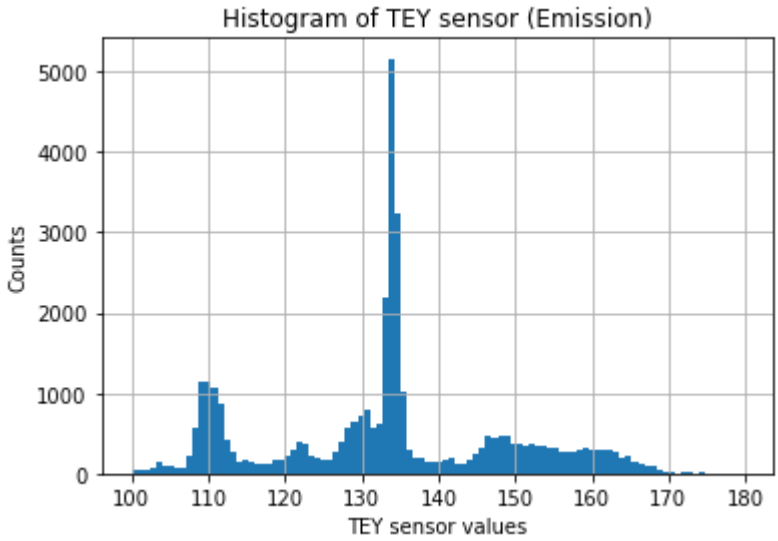

Histogram of GTEP sensor (Emission)
4000
3500
3000

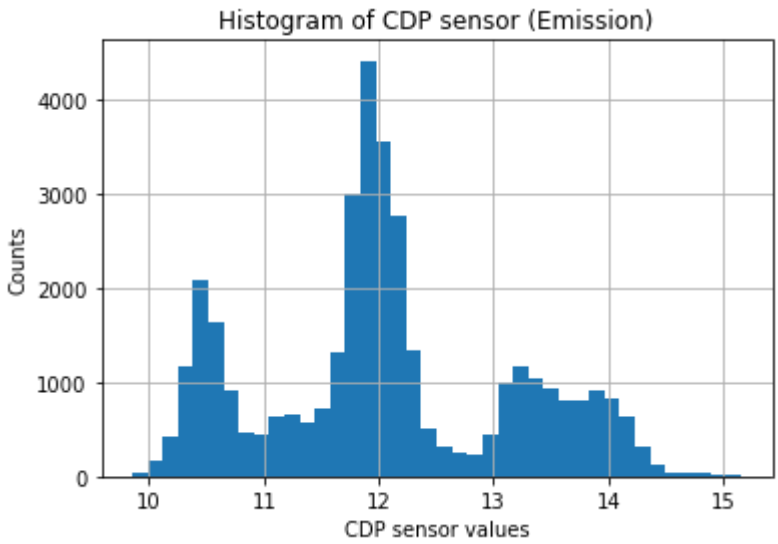
In [22]:

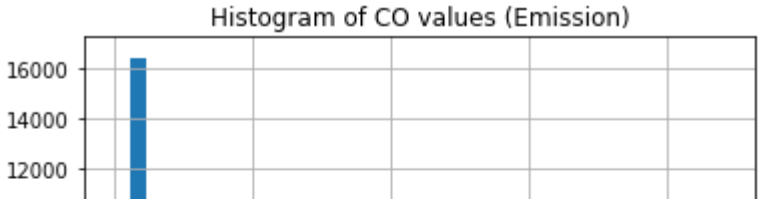
```
emissionData['TIT'].hist(bins = 40)
plt.title("Histogram of TIT sensor (Emission)")
plt.xlabel("TIT sensor values")
plt.ylabel("Counts")
plt.show()

emissionData['TAT'].hist(bins = 50)
plt.title("Histogram of TAT sensor (Emission)")
plt.xlabel("TAT sensor values")
plt.ylabel("Counts")
plt.show()


emissionData['TEY'].hist(bins = 100)
plt.title("Histogram of TEY sensor (Emission)")
plt.xlabel("TEY sensor values")
plt.ylabel("Counts")
plt.show()

emissionData['CDP'].hist(bins = 40)
plt.title("Histogram of CDP sensor (Emission)")
plt.xlabel("CDP sensor values")
plt.ylabel("Counts")
plt.show()

emissionData['CO'].hist(bins = 40)
plt.title("Histogram of CO values (Emission)")
plt.xlabel("CO values")
plt.ylabel("Counts")
plt.show()
```




Histogram of TAT sensor (Emission)
Counts
TAT sensor values
0
2500
5000
7500
10000
12500
15000
17500
20000
510
515
520
525
530
535
540
545
550


Histogram of TEY sensor (Emission)
Counts
TEY sensor values
0
1000
2000
3000
4000
5000
100
110
120
130
140
150
160
170
180


Histogram of CDP sensor (Emission)
Counts
CDP sensor values
0
1000
2000
3000
4000
10
11
12
13
14
15



Answer 3

The histograms of numerical features for both datasets are plotted above.

We can see from above histograms that the **prices and CO emissions** are positively skewed. It implies that lower prices are more dominant in the dataset which can lead to a biased model and thus predicting lower values as the models will see more of this type of data.

One way to deal with this is to trim the removing a few of the samples belonging to the left tail. Or properly shuffling the data using Cross validation so that data is properly distributed.

On the features which observe skewness we can use different transformations to make them less skewed.Eg.

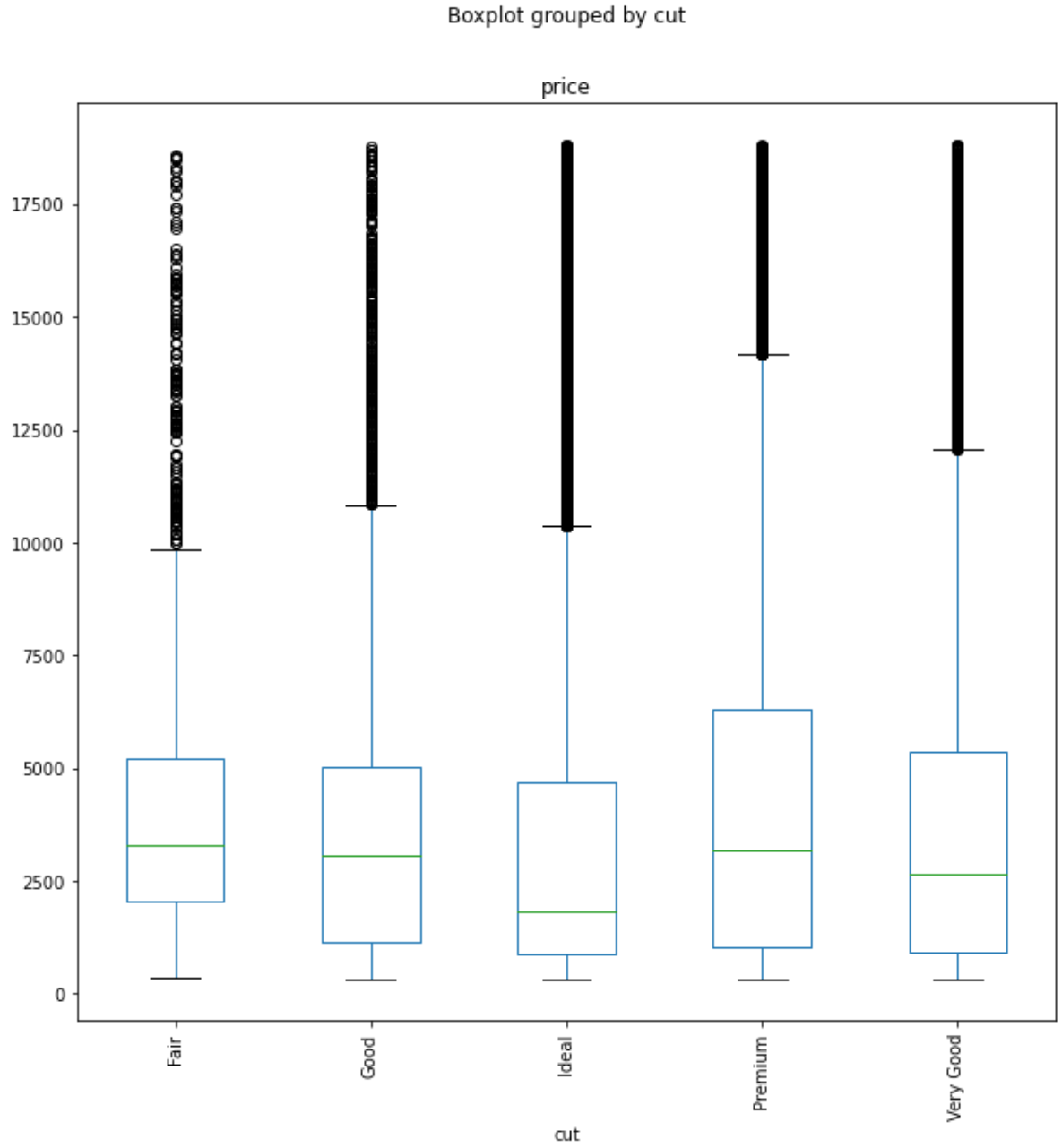
- Log transformation
- Square Root transformation
- Reciprocal transformation

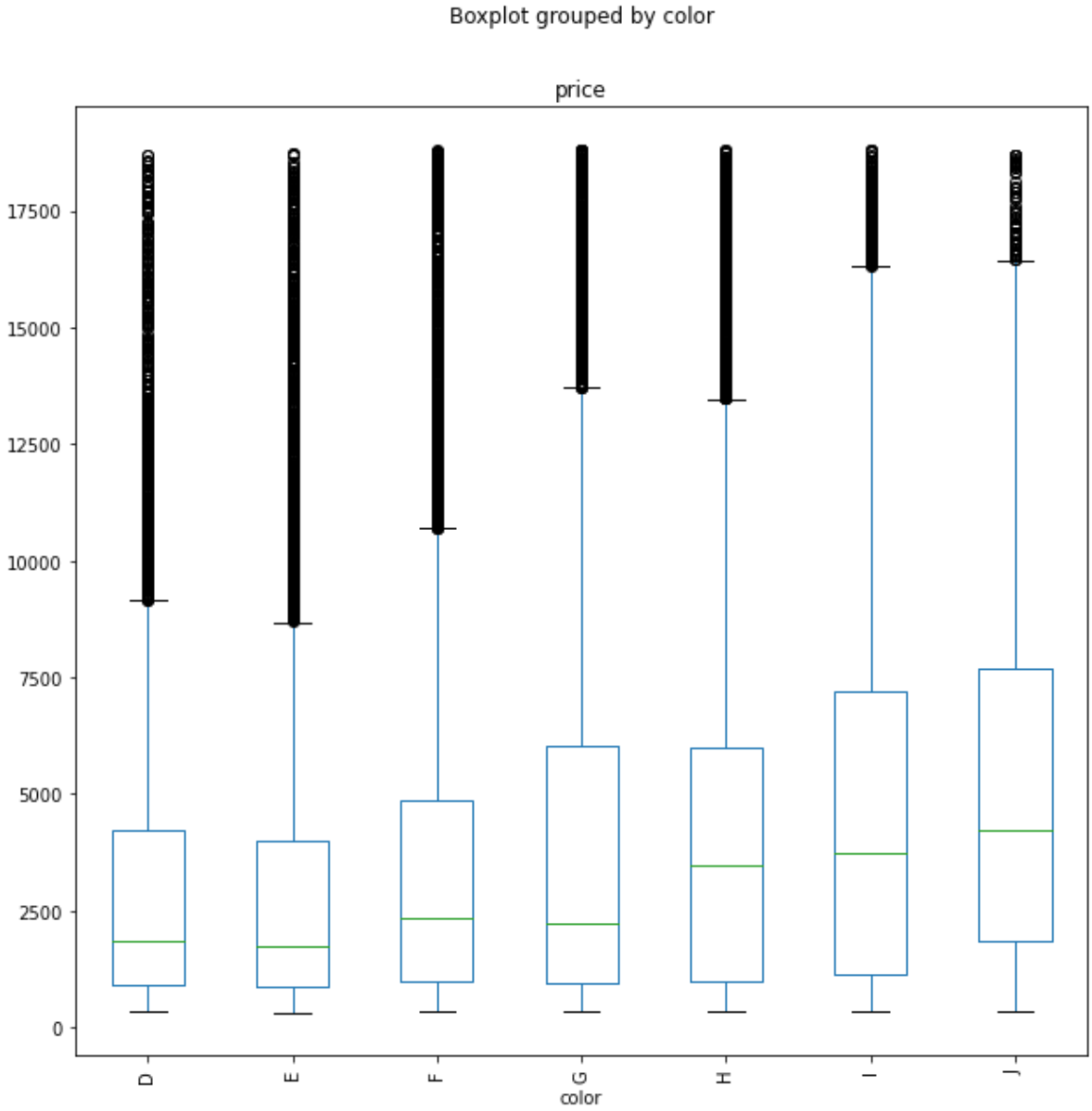
Data should be transformed first before standardisation as standardisation makes some values negative and thus, log, quare root transformation will fail.

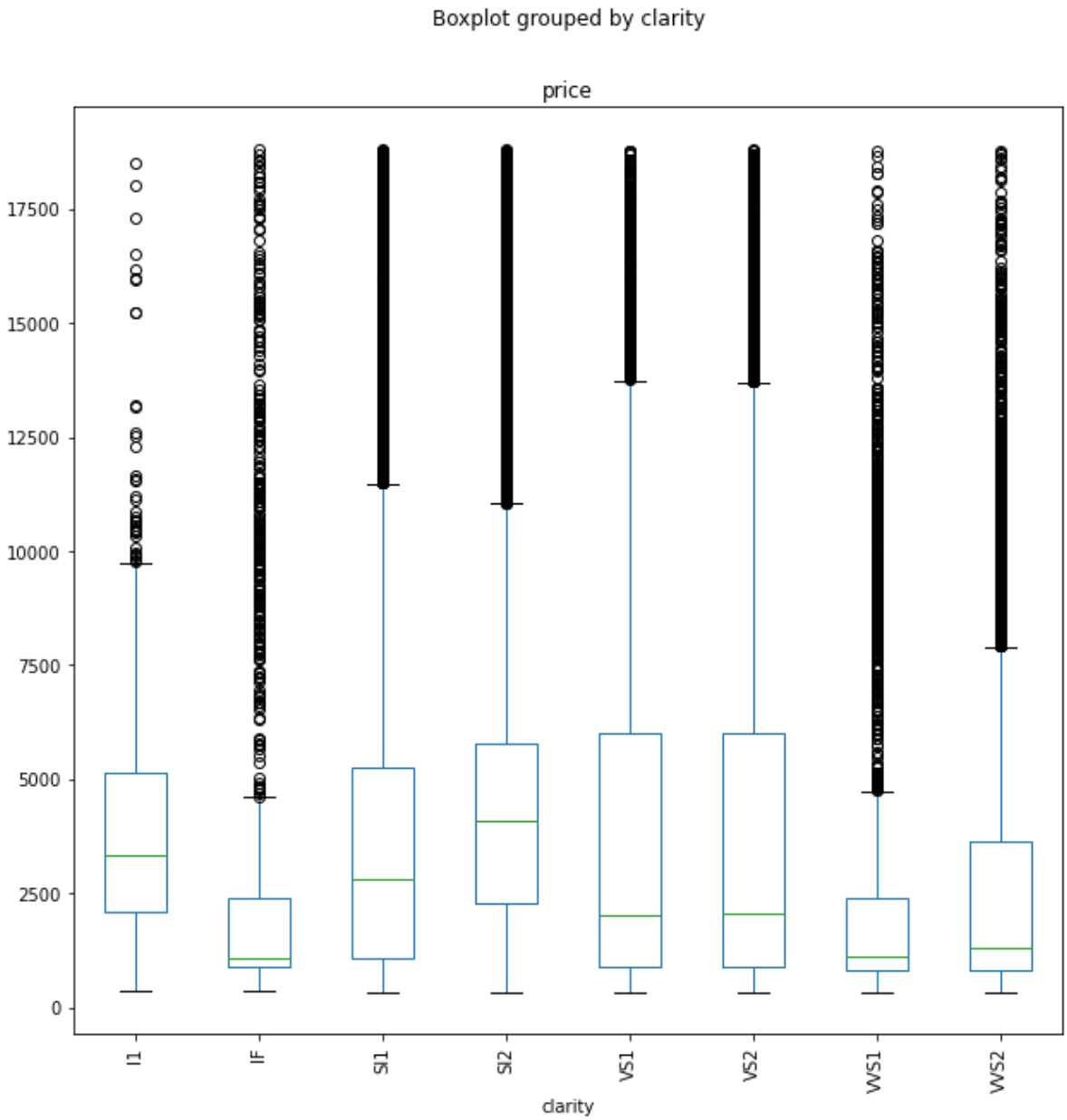
Box plots of Categorical features : Diamonds Dataset

In [23]:

```
cat_diamond_features = ['cut', 'color', 'clarity']
for feature in cat_diamond_features:
    plot = diamondsData.boxplot(column = 'price',figsize=(10,10), by = feature, gric
    plot.tick_params(axis='x', rotation=90)
```



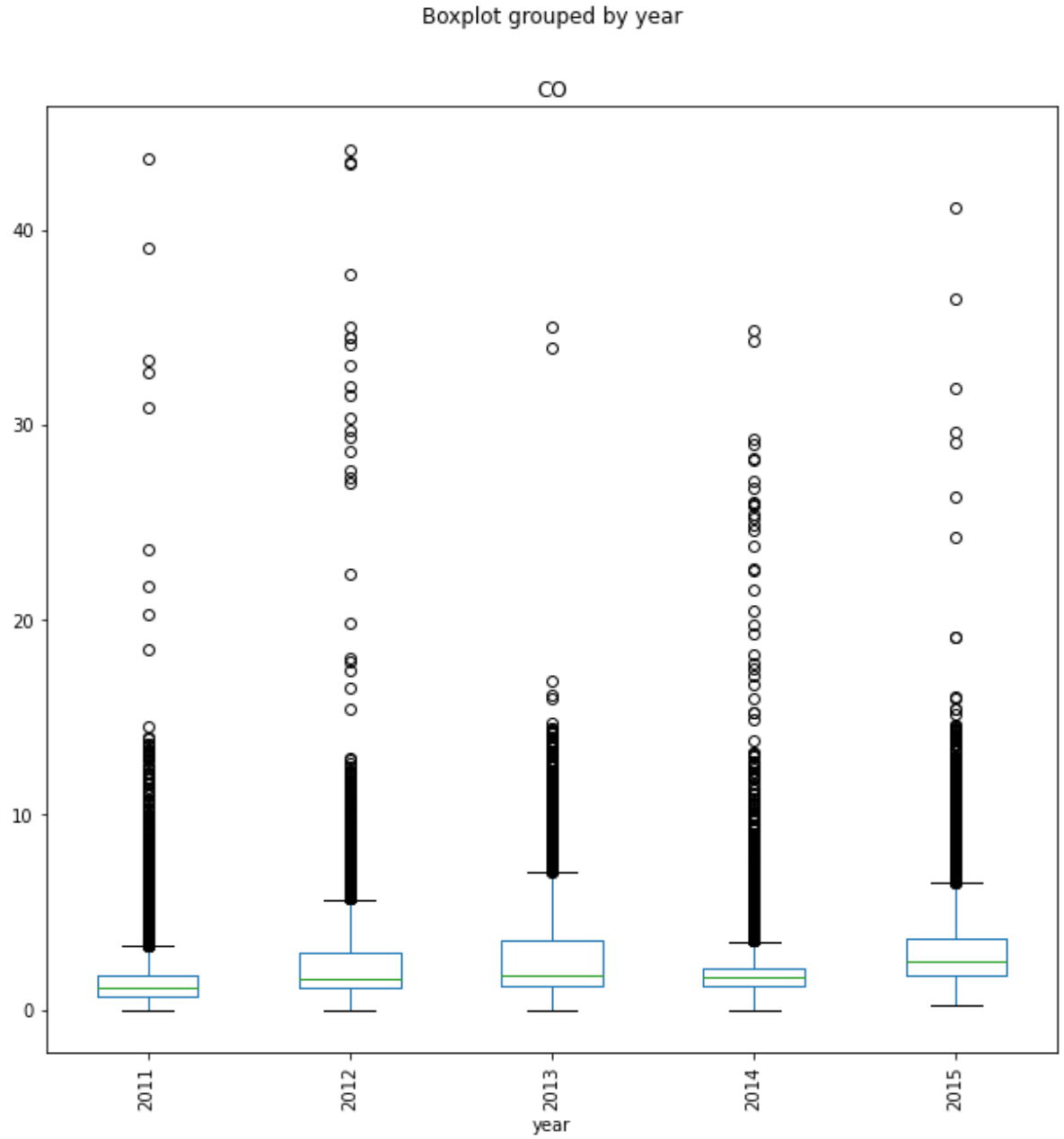

Boxplot grouped by color
price
17500
15000
12500
10000
7500
5000
2500
0
D
E
F
G
H
I
J
color


Boxplot grouped by clarity
price
17500
15000
12500
10000
7500
5000
2500
0
I1
IF
SI1
SI2
VS1
VS2
VVS1
VVS2
clarity

Box plots of Categorical features : Emission Dataset

In [24]:

```
cat_emission_features = ['year']
for feature in cat_emission_features:
    plot = emissionData.boxplot(column = 'CO',figsize=(10,10), by = feature, grid =
    plot.tick_params(axis='x', rotation=90)
```



Answer 4

Box plots of categorical features vs target variables are shown above for both the datasets.

For **Diamonds dataset**
Categorical features: color, cut, clarity
Target: Price

For **CO Emissions dataset**
Categorical features: year
Target: Emission ('CO')

Cut vs Price: I can see that premium diamonds generally have higher prices and those have really high outliers. Also all the categories in this features are right (positively skewed). The means are cut on different levels and thus is a helful feature for the prediction.

Color vs Price: Strangely the worst color (J) is seen to usually have higher prices. Color is a good feature to use given the means are distinguishable for different classes. All of the classes are positively skewed and large no. outliers.

Clarity vs price: All of the classes in clarity are very heavily positvely skewed. That is large number of outliers. Also it's a good feature to use given means are separated quite well. IF and VVS1 has the most variance in terms of prices.

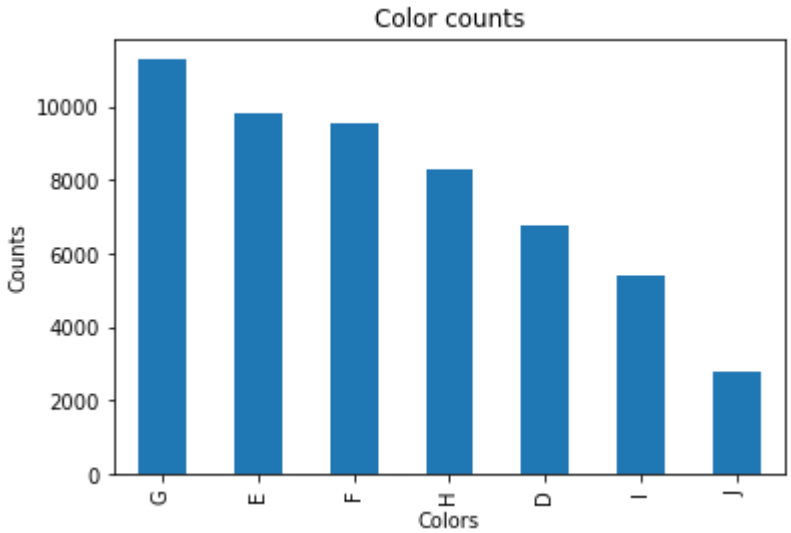
Emission dataset:

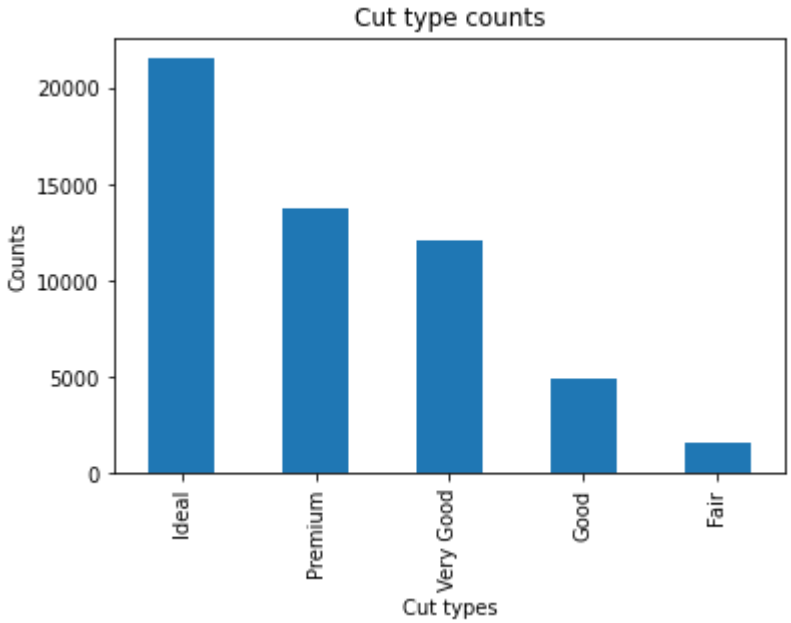
The means are very closely distributed per year which indicate that its not a very good feature to use for emission prediction. There are outliers but not very vaguely distributed.

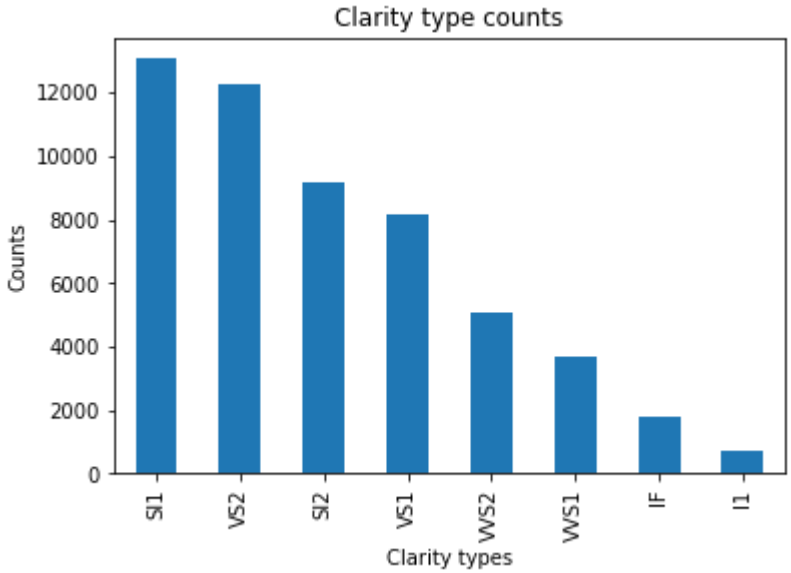
Counts by colour, cut and clarity for diamond dataset

In [25]:

```
diamondsData['color'].value_counts().plot(kind = 'bar')
plt.title('Color counts')
plt.xlabel('Colors')
plt.ylabel('Counts')
plt.show()

diamondsData['cut'].value_counts().plot(kind = 'bar')
plt.title('Cut type counts')
plt.xlabel('Cut types')
plt.ylabel('Counts')
plt.show()


diamondsData['clarity'].value_counts().plot(kind = 'bar')
plt.title('Clarity type counts')
plt.xlabel('Clarity types')
plt.ylabel('Counts')
plt.show()
```







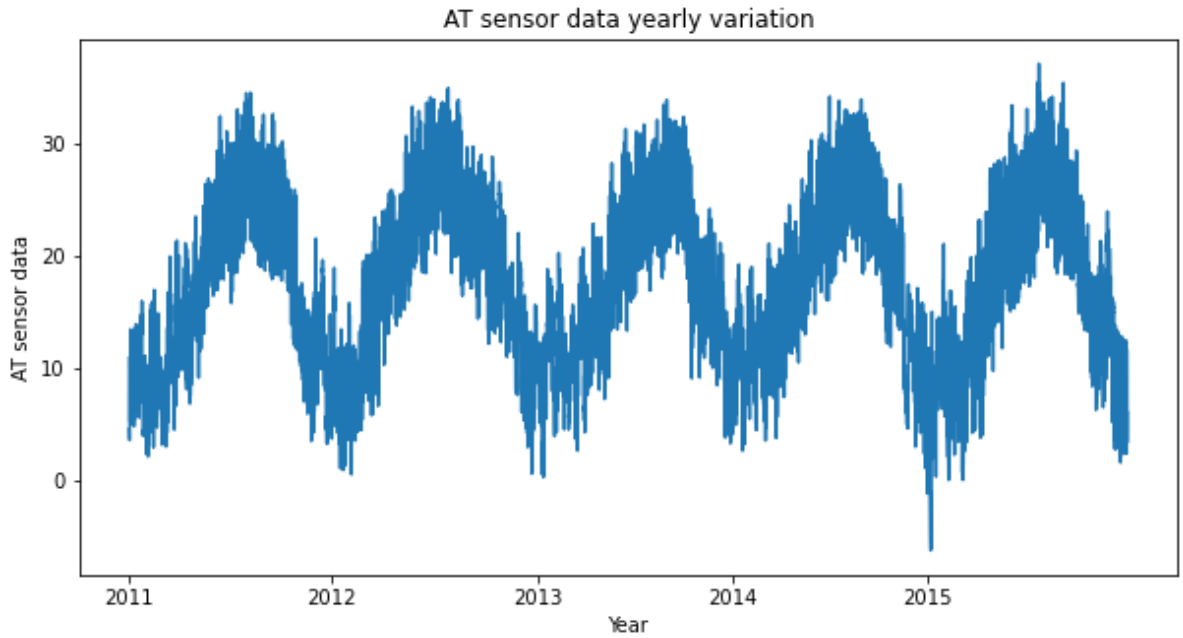
Answer 5

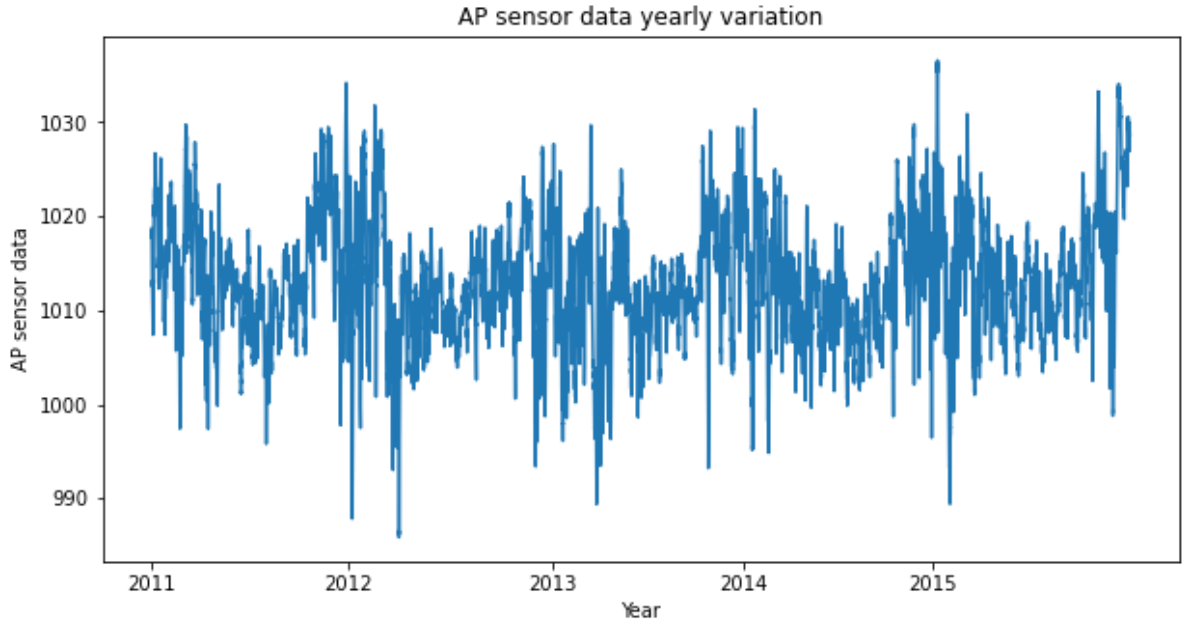
The plots of counts with respect to colour, cut and clarity are plotted above for diamonds dataset.It shows that there is imbalance of counts in these features values which might lead to not so good predictions for some values have lower counts in each feature.

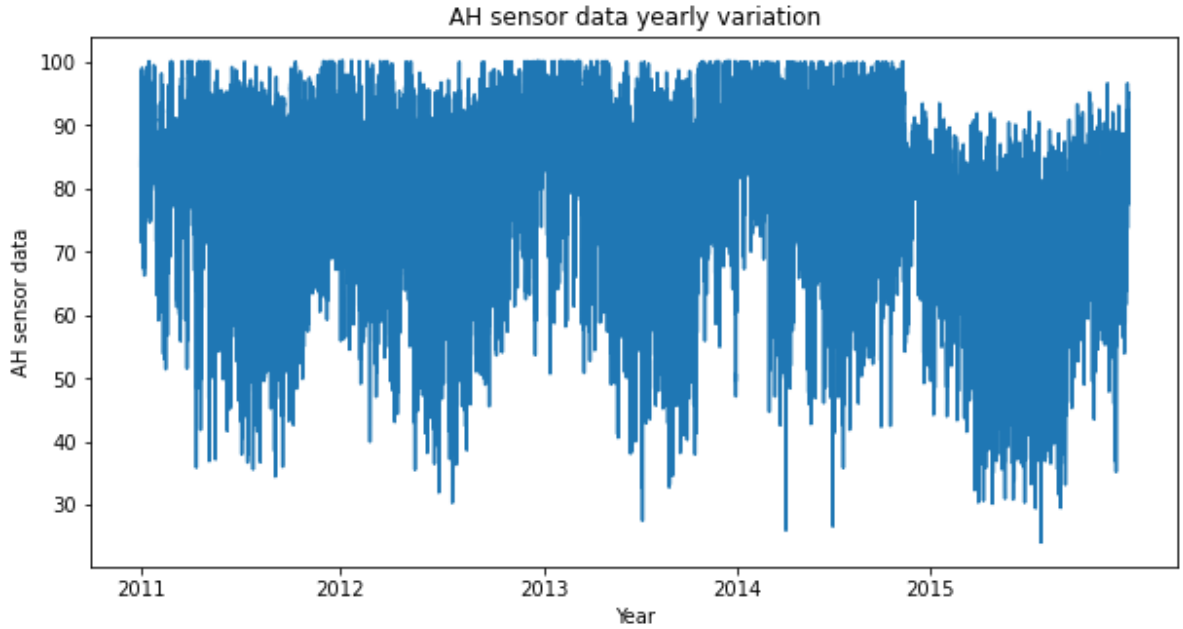
Yearly trends for each feature in Emission dataset

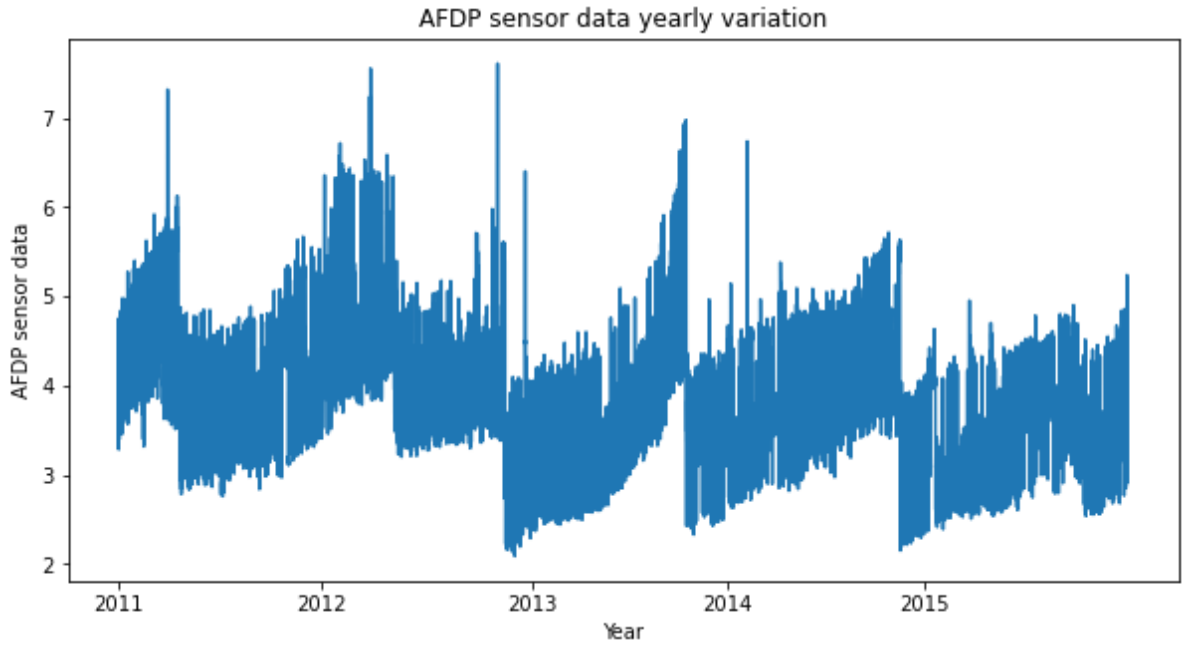
In [26]:

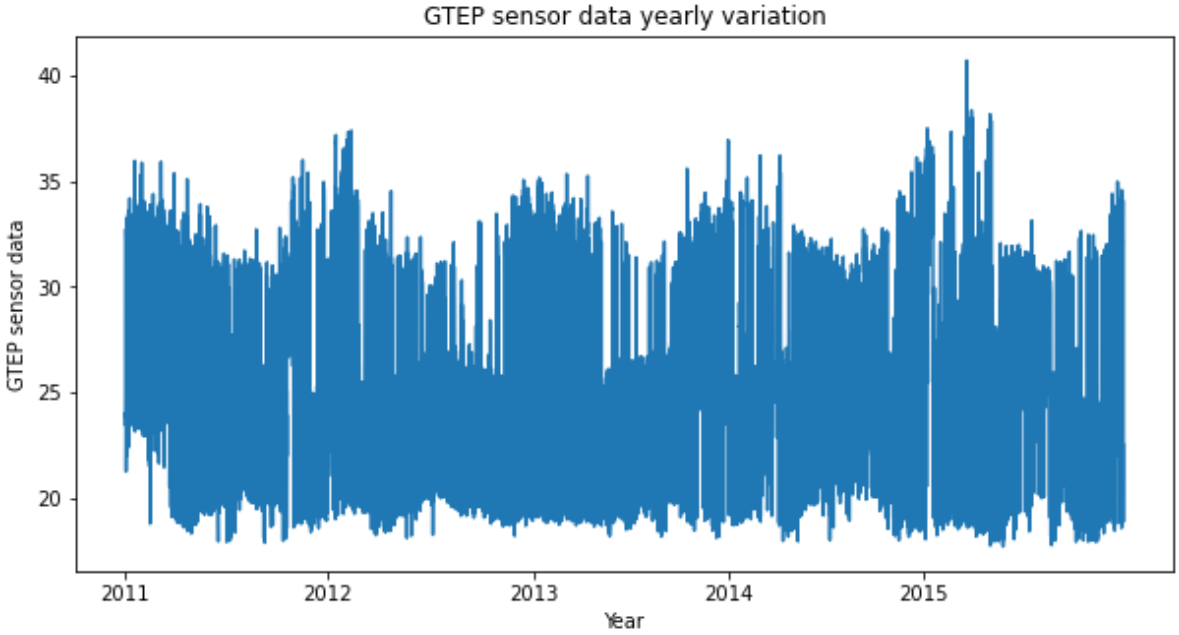
```
emissionIndices = emissionData.index.to_numpy()

x = emissionIndices

a = len(gt2011)
b = len(gt2012)
c = len(gt2013)
d = len(gt2014)
e = len(gt2015)

yearIndices = [0, a, a + b, a + b + c, a + b + c + d]


emissionFeats = ['AT', 'AP', 'AH', 'AFDP', 'GTEP', 'TIT', 'TAT', 'TEY', 'CDP', 'CO']

for feat in emissionFeats:
    f = plt.figure()
    f.set_figwidth(10)
    f.set_figheight(5)

    plt.plot(x, emissionData[feat])
    plt.xticks(yearIndices, ["2011", "2012", "2013", "2014", "2015"])
    plt.title("{} sensor data yearly variation".format(feat))
    plt.xlabel("Year")
    plt.ylabel("{} sensor data".format(feat))
    plt.show()
```

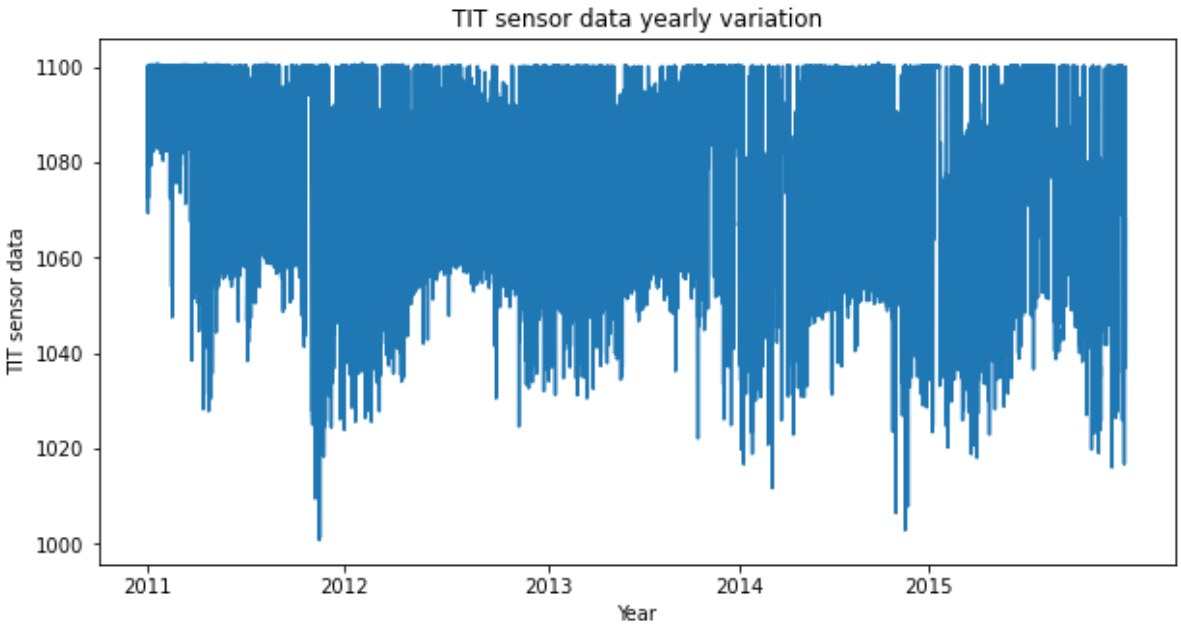


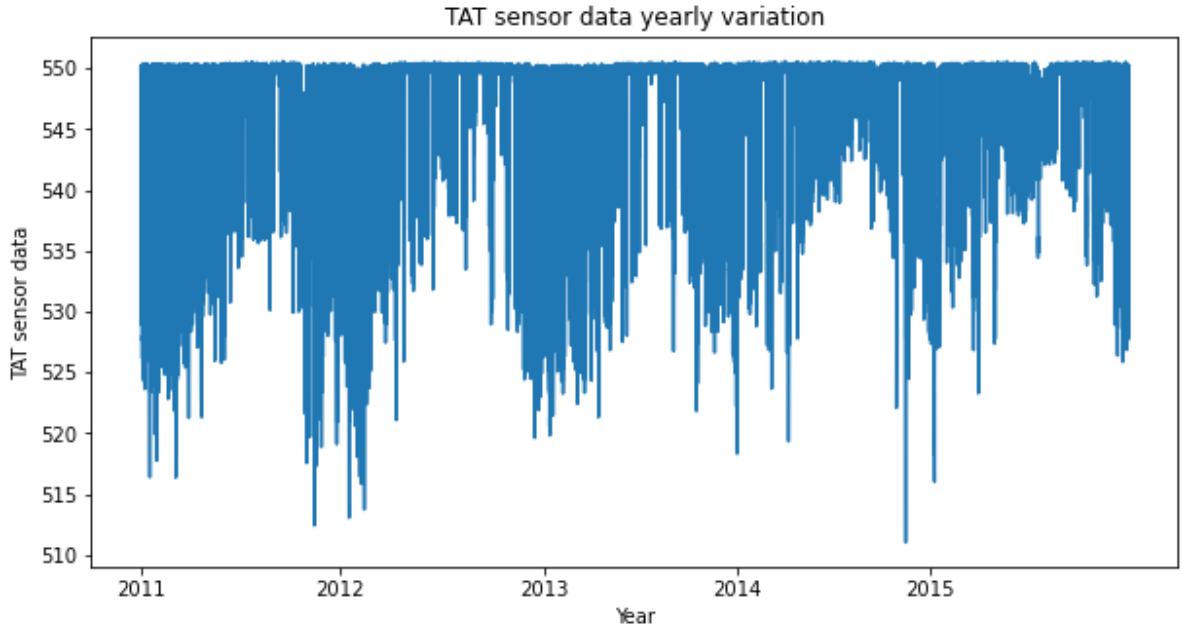

AP sensor data yearly variation
AP sensor data
1030
1020
1010
1000
990
2011
2012
2013
2014
2015
Year

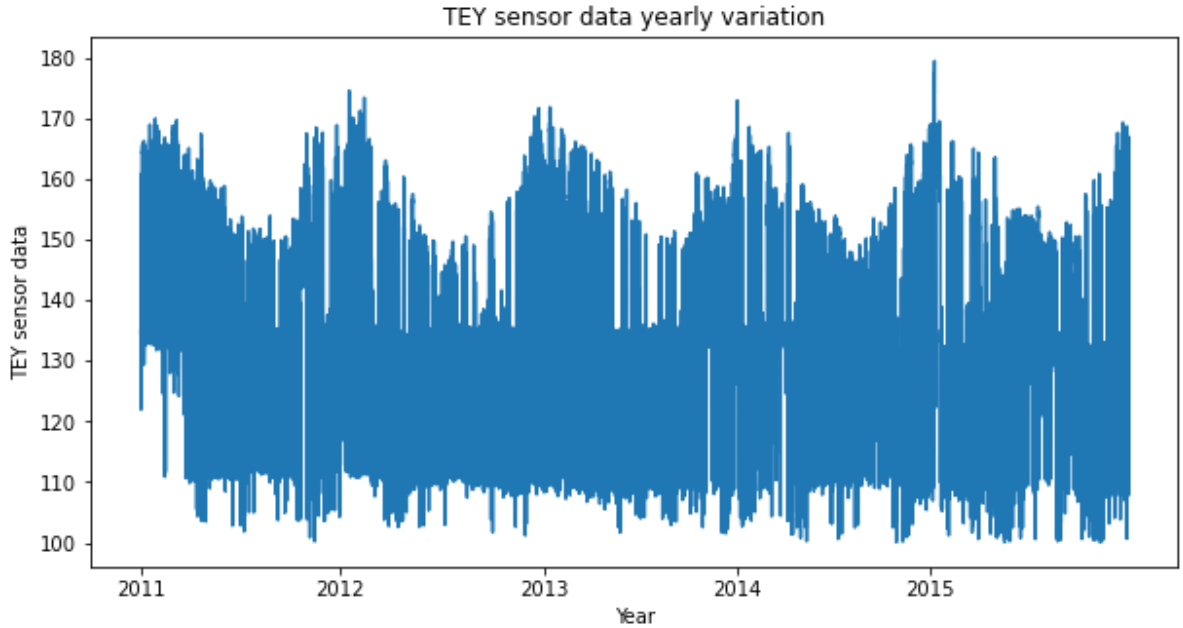

AH sensor data yearly variation
AH sensor data
100
90
80
70
60
50
40
30
2011
2012
2013
2014
2015
Year

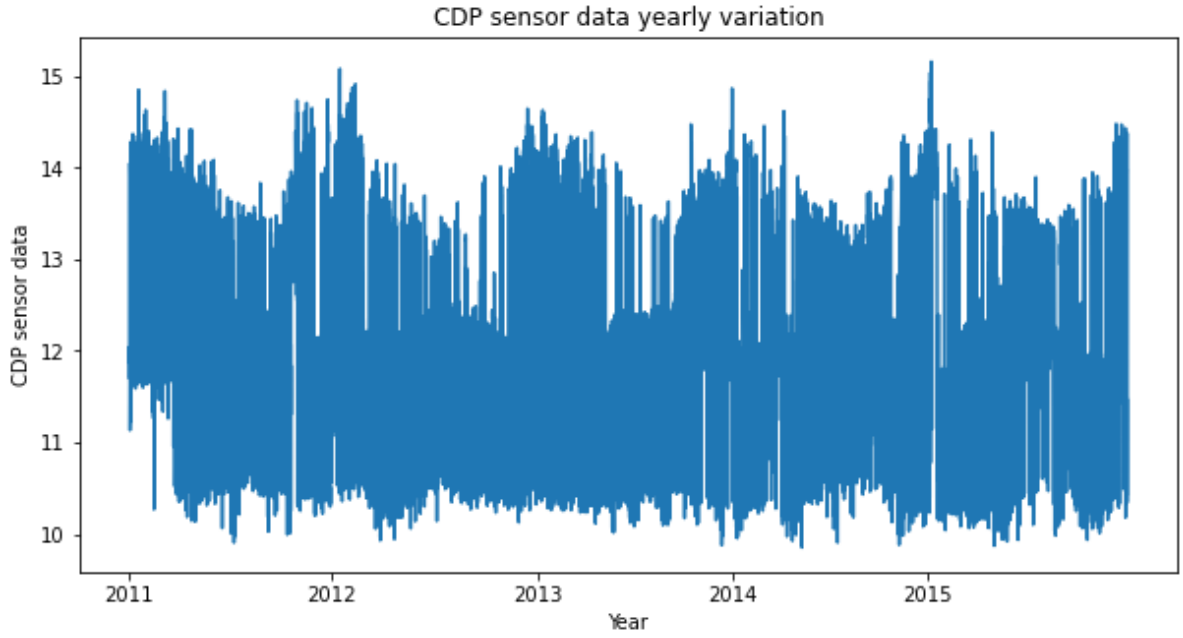

AFDP sensor data yearly variation
AFDP sensor data
7
6
5
4
3
2
2011
2012
2013
2014
2015
Year

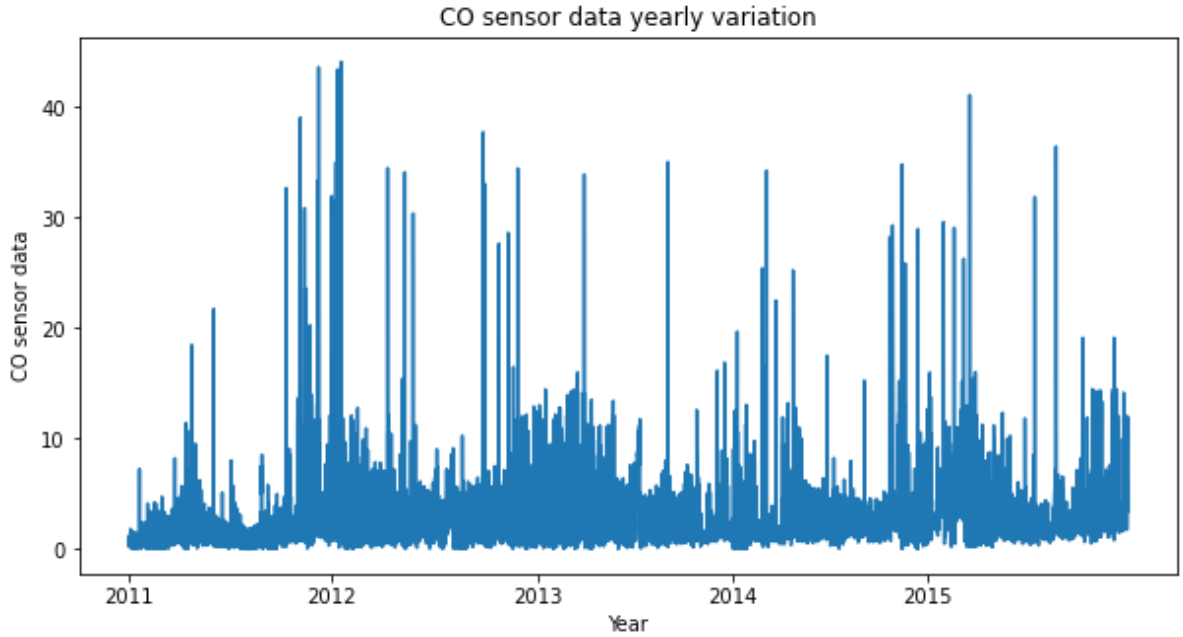

GTEP sensor data yearly variation
GTEP sensor data
40
35
30
25
20
2011
2012
2013
2014
2015
Year


TIT sensor data yearly variation
TIT sensor data
1100
1080
1060
1040
1020
1000
2011
2012
2013
2014
2015
Year


TAT sensor data yearly variation
TAT sensor data
550
545
540
535
530
525
520
515
510
2011
2012
2013
2014
2015
Year


TEY sensor data yearly variation
TEY sensor data
180
170
160
150
140
130
120
110
100
2011
2012
2013
2014
2015
Year





Answer 6

Yearly trends for each sensor data feature is plotted above for CO emission dataset.

For AT sensor, the yearly trend is same. The sensor readings increase during a particular season or (the first half of each year) and then again they go down. Comparing the peaks, 2015 has the biggest peak which means this sensor peak reading is going up in recent years. Readings range between (2 - 40)

AP sensor has the same pattern as AT sensor but just opposite. **Which makes sense since AT and AP are negatively correlated as seen from the correlation matrix.**
Sensor reading keep decreasing during first half of the year and then it increases. Readings range between (980 - 1050), The depressions are going slightly up each year.

AH sensor data is slowly decreasing over time. The overall trend is going down for this sensor.

For AFDP sensor the range is kept almost constant for the variations each year but the overall trend is taht its going doing.

GTEP seems to have quite constant trend over the years but the peaks seen at start of each year are having increasing trend.

TIT sensor data overall is achieving greater dpression levels each year and as seen **from correlations plot this has high negative correlation from CO thus it means CO levels increase each year**

TAT, TAY and CDP sensor data are having same trend over the years.

Feature Selection

- Diamonds Dataset

In [27]:

```
discrete_mask = [False, True, True, True, False, False, False, False, False]
mi_diamonds = mutual_info_regression(diamondsScaled, targetDiamonds,
                      discrete_features = discrete_mask, copy = True, random_state
```

In [28]:

```
m_info_diamonds = pd.Series(mi_diamonds)
m_info_diamonds.index = diamondsScaled.columns
m_info_diamonds.sort_values(ascending=False)
```

Out[28]:

```
carat      1.652888
y          1.421612
x          1.412996
z          1.361523
clarity    0.217263
color      0.136824
cut        0.056896
table      0.035091
depth      0.032032
dtype: float64
```

In [29]:

```
f_diamonds, p_diamonds = f_regression(diamondsScaled, targetDiamonds)
```

In [30]:

```
f_diamonds
```

Out[30]:

```
array([3.04051487e+05, 6.90464192e+02, 1.65440124e+03, 1.18800706e+03,
       6.11586346e+00, 8.86119363e+02, 1.93741523e+05, 1.60915662e+05,
       1.54923267e+05])
```

- Emission dataset

In [31]:

```
discrete_mask = [False, False, False, False, False, False, False, False, False, True
mi_emission = mutual_info_regression(scaledEmission, targetEmission,
                        discrete_features = discrete_mask, copy = True, random_state
```

In [32]:

```
m_info_emission = pd.Series(mi_emission)
m_info_emission.index = scaledEmission.columns
m_info_emission.sort_values(ascending=False)
```

Out[32]:

```
TIT     0.538170
TEY     0.495486
CDP     0.474436
GTEP    0.445490
AFDP    0.278731
TAT     0.159627
year    0.123832
AT      0.104885
AP      0.042072
AH      0.025524
dtype: float64
```

In [33]:

```
f_emission, p_emission = f_regression(scaledEmission, targetEmission)
```

In [34]:

```
f_emission
```

Out[34]:

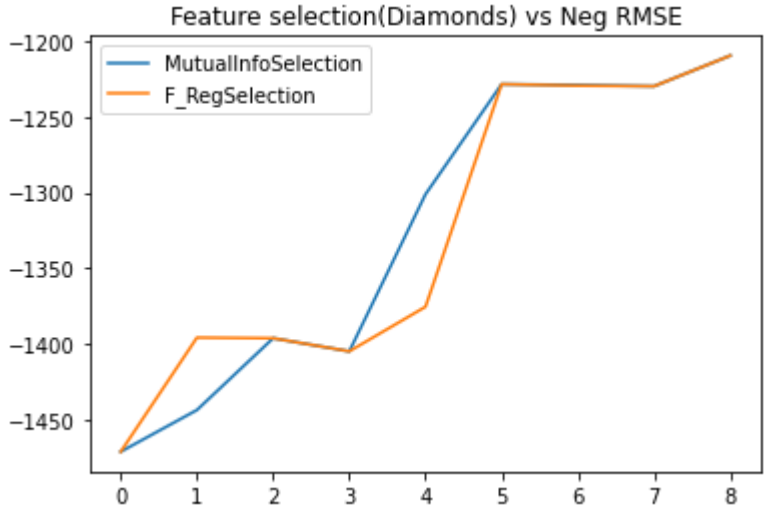
```
array([ 1151.22090472,   165.87752857,   422.08013102,  9245.08377364,
       13534.97054407, 36558.68834551,   125.50084238, 17660.02276429,
       16015.41677409,  1208.14433729])
```

In [35]:

```
train_diamonds = diamondsScaled
y_diamonds = targetDiamonds
diamond_cv_MIR = []
diamond_cv_FR = []

for i in range(1, diamondsScaled.shape[1] + 1):
    dataSelected = SelectKBest(score_func=mutual_info_regression, k=i).fit_transform
    scores = cross_validate(LinearRegression(), dataSelected, y_diamonds, scoring=['
    diamond_cv_MIR.append(scores['test_neg_root_mean_squared_error'].mean())

    dataSelected = SelectKBest(score_func=f_regression, k=i).fit_transform(train_dia
    scores = cross_validate(LinearRegression(), dataSelected, y_diamonds, scoring=['
    diamond_cv_FR.append(scores['test_neg_root_mean_squared_error'].mean())
```

In [36]:

```
plt.plot(diamond_cv_MIR,label='MutualInfoSelection')
plt.plot(diamond_cv_FR,label='F_RegSelection')
plt.title("Feature selection(Diamonds) vs Neg RMSE")
plt.legend()

best_MIR_diamonds = np.argmax(diamond_cv_MIR)
best_FR_diamonds = np.argmax(diamond_cv_FR)
```



In [37]:

```
best_MIR_diamonds, best_FR_diamonds
```

Out[37]:

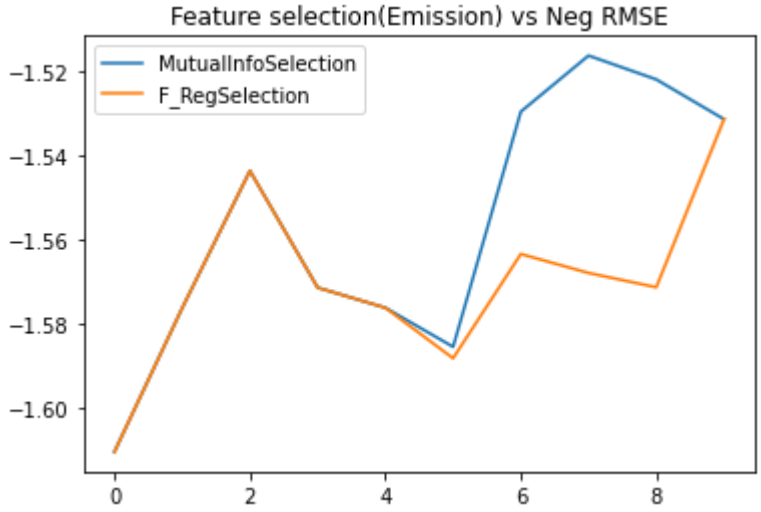
```
(8, 8)
```

In [38]:

```
train_emission = scaledEmission
y_emission = targetEmission

emission_cv_MIR = []
emission_cv_FR = []

for i in range(1, scaledEmission.shape[1] + 1):
    dataSelected = SelectKBest(score_func=mutual_info_regression, k=i).fit_transform
    scores = cross_validate(LinearRegression(), dataSelected, y_emission, scoring=['
    emission_cv_MIR.append(scores['test_neg_root_mean_squared_error'].mean())

    dataSelected = SelectKBest(score_func=f_regression, k=i).fit_transform(train_emi
    scores = cross_validate(LinearRegression(), dataSelected, y_emission, scoring=['
    emission_cv_FR.append(scores['test_neg_root_mean_squared_error'].mean())
```

In [39]:

```
plt.plot(emission_cv_MIR,label='MutualInfoSelection')
plt.plot(emission_cv_FR,label='F_RegSelection')
plt.title("Feature selection(Emission) vs Neg RMSE")
plt.legend()

best_MIR_emission = np.argmax(emission_cv_MIR)
best_FR_emission = np.argmax(emission_cv_FR)
```



In [40]:

```
best_MIR_emission, best_FR_emission
```

Out[40]:

```
(7, 9)
```

In [41]:

```
# Choosing all 8 best features from the diamonds dataset and 7 best features from Em

best_diamond_features = ['carat', 'y', 'x', 'z', 'clarity', 'color', 'cut', 'table',
best_emission_features = ['TIT', 'TEY', 'CDP', 'GTEP', 'AFDP', 'TAT', 'year']


finalDiamonds = diamondsScaled[best_diamond_features].copy()
finalEmission = scaledEmission[best_emission_features].copy()
```

Answer 7

Feature selection is a crucial step in building models. Redundant features which dont help much in regression are useless and increasing the training time and sometimes can worse the results as well given that more features require more data.

I used mutual_info_regression (MI) and f_regression (FR) as a comparative measure on how much features are necessary and which are those important features. This score is computed for features in both the datasets with respect to the target variables. And were sorted in descending order to get the most important features.

MIR Scores: Diamonds dataset

carat- 1.652888
y - 1.421612
x - 1.412996
z - 1.361523
clarity - 0.217263
color - 0.136824
cut - 0.056896
table -0.035091
depth -0.032032

We can see that carat is most important feature and depth is least important.

MIR Scores: Emissions dataset

TIT - 0.538170
TEY - 0.495486
CDP - 0.474436
GTEP - 0.445490
AFDP - 0.278731
TAT - 0.159627
year - 0.123832
AT - 0.104885
AP - 0.042072
AH - 0.025524

TIT, TEY are quite important features compared to AH and AP

Then I used SelectKBest() from sklearn which takes K best features using a comparison score for finding these K best as input. I ran selected K best features ranging from K = 1 to K = total number of features and used these features to the neg root mean squared error from a simple Linear Regression fit with a 10 fold cross validation.

The errors were plotted for both strategies to choose the K best features with Linear Regression as shown above.

For diamonds dataset, as the number of best features increase both the strategies (MI and FR) are giving lower errors. Until they saturate in region of (5 - 8) features and then a sudden increase when all feature are considered is observed. For diamonds dataset:

Best Errors occured were achieved at 8 features from both the strategies

For Emissions dataset, the trend is not uniform, its quite abrupt i.e. errors are increasing and decreasing with changes in number of features.

Best Errors occured were achieved at 7 features from (MR strategy) and 9 FROM (FR strategy)

All 9 features weere chosen for diamonds dataset.
7 best features were chosen based on MR strategy for Emissions dataset

Creating the train, validation and test datasets

Since I am going to use K-fold cross validation, therefore no validation dataset is required.

Splitting train and test data as 80/ 20 split.

In [42]:

```
X_train_D, X_test_D, y_train_D, y_test_D = train_test_split(finalDiamonds,
                                                            targetDiamonds, test_siz

X_train_E, X_test_E, y_train_E, y_test_E = train_test_split(finalEmission,
                                                            targetEmission, test_siz


# Not scaled
X_train_D_NS, X_test_D_NS, y_train_D_NS, y_test_D_NS = train_test_split(diamondsEncc
                                                            targetDiamonds, test_siz

X_train_E_NS, X_test_E_NS, y_train_E_NS, y_test_E_NS = train_test_split(emissionData
                                                            targetEmission, test_siz
```

Linear Regression

- Without Regularization
- Lasso
- Ridge

Answer 8

Objective functions:

1). Ordinary Least Squares:

$$Cost(J)(W) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

2). Lasso Regression:

$$Cost(J)(W) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2 + \lambda \sum_{j=1}^{n} |W|$$

3). Ridge Regression:

$$Cost(J)(W) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2 + \lambda \sum_{j=1}^{n} |W|^2$$

where:
y: True values. (1 X 1)
W: Weights corresponding to each feature. (n+1 X 1) (including bias feature)
x_i: Training sample (n+1 X 1)
m: number of training examples
n: number of features
$h_\theta(x_i) = \sum_{j=0}^{j=n} w_j x_{ij}$
λ: regularization strength

Effects of each regularization scheme:

Regularization makes on replace one stable and stable rules don't overfit. Thus regularization helps in reducing the overfitting. Also since we are adding a positive value to the cost function the training error generally is higher compared to non regularized schemes but this helps in reducing variance. Regularization increases the biasness.

For Ridge L2 or Tikhonov regression, its assumed that the data doesnt contain much outliers. It reduces the effect sof collinear features. It makes features sparse, as weights are distributed among features due to variance effect of the squared weight term. This is because subgradient of $|w|^2$ depends on both its sign and magnitude.

For Lasso L1 regularization, the weights of non important features are set to 0, thus Lasso regression helps in feature selection and weights learned are dense. This is because subgradient of $|w|$ depends on just it's sign and thus it shrinks features weigths to 0.

In [43]:

```
scores_D_LR = cross_validate(LinearRegression(), X_train_D, y_train_D,
                         scoring=['neg_root_mean_squared_error'], cv=10)

scores_E_LR = cross_validate(LinearRegression(), X_train_E, y_train_E,
                         scoring=['neg_root_mean_squared_error'], cv=10)
```

In [44]:

```
print("Diamonds Dataset")
print("Linear Regression Ordinary LS train time error: ",  scores_D_LR['fit_time'].m
print("Linear Regression Ordinary LS validation error: ",  scores_D_LR['test_neg_roc

print("Emission Dataset")
print("Linear Regression Ordinary LS train time error: ",  scores_E_LR['fit_time'].m
print("Linear Regression Ordinary LS validation error: ",  scores_E_LR['test_neg_roc
```

```
Diamonds Dataset
Linear Regression Ordinary LS train time error:  0.008420395851135253
Linear Regression Ordinary LS validation error:  -1220.4102585663093
Emission Dataset
Linear Regression Ordinary LS train time error:  0.0027782440185546873
Linear Regression Ordinary LS validation error:  -1.4314262802990454
```

Regularized Regression

In [45]:

```
pipe_LR = Pipeline([
    ('reg', Ridge(random_state=42))
])

param_grid = {
    'reg': [Lasso(random_state=42), Ridge(random_state=42)],
    'reg__alpha': [10.0**x for x in np.arange(-4,4)]
}
```

In [46]:

```
grid_LR_D = GridSearchCV(pipe_LR, param_grid=param_grid, cv=10, verbose=1, n_jobs=-1
                         scoring='neg_root_mean_squared_error', return_train_score=True)
```

```
Fitting 10 folds for each of 16 candidates, totalling 160 fits
```

In [47]:

```
print("Best regularization scheme: Diamonds Dataset (Scaled)")
print(grid_LR_D.best_estimator_)
print(grid_LR_D.best_params_)
print("Test RMSE: ", grid_LR_D.best_score_)
print("Train RMSE: ", np.max(grid_LR_D.cv_results_['mean_train_score']))
```

```
Best regularization scheme: Diamonds Dataset (Scaled)
Pipeline(steps=[('reg', Lasso(random_state=42))])
{'reg': Lasso(random_state=42), 'reg__alpha': 1.0}
Test RMSE:  -1220.0209647539734
Train RMSE:  -1219.751337929666
```

In [48]:

```
grid_LR_E = GridSearchCV(pipe_LR, param_grid=param_grid, cv=10, verbose=1, n_jobs=-1
                         scoring='neg_root_mean_squared_error', return_train_score=True)
```

```
Fitting 10 folds for each of 16 candidates, totalling 160 fits
```

In [49]:

```
print("Best regularization scheme: Emission Dataset (Scaled)")
print(grid_LR_E.best_estimator_)
print(grid_LR_E.best_params_)
print("Test RMSE: ", grid_LR_E.best_score_)
print("Train RMSE: ", np.max(grid_LR_E.cv_results_['mean_train_score']))
```

```
Best regularization scheme: Emission Dataset (Scaled)
Pipeline(steps=[('reg', Ridge(random_state=42))])
{'reg': Ridge(random_state=42), 'reg__alpha': 1.0}
Test RMSE:  -1.4314248070422952
Train RMSE:  -1.4413075208468773
```

In [50]:

```
# Not scaled

grid_LR_D_NS = GridSearchCV(pipe_LR, param_grid=param_grid, cv=10, verbose=1, n_jobs
                         scoring='neg_root_mean_squared_error', return_train_score=True)

grid_LR_E_NS = GridSearchCV(pipe_LR, param_grid=param_grid, cv=10, verbose=1, n_jobs
                         scoring='neg_root_mean_squared_error', return_train_score=True)
```

```
Fitting 10 folds for each of 16 candidates, totalling 160 fits
Fitting 10 folds for each of 16 candidates, totalling 160 fits
```

In [51]:

```
print("Best regularization scheme: Diamonds Dataset (Not Scaled)")
print(grid_LR_D_NS.best_estimator_)
print(grid_LR_D_NS.best_params_)
print("Test RMSE: ", grid_LR_D_NS.best_score_)
print("Train RMSE: ", np.max(grid_LR_D_NS.cv_results_['mean_train_score']))
```

```
Best regularization scheme: Diamonds Dataset (Not Scaled)
Pipeline(steps=[('reg', Lasso(random_state=42))])
{'reg': Lasso(random_state=42), 'reg__alpha': 1.0}
Test RMSE:  -1220.0891834559864
Train RMSE:  -1219.7513379296968
```

In [52]:

```
print("Best regularization scheme: Emission Dataset (Not Scaled)")
print(grid_LR_E_NS.best_estimator_)
print(grid_LR_E_NS.best_params_)
print("Test RMSE: ", grid_LR_E_NS.best_score_)
print("Train RMSE: ", np.max(grid_LR_E_NS.cv_results_['mean_train_score']))
```

```
Best regularization scheme: Emission Dataset (Not Scaled)
Pipeline(steps=[('reg', Ridge(random_state=42))])
{'reg': Ridge(random_state=42), 'reg__alpha': 1.0}
Test RMSE:  -1.4314257974922078
Train RMSE:  -1.4413075208468737
```

In [53]:

```
# Testing best models on test set.

# Testing scaled datasets
D_test_LR = Lasso(random_state=42).fit(X_train_D, y_train_D).predict(X_test_D)
E_test_LR = Ridge(random_state=42).fit(X_train_E, y_train_E).predict(X_test_E)
```

In [54]:

```
print("Mean squared Error for Diamonds dataset Scaled: ", mean_squared_error(y_test_
print("Mean squared Error for Emissions dataset Scaled: ", mean_squared_error(y_test
```

```
Mean squared Error for Diamonds dataset Scaled:  1228.5647060894441
Mean squared Error for Emissions dataset Scaled:  1.5187751212181595
```

In [55]:

```
D_test_LR_NS = Lasso(random_state=42).fit(X_train_D_NS, y_train_D_NS).predict(X_test
E_test_LR_NS = Ridge(random_state=42).fit(X_train_E_NS, y_train_E_NS).predict(X_test
```

In [56]:

```
# Testing not scaled datasets

print("Mean squared Error for Diamonds dataset Not Scaled: ", mean_squared_error(y_t
print("Mean squared Error for Emissions dataset Not Scaled: ", mean_squared_error(y_
```

```
Mean squared Error for Diamonds dataset Not Scaled:  1228.572661933617
4
Mean squared Error for Emissions dataset Not Scaled:  1.51876779617129
78
```

In [57]:

```
# P-valued regression

p_D_OLS = OLS(y_train_D, X_train_D).fit()
print(p_D_OLS.pvalues.sort_values(ascending=True))
```

```
carat       0.000000e+00
clarity     0.000000e+00
color      1.121368e-153
depth       8.405394e-14
x           1.998451e-11
table       2.074957e-07
cut         4.264263e-01
y           6.231743e-01
z           7.936728e-01
dtype: float64
```

In [58]:

```
p_E_OLS = OLS(y_train_E, X_train_E).fit()
print(p_E_OLS.pvalues.sort_values(ascending=True))
```

```
TAT     1.947095e-139
year    4.021402e-118
TEY      1.535366e-34
CDP      2.860426e-12
TIT      1.033824e-08
GTEP     1.060634e-05
AFDP     4.214264e-01
dtype: float64
```

Answer 9

To find the best regularization scheme:

Results were compared for all three methods: Ordinary Least squares(OLS), Lasso, Ridge Rgression. For Lasso and Ridge regression GridSearchCV was with regularization hyperparameter being varied in the range of 10^[-4, 4] with 10 Fold cross validation with neg rmse as comparison metric.

Results are given below: Since the target variable is not scaled, the RMSE is high but its just the interpretation. If scaled values are again scaled back it would be in this range only

Diamonds Dataset (OLS):

Cross validation is used to get the below results (10 folds for OLS) Linear Regression Ordinary LS train time error: 0.008420395851135253
Linear Regression Ordinary LS validation error: -1220.4102585663093
Emission Dataset (OLS):

Linear Regression Ordinary LS train time error: 0.0027782440185546873
Linear Regression Ordinary LS validation error: -1.4314262802990454

Best regularization scheme: Diamonds Dataset (Scaled)

Pipeline(steps=[('reg', Lasso(random_state=42))])
{'reg': Lasso(random_state=42), 'reg__alpha': 1.0}
Test RMSE: -1220.0209647539734
Train RMSE: -1219.751337929666

Best regularization scheme: Emission Dataset (Scaled)

Pipeline(steps=[('reg', Ridge(random_state=42))])
{'reg': Ridge(random_state=42), 'reg__alpha': 1.0}
Test RMSE: -1.4314248070422952
Train RMSE: -1.4413075208468773

For diamonds dataset Lasso regression is better and for emission dataset ridge regression is better.
Overall For diamonds dataset: Regularization(Lasso) gives slightly low validation error.

Overall For emission dataset: Regularization(Ridge) is quite same as OLS.

Answer 10

Effect of feature scaling:

Same approach is used as for Question 9 for determining best params for Not scaled features.

Best regularization scheme: Diamonds Dataset (Not Scaled)
Pipeline(steps=[('reg', Lasso(random_state=42))])
{'reg': Lasso(random_state=42), 'reg__alpha': 1.0}
Test RMSE: -1220.0891834559864
Train RMSE: -1219.7513379296968

Best regularization scheme: Emission Dataset (Not Scaled)
Pipeline(steps=[('reg', Ridge(random_state=42))])
{'reg': Ridge(random_state=42), 'reg__alpha': 1.0}
Test RMSE: -1.4314257974922078
Train RMSE: -1.4413075208468737

Testing best hypothesis on testsets (Scaled and not scaled):

Mean squared Error for Diamonds dataset Scaled: 1228.5647060894441
Mean squared Error for Emissions dataset Scaled: 1.5187751212181595

Mean squared Error for Diamonds dataset Not Scaled: 1228.5726619336174
Mean squared Error for Emissions dataset Not Scaled: 1.5187677961712978

Scaling helps in reducing validation RMSE as seen above. For diamonds dataset best test error is reduced by 0.01 and for emission dataset this is improved by 0.0001. Thus there are not very drastic imporvements to be seen since the distributions remain same for the features and weights thus just scale linearly. But this would be

such since the distributions remain same for the features and weights thus just scale linearly. But this would be eventually helpful for faster convergence and in cases where the data has many outliers.

Answer 11

In terms of regression, p-value is a measure to test the validity of Null Hypothesis i.e. a feature is not related to the target variable. If p-value > 0.05 , then the Null hypothesis is proved right and alternate hypothesis that feature does makes a change in target variable is validated. If p-value < 0.05 then probability that a feature would be assigned a 0 weight is low (Alternate hypothesis.)

statsmodels generally give p-values as one of the outputs. Its OLS is used to find the p-values for features in both the datasets.

p-values for diamond dataset:
carat- 0.000000e+00
clarity - 0.000000e+00
color - 1.121368e-153
depth - 8.405394e-14
x - 1.998451e-11
table - 2.074957e-07
cut - 4.264263e-01
y - 6.231743e-01
z - 7.936728e-01

p-values for emission dataset:
TAT - 1.947095e-139
year - 4.021402e-118
TEY - 1.535366e-34
CDP - 2.860426e-12
TIT - 1.033824e-08
GTEP - 1.060634e-05
AFDP - 4.214264e-01

The results are quite comparable to the correlation matrix plotted above. We see no features have p-values > 0.05 this is because these features are the best features found from f-regression above.

Polynomial Regression

In [62]:

```
# Polynomial features pipeline

# Checking 10 degree polynomial at max

PR_pipe_diamonds = Pipeline([
    ('features', PolynomialFeatures()),
    ('model', Ridge(random_state=42))
])

PR_pipe_emission = Pipeline([
    ('features', PolynomialFeatures()),
    ('model', Ridge(random_state=42))
])

PR_grid = {
    'features_degree': np.arange(1, 11, 1),
    'model__alpha': [10.0** x for x in np.arange(-4, 4)]
}
```

In [63]:

```
degrees = np.arange(1,6,1)

poly_diamonds = Pipeline([
    ('feats', PolynomialFeatures()),
    ('model', Ridge(random_state=42))
])

poly_emission = Pipeline([
    ('feats', PolynomialFeatures()),
    ('model', Ridge(random_state=42))
])

poly_grid = {
    'feats__degree': degrees,
    'model__alpha': [10.0**x for x in np.arange(-4, 4)]
}
```

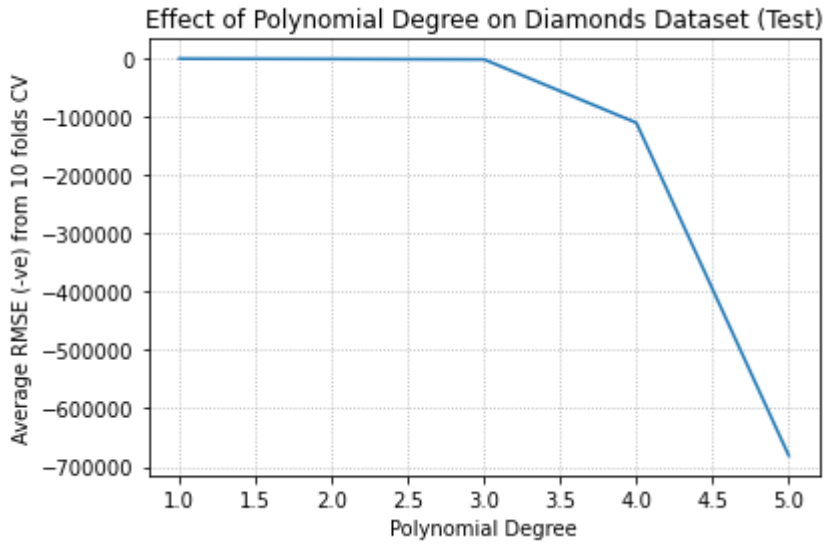
In [64]:

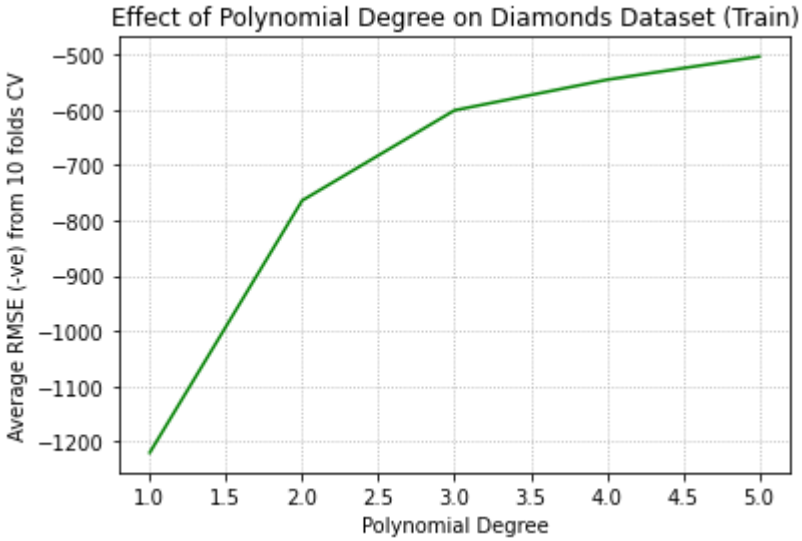
```
grid_poly_D = GridSearchCV(poly_diamonds, param_grid=poly_grid, cv=10, n_jobs=-1, ve
                    scoring='neg_root_mean_squared_error', return_train_score=True)
```

```
Fitting 10 folds for each of 40 candidates, totalling 400 fits
```

In [65]:

```
poly_diamonds_GD = pd.DataFrame(grid_poly_D.cv_results_)[['mean_test_score','mean_tr

diamond_score_P = []
diamond_train_P = []
diamond_alpha_P = []

for i in degrees:
    diamond_score_P.append((poly_diamonds_GD.loc[poly_diamonds_GD['param_feats__degr
    diamond_train_P.append((poly_diamonds_GD.loc[poly_diamonds_GD['param_feats__degr
    diamond_alpha_P.append(float(poly_diamonds_GD['param_model__alpha'][
        (poly_diamonds_GD.loc[poly_diamonds_GD['param_feats__degree'] == i])
        [['mean_test_score']].idxmax()].to_numpy()))


plt.plot(degrees, diamond_score_P)
plt.grid(linestyle=':')
plt.xlabel('Polynomial Degree')
plt.ylabel('Average RMSE (-ve) from 10 folds CV')
plt.title('Effect of Polynomial Degree on Diamonds Dataset (Test)')
plt.show()

plt.plot(degrees, diamond_train_P, 'g')
plt.grid(linestyle=':')
plt.xlabel('Polynomial Degree')
plt.ylabel('Average RMSE (-ve) from 10 folds CV')
plt.title('Effect of Polynomial Degree on Diamonds Dataset (Train)')
plt.show()
```





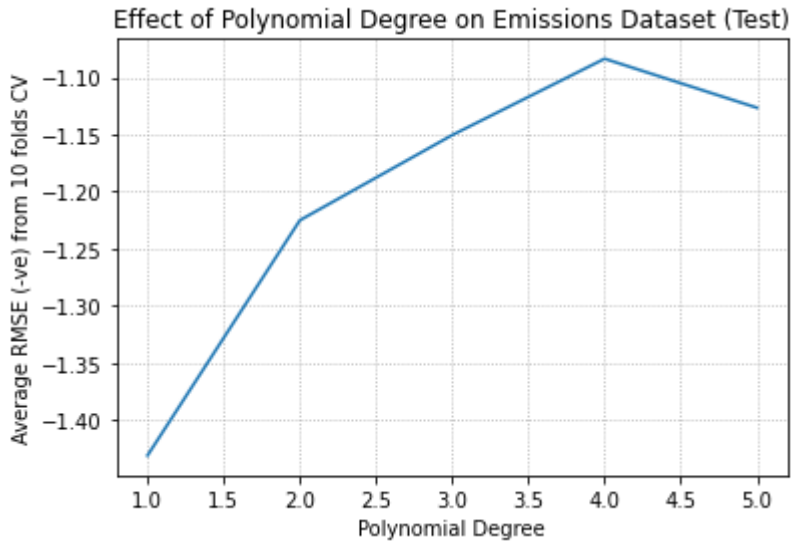
In [66]:

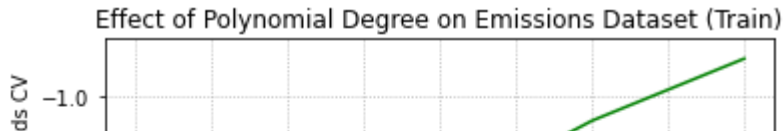
```
grid_poly_E = GridSearchCV(poly_emission, param_grid=poly_grid, cv=10, verbose=1, n_
                           scoring='neg_root_mean_squared_error', return_train_score=True)
```

```
Fitting 10 folds for each of 40 candidates, totalling 400 fits
```

In [67]:

```
poly_emission_GD = pd.DataFrame(grid_poly_E.cv_results_)[['mean_test_score','mean_tr

emission_score_P = []
emission_train_P = []
emission_alpha_P = []

for i in degrees:
    emission_score_P.append((poly_emission_GD.loc[poly_emission_GD['param_feats__deg
    emission_train_P.append((poly_emission_GD.loc[poly_emission_GD['param_feats__deg
    emission_alpha_P.append(float(poly_emission_GD['param_model__alpha'][
        (poly_emission_GD.loc[poly_emission_GD['param_feats__degree'] == i])
        [['mean_test_score']].idxmax()].to_numpy()))


plt.plot(degrees, emission_score_P)
plt.grid(linestyle=':')
plt.xlabel('Polynomial Degree')
plt.ylabel('Average RMSE (-ve) from 10 folds CV')
plt.title('Effect of Polynomial Degree on Emissions Dataset (Test)')
plt.show()

plt.plot(degrees, emission_train_P, 'g')
plt.grid(linestyle=':')
plt.xlabel('Polynomial Degree')
plt.ylabel('Average RMSE (-ve) from 10 folds CV')
plt.title('Effect of Polynomial Degree on Emissions Dataset (Train)')
plt.show()
```





In [68]:

```
kbest_P = SelectKBest(score_func=f_regression, k='all')
diamonds_Poly_K = kbest_P.fit_transform(X_train_D, y_train_D)
col_names = X_train_D.columns[kbest_P.get_support()]

best_params = grid_poly_D.best_estimator_.get_params()
best_coefs = best_params['model'].coef_
best_feature_names = list(col_names)
best_names = best_params['feats'].get_feature_names(best_feature_names)
best_sorted_indices = np.argsort(-abs(best_coefs))
salient_features_diamonds =[best_names[i] for i in best_sorted_indices[:10]]
print ('Top 10 sorted Salient features (Diamonds):', salient_features_diamonds)
```

```
Top 10 sorted Salient features (Diamonds): ['carat', 'x', 'clarity',
'color', 'depth', 'table', 'z', 'cut', 'y', '1']
```

In [69]:

```
kbest_P = SelectKBest(score_func=f_regression, k='all')
emission_Poly_K = kbest_P.fit_transform(X_train_E, y_train_E)
col_names = X_train_E.columns[kbest_P.get_support()]

best_params = grid_poly_E.best_estimator_.get_params()
best_coefs = best_params['model'].coef_
best_feature_names = list(col_names)
best_names = best_params['feats'].get_feature_names(best_feature_names)
best_sorted_indices = np.argsort(-abs(best_coefs))
salient_features_emissions =[best_names[i] for i in best_sorted_indices]
print ('Top 10 sorted Salient features (Emissions):', salient_features_emissions[:10
```

```
Top 10 sorted Salient features (Emissions): ['TIT^2 TEY^2', 'TIT^3 TE
Y', 'TIT^2 TEY TAT', 'TIT^2 TEY CDP', 'TIT CDP TAT^2', 'TEY^2 CDP^2',
'TIT CDP^2 TAT', 'TIT^2 TEY year', 'TIT TEY^2 CDP', 'TIT TEY^2 TAT']
```

Answer 12

A pipeline is created which tranforms simple features to polynomial features of higher orders and thus a simple regression can be made to work like a polynomial regression.

I used Ridge regression for deciding the best polynomial degree and to analysize the effects of higher order features.

The number of degrees used are 1- 6.

A 10 fold cross validation with GridSearchCV is used to determine the impact of high order features.

For Diamonds dataset: polynomial degree 1 features came out to be the best performing features.
Top 10 performing feature from polynomial degree 1:
['carat', 'x', 'clarity', 'color', 'depth', 'table', 'z', 'cut', 'y', '1']

For emission dataset: polynomial degree 4 features came out to be the best performing features.
Top 10 performing features from polynomial degree 4:
['TIT^2 TEY^2', 'TIT^3 TEY', 'TIT^2 TEY TAT', 'TIT^2 TEY CDP', 'TIT CDP TAT^2', 'TEY^2 CDP^2', 'TIT CDP^2 TAT', 'TIT^2 TEY year', 'TIT TEY^2 CDP', 'TIT TEY^2 TAT']

The top-10 features are selected using SelectKBest with f_regression score for the best polynomial degree found.

Answer 13

Polynomial degree of 1 is best for diamond dataset. Polynomial degree of 4 is best for emission dataset.

The best polynomial degrees are found from the 4 plots plotted above for the avg. train and validation errors across folds for different degrees. A total of 6 degrees were analysed.

We see a decrease in train error as the degree is increased and validation error decreases as well but then after certain increase in degree, the train error keeps on decreasing but the validation error starts to increase. This phenomenon is related to overfitting. After a certain degree the model starts to overfit on the training data and doesnt generalize well.

This optimal point is chosen as the best degree for each dataset where both the train and validation errors are low at same time.

Answer 14

For the diamond dataset, yes it makes sense to construct new feature combinations but then after analysing the 6 degrees and all sorts of combinations for the features in diamonds degree 1 polynomial turns out to be the best as sen from above plots. No such boost in accuracy is seen from other polynomial degrees. It's observed degrees > 1 are increasing the validation errors always for this dataset. One of the reasons might be that some featuresin diamond dataset like depth and table are alreadt complex functions (mentioned in spec) of other features and building new features on top of ruins the learnable traits about the dataset.

Neural Networks

In [70]:

```
pipe_NN_diamonds = Pipeline([
    ('model', MLPRegressor(random_state=42))
])

pipe_NN_emissions = Pipeline([
    ('model', MLPRegressor(random_state=42))
])

units = [(10), (30), (50), (10, 10), (30, 30), (50, 50),
         (10, 30, 50), (10, 10, 10), (30, 30, 30), (50, 50, 50)]

params_NN = {
    'model__activation': ['tanh', 'relu'],
    'model__hidden_layer_sizes': units,
    'model__alpha': [10.0**x for x in np.arange(-3, 1)]
}
```

In [71]:

```
grid_NN_diamonds = GridSearchCV(pipe_NN_diamonds, param_grid=params_NN, cv = 3, verb
                                scoring='neg_root_mean_squared_error',return_train_sc
```

```
Fitting 3 folds for each of 80 candidates, totalling 240 fits
```

In [72]:

```
grid_NN__diamonds_df = pd.DataFrame(grid_NN_diamonds.cv_results_)[[
    'param_model__activation', 'param_model__alpha',
    'param_model__hidden_layer_sizes', 'mean_test_score', 'mean_train_score']]

print("Best NN configuration for Diamonds dataset: ", grid_NN_diamonds.best_params_)
print("Best NN test score for Diamonds dataset: ", grid_NN_diamonds.best_score_)
print("Best NN train score for Diamonds dataset: ", max(grid_NN__diamonds_df['mean_t
```

```
Best NN configuration for Diamonds dataset:  {'model__activation': 're
lu', 'model__alpha': 0.001, 'model__hidden_layer_sizes': (30, 30, 30)}
Best NN test score for Diamonds dataset:  -674.3019330168225
Best NN train score for Diamonds dataset:  -575.1597739765449
```

In [73]:

```
grid_NN_emissions = GridSearchCV(pipe_NN_emissions, param_grid=params_NN, cv = 3, ve
                                scoring='neg_root_mean_squared_error',return_train_sc
```

```
Fitting 3 folds for each of 80 candidates, totalling 240 fits
```

In [74]:

```
grid_NN_emissions_df = pd.DataFrame(grid_NN_emissions.cv_results_)[[
    'param_model__activation', 'param_model__alpha',
    'param_model__hidden_layer_sizes', 'mean_test_score', 'mean_train_score']]

print("Best NN configuration for emissions dataset: ", grid_NN_emissions.best_params
print("Best NN test score for emissions dataset: ", grid_NN_emissions.best_score_)
print("Best NN train score for emissions dataset: ", max(grid_NN_emissions_df['mean_
```

```
Best NN configuration for emissions dataset:  {'model__activation': 'r
elu', 'model__alpha': 0.01, 'model__hidden_layer_sizes': (30, 30, 30)}
Best NN test score for emissions dataset:  -1.0858278843127869
Best NN train score for emissions dataset:  -0.9199761645723751
```

In [75]:

```
# Testing best NN configurations on the test set.

D_test_NN = MLPRegressor(random_state=42, activation='relu',
                         alpha=0.001, hidden_layer_sizes=(30, 30, 30)).fit(X_train_D



E_test_NN = MLPRegressor(random_state=42, activation='relu',
                         alpha=0.001, hidden_layer_sizes=(30, 30, 30)).fit(X_train_E
```

```
/Users/gauravsingh/miniforge3/envs/tensorflow/lib/python3.8/site-packa
ges/sklearn/neural_network/_multilayer_perceptron.py:614: ConvergenceW
arning: Stochastic Optimizer: Maximum iterations (200) reached and the
optimization hasn't converged yet.
  warnings.warn(
/Users/gauravsingh/miniforge3/envs/tensorflow/lib/python3.8/site-packa
ges/sklearn/neural_network/_multilayer_perceptron.py:614: ConvergenceW
arning: Stochastic Optimizer: Maximum iterations (200) reached and the
optimization hasn't converged yet.
  warnings.warn(
```

In [76]:

```
print("Mean NN squared Error for Diamonds dataset: ", mean_squared_error(y_test_D, D
print("Mean NN squared Error for Emissions dataset: ", mean_squared_error(y_test_E,
```

```
Mean NN squared Error for Diamonds dataset:  596.5803439579738
Mean NN squared Error for Emissions dataset:  1.1254830951737709
```

Answer 15

Comparison of MLP regressor and Linear regression on test set.

For diamonds dataset:
Mean (MLP) squared Error for Diamonds dataset: 596.5803439579738
Mean (Linear Regression) squared Error for Diamonds dataset Scaled: 1228.5647060894441

For Emission dataset: Mean (MLP) squared Error for Emissions dataset: 1.1254830951737709
Mean (Linear Regression) squared Error for Emissions dataset Scaled: 1.5187751212181595

From above results we can see that Neural networks do much better compared to Linear regressors. The reasons are:

1) NN have much more capacity compared to linear regressors. The number of trainable parameters are exponentially higher than that of Linear regressors thus they can learn much more from data.

2) NN have much inbuild non-linearity due to the fact that they use non-linear activation functions which helps them to analyse and learn more complex features.

Answer 16

GridSearchCV with 10 folds is used to fine tune the hypterparameters of the NN.
Following grid is used:

Hidden units and layers:
(10), (30), (50), (10, 10), (30, 30), (50, 50), (10, 30, 50), (10, 10, 10), (30, 30, 30), (50, 50, 50)

Activations: tanh and relu
Weight_decay: [10^-3, 10]

The best parameters came out to be:

For emissions dataset:
{'model__activation': 'relu', 'model__alpha': 0.01, 'model__hidden_layer_sizes': (30, 30, 30)}

For diamonds dataset:
{'model__activation': 'relu', 'model__alpha': 0.001, 'model__hidden_layer_sizes': (30, 30, 30)}

Answer 17

The activation function to be used for regression tasks is usually None since the output can range from $-\infty$ to ∞. We can also use a linear activation function.

tanh, relu, sigmoid should be avoided. Since they are bounded by values. Like relu takes values between [0, inf], sigmoid takes values between [0 - 1], tanh is bounded by [-1, 1].

Thus for regression only linear or no activation makes sense.
I used no activation function for the above questions.

Answer 18

There are several risks associated with making the NN too deep:

1). More data requirement. Since there are more number of trainable parameters, more training data is usually required to justify this big network and to train it optimally.

2). Deep neural networks can lead to overfitting on the training set.
3). More compute power is required to train large NN which is not always feasible.
4). As the gradient has to propagate from deeper layers to the input layer, the gradients might zero out in between and can thus lead to vanishing gradients (0 gradient) problem where no learning is done.

5). During forward passing of data, the acitvations can reach to very high values which will be propagated to infinte values if a network is very deep. This leads to problem of exploding gradients where no learning is possible.

Random Forest

In [77]:

```
pipe_RF_diamonds = Pipeline([
    ('model', RandomForestRegressor(random_state=42, oob_score=True))
])

pipe_RF_emissions = Pipeline([
    ('model', RandomForestRegressor(random_state=42, oob_score=True))
])


params_RF_D = {
    'model__max_features': np.arange(1, 10, 1),
    'model__n_estimators': np.arange(10, 210, 10),
    'model__max_depth': np.arange(1, 10, 1)
}

params_RF_E = {
    'model__max_features': np.arange(1, 8, 1),
    'model__n_estimators': np.arange(10, 210, 10),
    'model__max_depth': np.arange(1, 8, 1)
}
```

In [78]:

```
grid_RF_diamonds = GridSearchCV(pipe_RF_diamonds, param_grid=params_RF_D, cv = 10, v
                                scoring='neg_root_mean_squared_error',return_train_sc
```

```
Fitting 10 folds for each of 1620 candidates, totalling 16200 fits
```

In [79]:

```
grid_RF_diamonds_df = pd.DataFrame(grid_RF_diamonds.cv_results_)[[
    'param_model__max_depth', 'param_model__max_features',
    'param_model__n_estimators', 'mean_test_score', 'mean_train_score']]

print("Best Random Forest configuration for Diamonds dataset: ", grid_RF_diamonds.be
print("Best Random Forest test score for Diamonds dataset: ", grid_RF_diamonds.best_
print("Best Random Forest train score for Diamonds dataset: ", max(grid_RF_diamonds_
```

```
Best Random Forest configuration for Diamonds dataset:  {'model__max_d
epth': 9, 'model__max_features': 8, 'model__n_estimators': 200}
Best Random Forest test score for Diamonds dataset:  -587.695660706638
9
Best Random Forest train score for Diamonds dataset:  -518.17601649604
53
```

In [80]:

```
grid_RF_emissions = GridSearchCV(pipe_RF_emissions, param_grid=params_RF_E, cv = 10,
                                 scoring='neg_root_mean_squared_error',return_train_sc
```

```
Fitting 10 folds for each of 980 candidates, totalling 9800 fits
```

In [86]:

```
grid_RF_emissions_df = pd.DataFrame(grid_RF_emissions.cv_results_)[[
    'param_model__max_depth', 'param_model__max_features',
    'param_model__n_estimators', 'mean_test_score', 'mean_train_score']]

print("Best Random Forest configuration for Diamonds dataset: ", grid_RF_emissions.b
print("Best Random Forest test score for Diamonds dataset: ", grid_RF_emissions.best
print("Best Random Forest train score for Diamonds dataset: ", max(grid_RF_emissions
```

```
Best Random Forest configuration for Diamonds dataset:  {'model__max_d
epth': 7, 'model__max_features': 3, 'model__n_estimators': 90}
Best Random Forest test score for Diamonds dataset:  -1.14768795031255
26
Best Random Forest train score for Diamonds dataset:  -0.9499263129891
538
```

In [225]:

```
# Testing best RF configurations on the test set.

D_test_RF = RandomForestRegressor(random_state=42, max_depth=9, max_features=8, oob_
                                  n_estimators=200).fit(X_train_D, y_train_D).predict



E_test_RF = RandomForestRegressor(random_state=42, max_depth=7, max_features=3, oob_
                                  n_estimators=90).fit(X_train_E, y_train_E).predict(
```

In [226]:

```
print("OOB Score for Best RF Regressor for Diamonds dataset: ",
      RandomForestRegressor(random_state=42, max_depth=9, max_features=8, oob_score=
                                  n_estimators=200).fit(X_train_D, y_train_D).oob_scc

print("OOB Score for Best RF Regressor for Emissions dataset: ",
      RandomForestRegressor(random_state=42, max_depth=7, max_features=3, oob_score=
                                  n_estimators=90).fit(X_train_E, y_train_E).oob_scor
```

```
OOB Score for Best RF Regressor for Diamonds dataset:  0.9784128558462
39
OOB Score for Best RF Regressor for Emissions dataset:  0.729510175638
0225
```

In [88]:

```
print("Mean Random Forest squared Error for Diamonds dataset: ", mean_squared_error(
print("Mean Random Forest squared Error for Emissions dataset: ", mean_squared_error
```

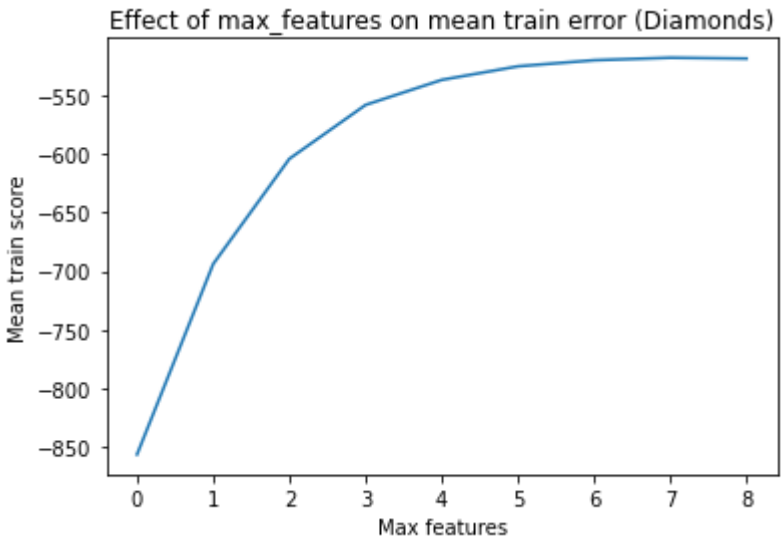
```
Mean Random Forest squared Error for Diamonds dataset:  580.9118090841
522
Mean Random Forest squared Error for Emissions dataset:  1.16158515416
64983
```

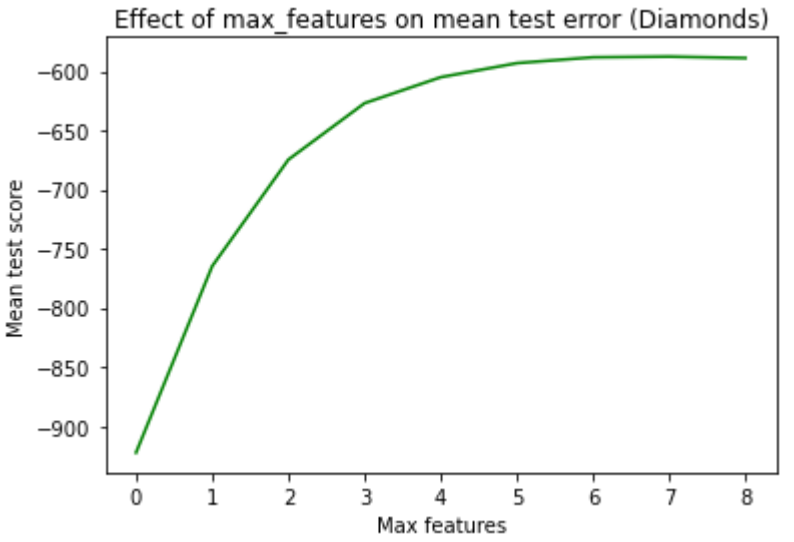
In [89]:

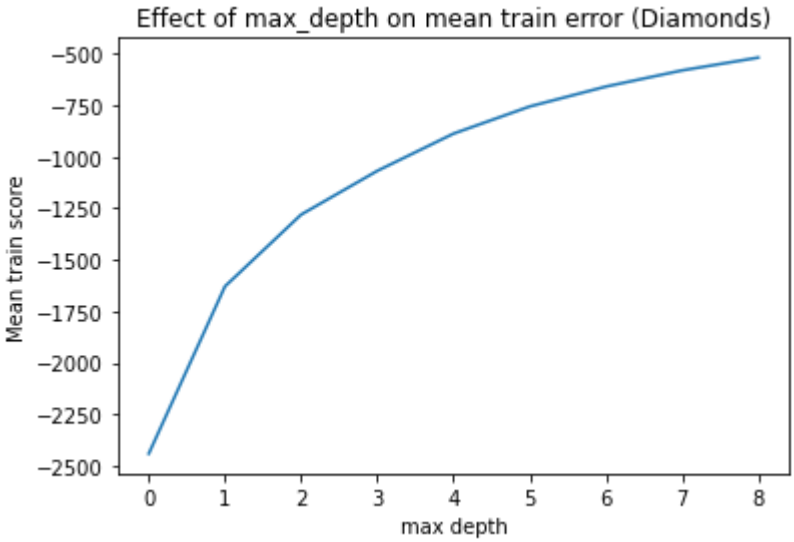
```
# Effects on diamonds dataset
#Effect of max_features, n_estimators, depth.

mx_features = grid_RF_diamonds_df[(grid_RF_diamonds_df['param_model__max_depth'] ==
                                   (grid_RF_diamonds_df['param_model__n_estimators']


mx_depth = grid_RF_diamonds_df[(grid_RF_diamonds_df['param_model__max_features'] ==
                                   (grid_RF_diamonds_df['param_model__n_estimators']


mx_estimators = grid_RF_diamonds_df[(grid_RF_diamonds_df['param_model__max_depth'] =
                                   (grid_RF_diamonds_df['param_model__max_features']
```

In [90]:

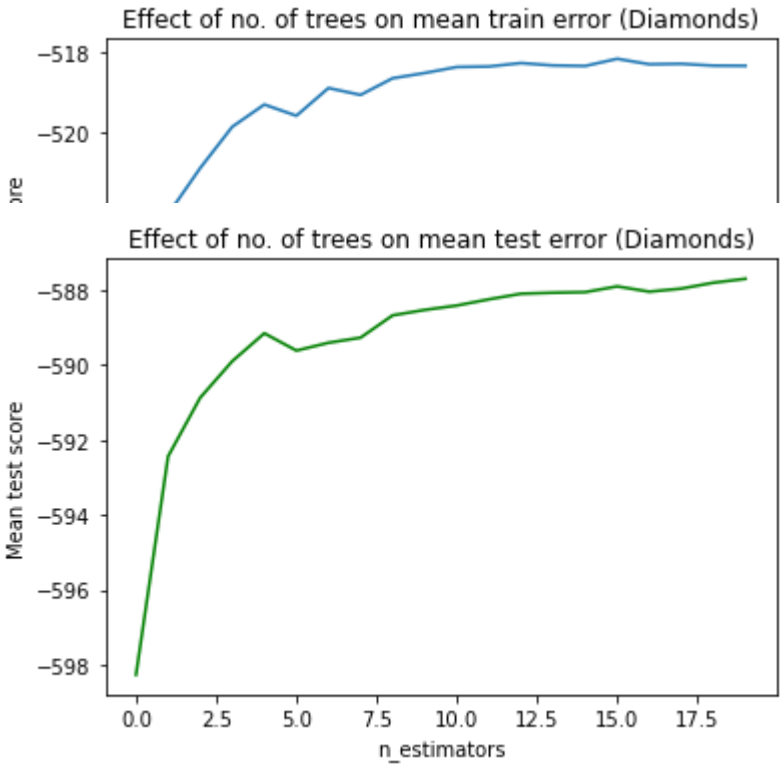
```
plt.plot(np.arange(len(mx_features)), mx_features['mean_train_score'])
plt.title("Effect of max_features on mean train error (Diamonds)")
plt.xlabel("Max features")
plt.ylabel("Mean train score")
plt.show()

plt.plot(np.arange(len(mx_features)), mx_features['mean_test_score'], 'g')
plt.title("Effect of max_features on mean test error (Diamonds)")
plt.xlabel("Max features")
plt.ylabel("Mean test score")
plt.show()

plt.plot(np.arange(len(mx_depth)), mx_depth['mean_train_score'])
plt.title("Effect of max_depth on mean train error (Diamonds)")
plt.xlabel("max depth")
plt.ylabel("Mean train score")
plt.show()

plt.plot(np.arange(len(mx_depth)), mx_depth['mean_test_score'], 'g')
plt.title("Effect of max_depth on mean test error (Diamonds)")
plt.xlabel("max depth")
plt.ylabel("Mean test score")
plt.show()

plt.plot(np.arange(len(mx_estimators)), mx_estimators['mean_train_score'])
plt.title("Effect of no. of trees on mean train error (Diamonds)")
plt.xlabel("n_estimators")
plt.ylabel("Mean train score")
plt.show()

plt.plot(np.arange(len(mx_estimators)), mx_estimators['mean_test_score'], 'g')
plt.title("Effect of no. of trees on mean test error (Diamonds)")
plt.xlabel("n_estimators")
plt.ylabel("Mean test score")
plt.show()
```




Effect of max_features on mean test error (Diamonds)
Mean test score
−600
−650
−700
−750
−800
−850
−900
0
1
2
3
4
5
6
7
8
Max features


Effect of max_depth on mean train error (Diamonds)
Mean train score
−500
−750
−1000
−1250
−1500
−1750
−2000
−2250
−2500
0
1
2
3
4
5
6
7
8
max depth


Effect of max_depth on mean test error (Diamonds)
Mean test score
−500
−750
−1000
−1250
−1500
−1750
−2000
−2250
−2500
0
1
2
3
4
5
6
7
8
max depth



Answer 19

For Random Forest Regressor, a GridSearchCV is run for 10 folds using below hyperparameters field.

'max_features': np.arange(1, 10, 1)
'n_estimators': np.arange(10, 210, 10)
'max_depth': np.arange(1, 10, 1)

By finetuning the best model params for Diamonds dataset came out to be:

{'max_depth': 9, 'max_features': 8, 'n_estimators': 200}

Best Random Forest validation score for Diamonds dataset: -587.6956607066389
Best Random Forest train score for Diamonds dataset: -518.1760164960453

For Emission dataset the best model params are:

Best Random Forest configuration for Diamonds dataset:
{'max_depth': 7, 'max_features': 3, 'n_estimators': 90}

Best Random Forest validation score for Diamonds dataset: -1.1476879503125526
Best Random Forest train score for Diamonds dataset: -0.9499263129891538

The effect of each of these hyperparameters on the Diamonds dataset is studied by varying each parameter while keeping other two fixed from the best found hypterparameters from gridsearch. Plots were plotted for change in these parameters and the train and validation errors as shown above to study their effects.

By analysing the above plots above we see that:
For number of trees: as it increases the training error and test error both decrease upto certain level.
For max depth an max features, as they increase the train error and test error both decrease but we can see that there is somewhat increase in the test error for both of these. Thus if allowed to grow these hyperparameters can result in high test error which implies these have some regularization effect. Max

depth increases the capacity of the model with more splits and fine graining to work on the features which can results in overfitting. Changing the max features changes the amount of information being passed to each tree and thus limits to overtrain on data

Answer 20

On evaluating the best obtained Random forest regressors for both diamond and emission adatasets we see that the errors are quite low even lower than the Neural networks got for the diamonds dataset. Thus Random forest does very well compared to Linear Regressors.

Mean Random Forest squared Error for Diamonds dataset: 580.9118090841522
Mean Random Forest squared Error for Emissions dataset: 1.1615851541664983

Random Forest is an ensemble method. It uses bagging to boost its performance. With large number of uncorrelated trees Random forests are quite immune to overfitting as they take majority voting from a large number of uncorrelated trees. More trees with different randomly picked features means there is always something new learnt by these each tree and thus the combined results work much well compared to low capacity models like Linear regressors. RF also doesnt require feature scaling to perform well.

In [91]:

```
# Choosing best RF params but with max depth of 4.

vis_tree_D = RandomForestRegressor(random_state=42, max_depth=4, max_features=8,
                                   n_estimators=200, oob_score=True).fit(X_train_D, y_



vis_tree_E = RandomForestRegressor(random_state=42, max_depth=4, max_features=3,
                                   n_estimators=90, oob_score=True).fit(X_train_E, y_t
```

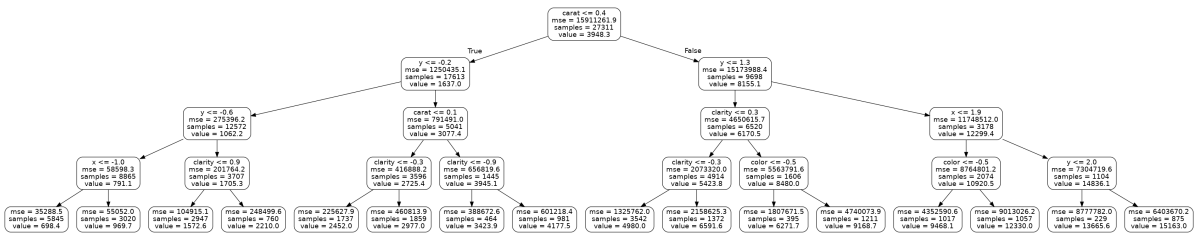
In [97]:

```
tree = vis_tree_D.estimators_[1]
export_graphviz(tree, out_file = 'tree.dot', feature_names = X_train_D.columns, rour
(graph, ) = pydot.graph_from_dot_file('tree.dot')
```

In [113]:

```
Image('tree.png', width = 1200, height = 1200)
```

Out[113]:



Answer 21

For tree visualisation, first we set the max_depth of the Random forest regressor as 4 and trained it with other best hypterparameters. Then an estimator was randomly chosen from the trees in the regressor , export_graphviz from sklearn is used to save the tree structure as .dot file which is then read used pydot and

visualised as shown above.

I did it for diamonds dataset. Where I observe that:

carat feature has the most information gain and thus feature is at the root to split the tree (carat <= 0.4).
Next feature with maximum information gain is 'y' which makes the next split at level 2 (y <= 0.2) (y <=1.3)
For level 3, 'y', 'carat', 'clarity' and 'x' perform the splits.

Comparing it with the most important features found earlier for diamonds dataset:
carat, y, x, z, clarity, color, cut, table, depth (in decreasing order)

In top 5 features, we can see that there are 4 overlapping features and even there order is very same. Thus we can safely conclude that the splitting features in higher levels are the important features.

LightGBM and CatBoost on Diamonds dataset

- LightGBM
- Choosing Diamonds dataset

In [134]:

```
optGBM = BayesSearchCV(
    lgb.LGBMRegressor(random_state=42, verbose=1, n_jobs=-1),
    {
        'boosting_type': ['gbdt', 'dart','rf'],
        'num_leaves': np.arange(20,1000,20),
        'max_depth': np.arange(1,100,10),
        'n_estimators': np.arange(40,2000,100),
        'reg_alpha': [10.0**x for x in np.arange(-4,1)],
        'reg_lambda': [10.0**x for x in np.arange(-4,1)],
        'subsample': np.arange(0.1,1,0.2),
        'subsample_freq': np.arange(0,50,10),
        'min_split_gain': [10.0**x for x in np.arange(-4,0)]
    },
    n_iter=20,
    cv=10,
    n_jobs=-1,
    verbose=1,
    random_state=42,
    scoring = 'neg_root_mean_squared_error',
    return_train_score = True
)
```

In [135]:

```
optGBMRes = optGBM.fit(X_train_D, y_train_D)
```

```
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overh
ead of testing was 0.000625 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1274
[LightGBM] [Info] Number of data points in the train set: 38837, numbe
r of used features: 9
[LightGBM] [Info] Start training from score 3934.823184
```

In [136]:

```
print('Best parameters (Diamonds):',optGBMRes.best_params_,',Test RMSE:',optGBMRes.b
print('Train RMSE:',min(optGBMRes.cv_results_['mean_train_score']))
```

```
Best parameters (Diamonds): OrderedDict([('boosting_type', 'dart'),
('max_depth', 11), ('min_split_gain', 0.01), ('n_estimators', 940),
('num_leaves', 500), ('reg_alpha', 0.001), ('reg_lambda', 0.001), ('su
bsample', 0.5000000000000001), ('subsample_freq', 30)]) ,Test RMSE: -5
40.7715904210484
Train RMSE: -1071.0585848019841
```

In [137]:

```
lgbresults = pd.DataFrame(optGBMRes.cv_results_)
```

In [146]:

```
lgbresults
```

Out[146]:

std_score_time	param_boosting_type	param_max_depth	param_min_split_gain	param_n_estimato
1.373599	gbdt	71	0.1	6
2.705964	rf	81	0.001	18
1.141868	gbdt	81	0.0001	8
1.955603	rf	21	0.01	15
1.946620	rf	41	0.01	14
0.597885	gbdt	81	0.0001	4
2.629745	gbdt	71	0.001	17
1.800186	gbdt	81	0.001	16
0.904058	rf	61	0.1	8
0.765644	dart	71	0.01	5
0.693844	dart	61	0.01	5
0.544078	rf	51	0.01	4
0.943487	dart	41	0.01	9
0.026940	dart	41	0.01	3
0.019038	dart	1	0.01	13
0.224595	gbdt	51	0.0001	3
0.120260	gbdt	11	0.1	6
1.351968	dart	11	0.01	9
1.390358	dart	51	0.1	9
2.005639	dart	81	0.001	11

Catboost regression

In [ ]:

```
optCatReg = BayesSearchCV(
    CatBoostRegressor(random_state=42,verbose=1,thread_count=-1,bootstrap_type='Baye
    {
        'num_trees': np.arange(10,1000,100),
        'l2_leaf_reg': [10.0**x for x in np.arange(-3,1)],
        'max_depth': np.arange(2,10,2),
        'grow_policy': ['SymmetricTree','Depthwise'],
        'score_function': ['Cosine','L2']
    },
    n_iter=20,
    cv=5,
    n_jobs=-1,
    verbose=1,
    random_state=42,
    scoring = 'neg_root_mean_squared_error',
    return_train_score = True
)
```

In [ ]:

```
optCatReg.fit(X_train_D, y_train_D)
```

In [142]:

```
# BayesSearchCV(cv=5,
#               estimator=<catboost.core.CatBoostRegressor object at 0x7f6d57b599d0>
#               n_iter=20, n_jobs=-1, random_state=42, return_train_score=True,
#               scoring='neg_root_mean_squared_error',
#               search_spaces={'grow_policy': ['SymmetricTree', 'Depthwise'],
#                              'l2_leaf_reg': [0.001, 0.01, 0.1, 1.0],
#                              'max_depth': array([2, 4, 6, 8]),
#                              'num_trees': array([ 10, 110, 210, 310, 410, 510, 610
#                              'score_function': ['Cosine', 'L2']},
#               verbose=1)
```

In [ ]:

```
print('Best parameters:', optCatReg.best_params_,',Test RMSE:',optCatReg.best_score_
print('Train RMSE:',min(optCatReg.cv_results_['mean_train_score']))
```

Best parameters: OrderedDict([('grow_policy', 'Depthwise'), ('l2_leaf_reg', 1.0), ('max_depth', 8), ('num_trees', 310), ('score_function', 'L2')]) ,

Test RMSE: -527.3136022252144

Train RMSE: -958.5509871349407

In [145]:

```
# Catboost was run on Colab and best results were saved as .csv and loaded here for

catBoostResults = pd.read_csv('catboostres.csv')
```

In [144]:

```
catBoostResults
```

Out[144]:

	Unnamed: 0	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_grow_policy
0	0	28.306721	8.467917	0.118426	0.027587	Depthwise
1	1	8.484351	2.140567	0.019278	0.004023	SymmetricTree
2	2	4.686728	1.204742	0.058549	0.016292	Depthwise
3	3	9.401615	2.390694	0.023860	0.007754	SymmetricTree
4	4	8.372677	2.044060	0.019523	0.006997	SymmetricTree
5	5	1.543984	0.384160	0.009458	0.003466	SymmetricTree
6	6	7.246397	1.760729	0.021159	0.006360	SymmetricTree
7	7	7.762688	1.909945	0.021989	0.006846	SymmetricTree
8	8	8.991404	2.373658	0.020156	0.007235	SymmetricTree
9	9	8.399700	2.361296	0.070034	0.019306	Depthwise
10	10	10.272867	2.700489	0.109918	0.029916	Depthwise
11	11	35.574814	10.456367	0.138563	0.024318	Depthwise
12	12	30.333591	9.287379	0.120621	0.031857	Depthwise
13	13	38.639761	11.444176	0.146963	0.032160	Depthwise
14	14	28.591520	8.333720	0.117799	0.023409	Depthwise
15	15	29.037464	8.562886	0.112029	0.028173	Depthwise
16	16	30.057472	9.023919	0.113893	0.023495	Depthwise
17	17	29.937697	8.885262	0.119551	0.022902	Depthwise
18	18	29.291734	8.689906	0.115579	0.028901	Depthwise
19	19	28.630000	8.166355	0.121003	0.031421	Depthwise

20 rows × 27 columns

Answer 22 - 23

I chose the diamonds dataset to experiment with the hyperparameters of the CatBoost and LightGBM regressors. The features were already reduced from above questions and thus only the best features are used here which are total 7 in number.

For LightGBM following hyperparameters are used:

1) Boosting type - ['gbdt', 'dart','rf'].
GBDT: Gradient boosting decision tree, DART: Dropouts meet multiple additive regression trees, RF: random forest.

GBDT treats individual decision trees within an ensemble as weak learners, with the first tree aiming to fit the feature set to target variables and the succeeding trees aiming to reduce the residual error between the predicted target variable and ground truth target variable of preceding trees, with the entire ensemble trained via backpropagation of error gradients.

DART solves the overspecialization problem in GBDT by introducing dropout which has bagging effect.

RF is same as discussed in above questions.

2) Number of leaves: [20, 1000] with step of 20
Similar to the depth of each tree, the number of leaf nodes acts as a regularizer and performance contributor, moderate values improve both fitting and generalizability, while very large values lead to overfitting by treating each data point including noise in the dataset as a leaf in the tree.

3) max depth: [1, 100] with step of 10
Large depth leads to more number of splits and no. of leaf nodes and thus more capacity and alsso has regularization effect.

4) number of trees [40, 2000] with step of 100
This means number of boosted trees to consider for making the regression prediction. Increasing the number of trees improves and stabilizes model.

5) alpha: [10^-4, 10]
This is same as L1 regularization.

6) lambda: [10^4, 10]
This is same as L2 regularization.

7) subsample: [0.1, 1] with step of 0.2
No. of rows to be randomly sampled from dataset for fitting each tree (bagging). More subsampling causes overfitting.

8) subsample_freq: [0, 50] step of 10
Controls the freq to perform bagging.

9) min_split_gain: [10^-4, 10]
This controls when to split and grow a tree. This is chosen by considering the gain obtained for a feature.

For CatBoost following hyperparameters are used:

Less no. of parameters are chosen and a Kfold of 5 was applied since, I am having a M1 mac which doesnt support catboost and due to time and memory constriants on colab less num. of specs are chosen to experiment with.

1) number of trees: [10, 1000] at step of 100
Specifies the number of boosted trees to fit for the task. Increasing number of trees doesnt lead to overfitting since they are uncorrelated.

2) l2 leaf regularization: [10^-3, 10]
This is same as tikhonov regularization penalty.

3) max depth: [2, 10] at step of 2
Max depth that can be achieved by a tree. Max depth increases the model capacity and has regularization effect.

4) grow policy: ['SymmetricTree','Depthwise']
This specifies the strategy to grow trees from leaves. Catboost grows a balanced tree.

5) score function: ['Cosine','L2']
This is used for only 'symmetric' and 'depthwise' grow policy. It is used to choose the tree candidates when adding it to the ensemble.

Hyperparameter tuning:

The best hyperparameters from the given above field of search for LightGBM and CatBoost is found using Bayesian Optimization which is an approach that uses Bayes Theorem to direct the search in order to find the minimum or maximum of an objective function. Optimizing the model using bayesian can create a balance between exploration and exploitation of hyperparameters.

For CatBoost 20 iteration with CV of 5 is used, while for LightGBM 20 iteration with CV of 10 is used. The best hyperparameters found are reported below:

Best Hyperparameters For LightGBM: (Diamonds Dataset)

[('boosting_type', 'dart'), ('max_depth', 11), ('min_split_gain', 0.01), ('n_estimators', 940), ('num_leaves', 500), ('reg_alpha', 0.001), ('reg_lambda', 0.001), ('subsample', 0.5000000000000001), ('subsample_freq', 30)] ,

Test RMSE: -540.7715904210484
Train RMSE: -1071.0585848019841

Best Hyperparameters For CatBoost: (Diamonds dataset)

[('grow_policy', 'Depthwise'), ('l2_leaf_reg', 1.0), ('max_depth', 8), ('num_trees', 310), ('score_function', 'L2')]

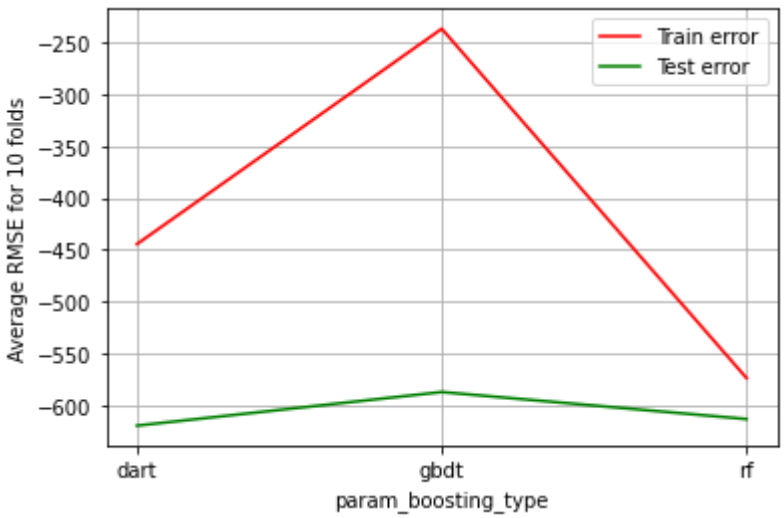
Test RMSE: -527.3136022252144
Train RMSE: -958.5509871349407

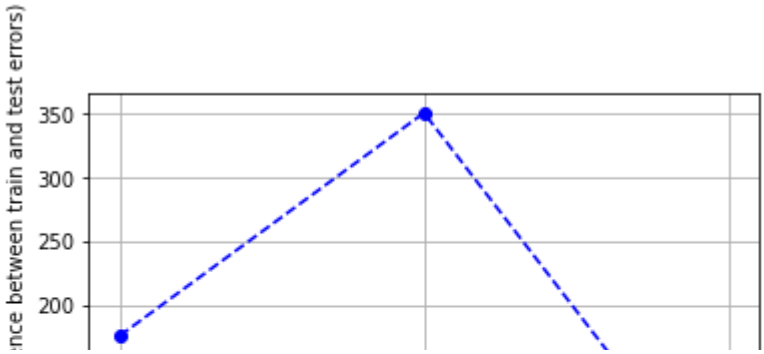
We can see that LightGBM performs better compared to CatBoost in terms of test and train RMSE errors.

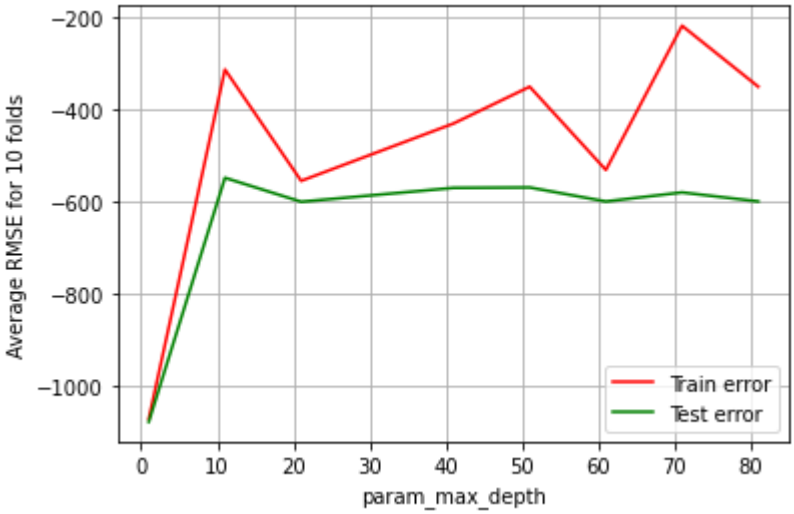
Analysing the effects of parameters of LigthBGM

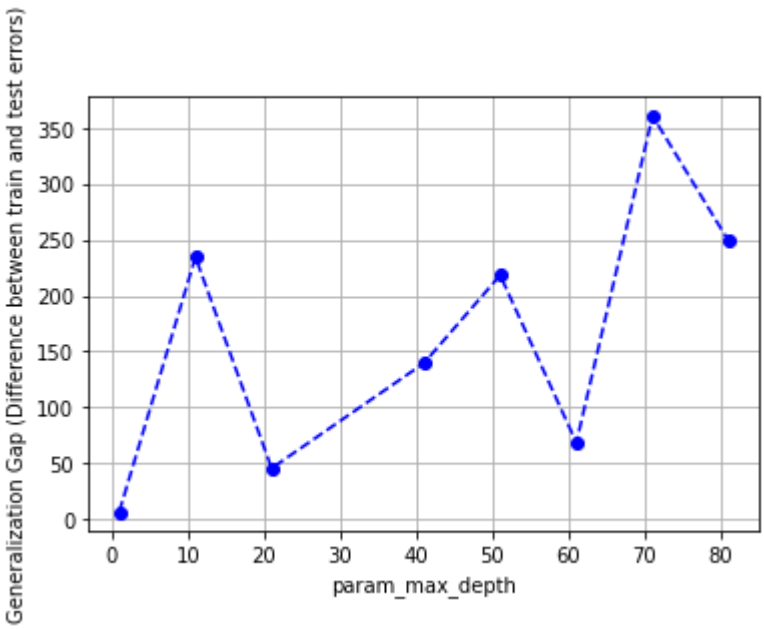
In [223]:

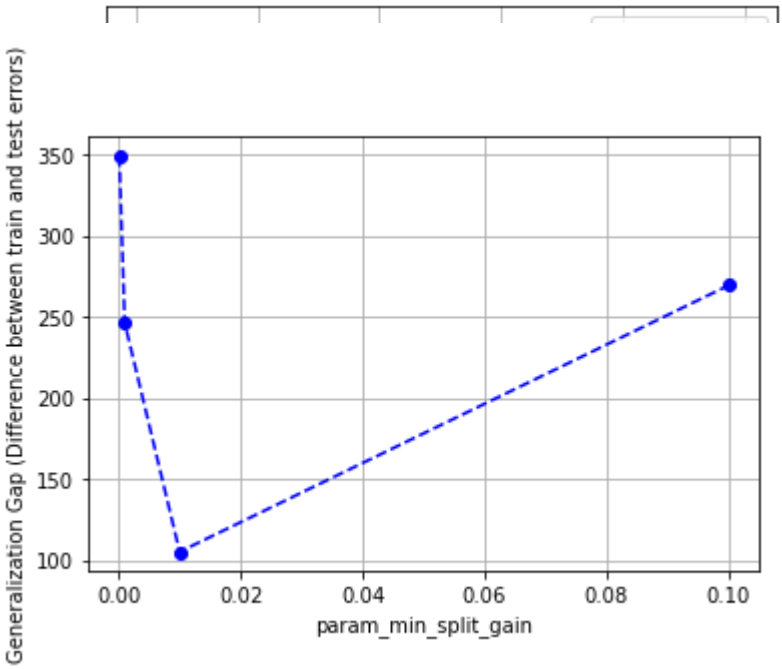
```
lgbParams = ['param_boosting_type', 'param_max_depth', 'param_min_split_gain',
       'param_n_estimators', 'param_num_leaves', 'param_reg_alpha',
       'param_reg_lambda', 'param_subsample', 'param_subsample_freq']

i = 1
for param in lgbParams:
    values = np.sort(lgbresults[param].unique().tolist())

    mean_train_error = []
    mean_test_error = []

    for val in values:
        mean_train_error.append(lgbresults[lgbresults[param] == val]['mean_train_scc
        mean_test_error.append(lgbresults[lgbresults[param] == val]['mean_test_score

    difference = np.abs(np.array(mean_train_error) - np.array(mean_test_error))
    plt.figure(i)
    i+= 1
    plt.grid()
    plt.plot(values, mean_train_error, 'r')
    plt.plot(values, mean_test_error, 'g')
    plt.xlabel(str(param))
    plt.ylabel("Average RMSE for 10 folds")
    plt.legend(['Train error', 'Test error'])
    plt.show()

    plt.grid()
    plt.plot(values, difference, 'b--', marker='o')
    plt.ylabel("Generalization Gap (Difference between train and test errors)")
    plt.xlabel(str(param))
    plt.show()
    i+=1
```

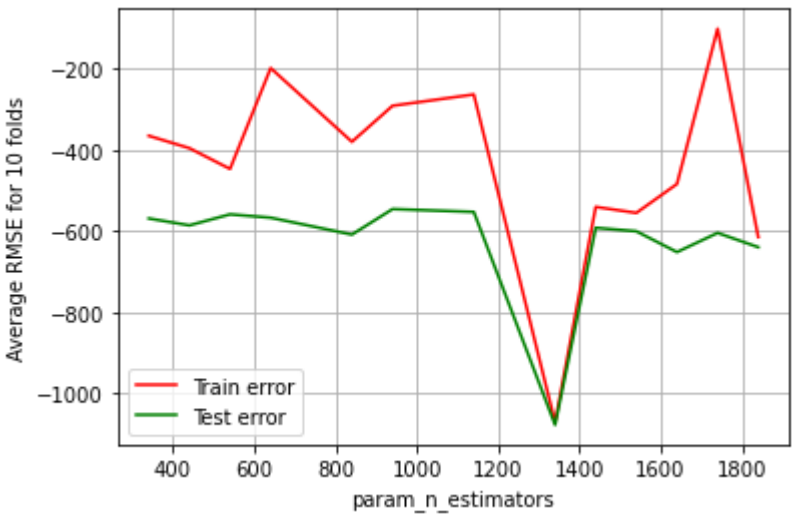


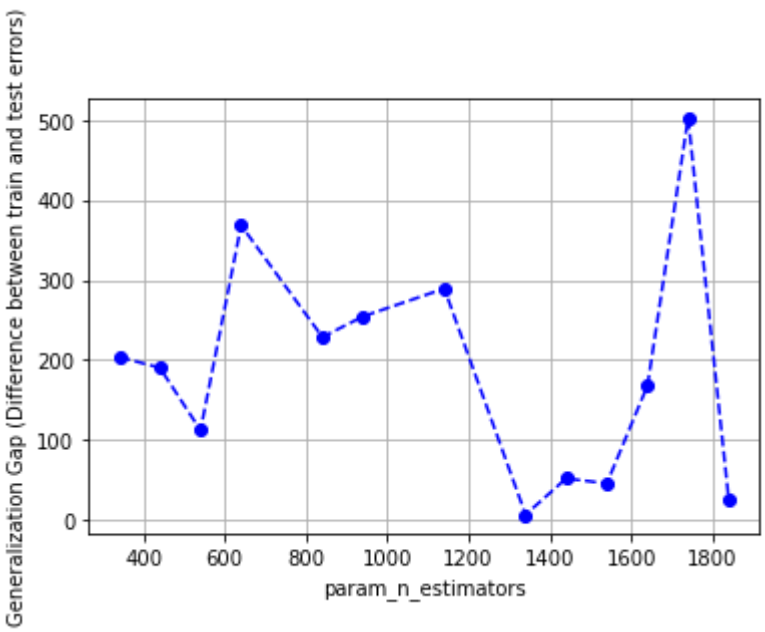

nce between train and test errors)
350
300
250
200

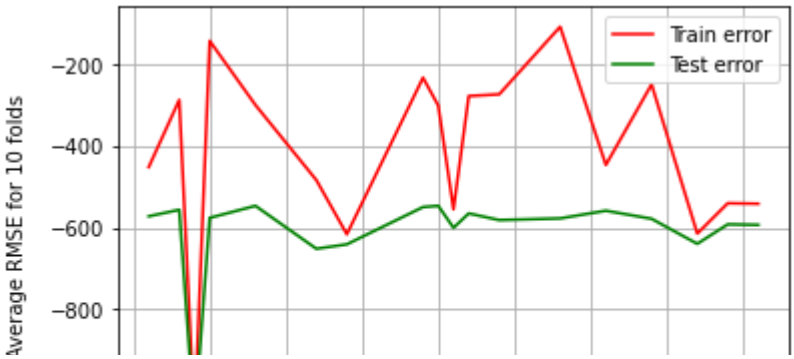

Average RMSE for 10 folds
−200
−400
−600
−800
−1000
Train error
Test error
0
10
20
30
40
50
60
70
80
param_max_depth

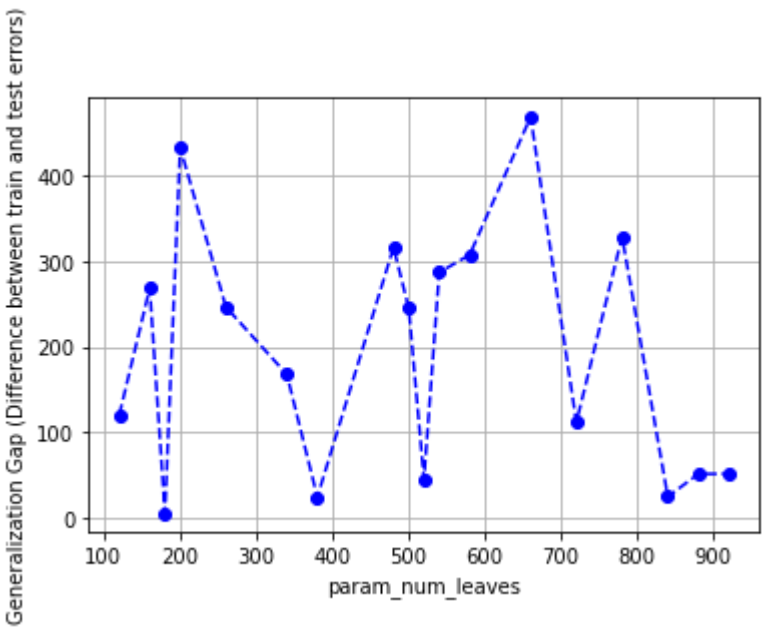

Generalization Gap (Difference between train and test errors)
350
300
250
200
150
100
50
0
0
10
20
30
40
50
60
70
80
param_max_depth

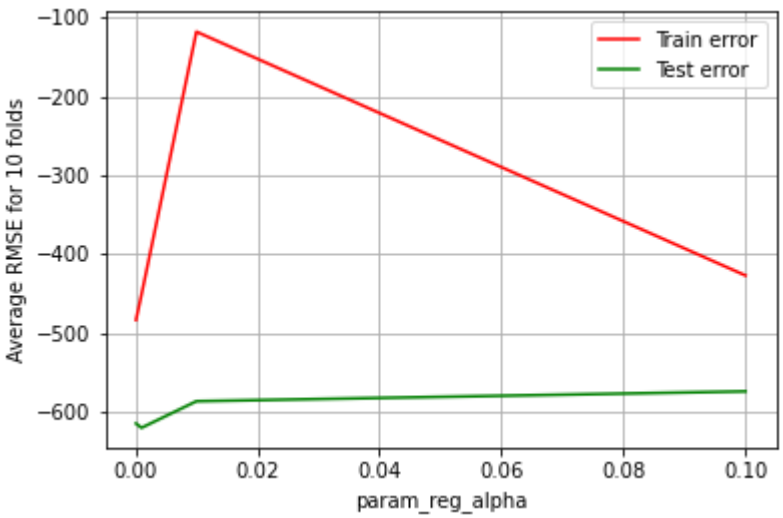

Generalization Gap (Difference between train and test errors)
350
300
250
200
150
100
0.00
0.02
0.04
0.06
0.08
0.10
param_min_split_gain

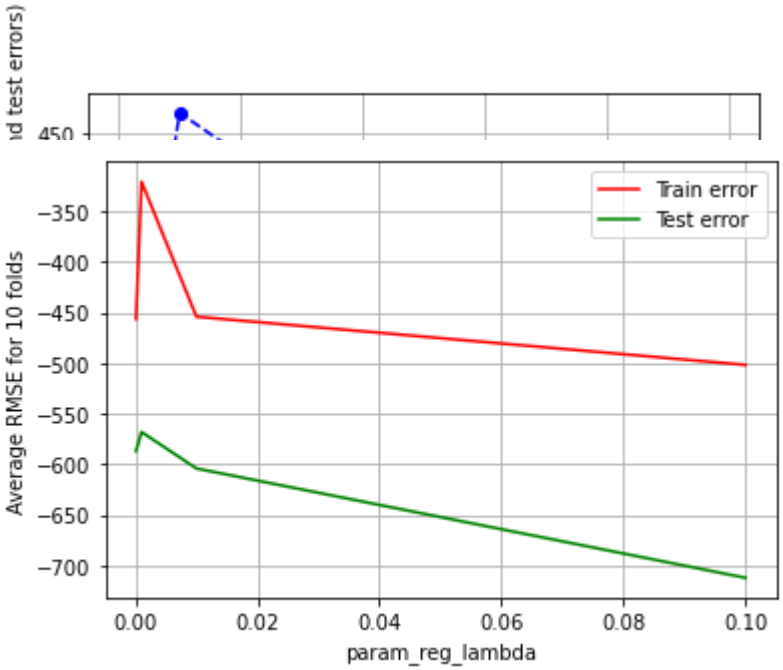

Average RMSE for 10 folds
−200
−400
−600
−800
−1000
Train error
Test error
400
600
800
1000
1200
1400
1600
1800
param_n_estimators

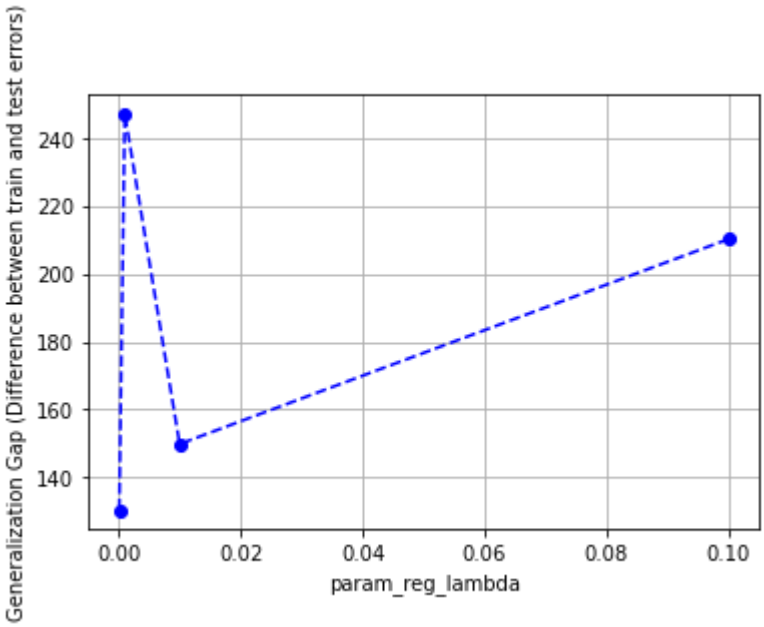

Generalization Gap (Difference between train and test errors)
500
400
300
200
100
0
400
600
800
1000
1200
1400
1600
1800
param_n_estimators

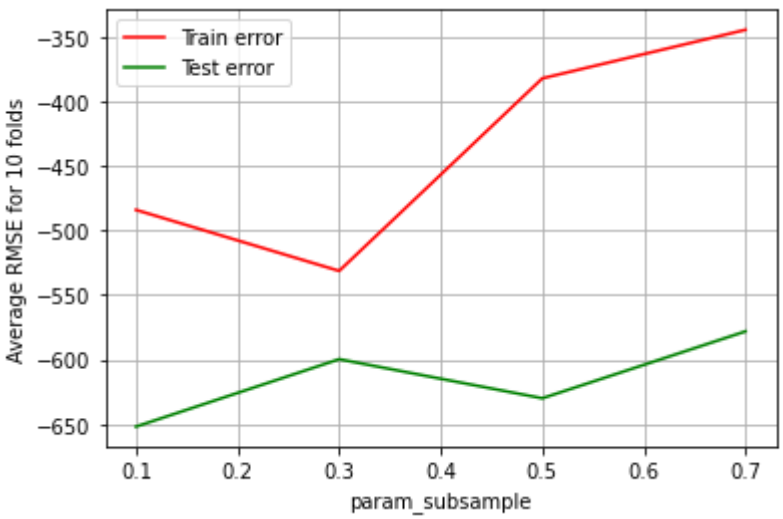

Train error
Test error
−200
−400
−600
−800

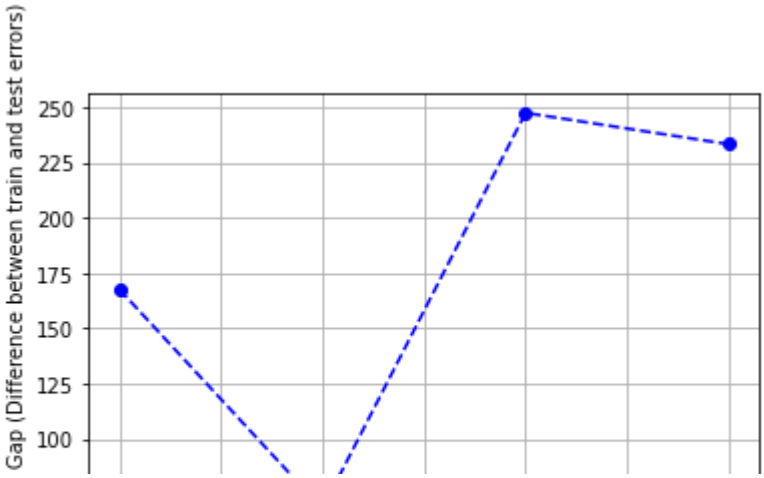

Generalization Gap (Difference between train and test errors)
0
100
200
300
400
100
200
300
400
500
600
700
800
900
param_num_leaves

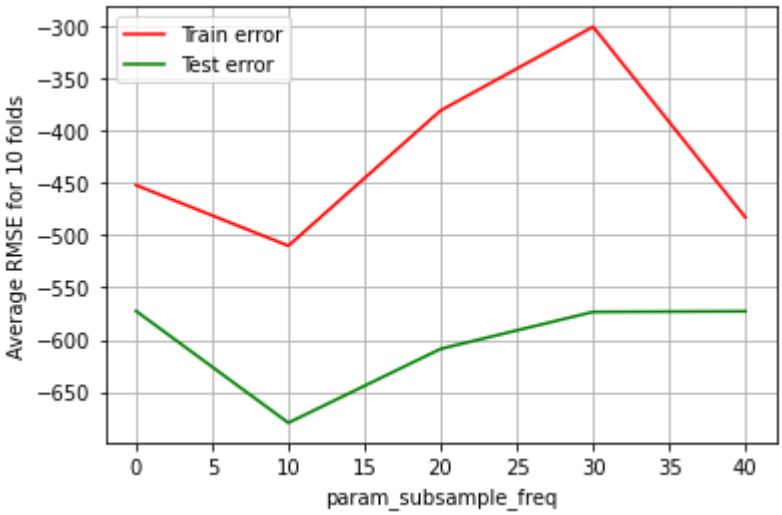

Train error
Test error
Average RMSE for 10 folds
−100
−200
−300
−400
−500
−600
0.00
0.02
0.04
0.06
0.08
0.10
param_reg_alpha

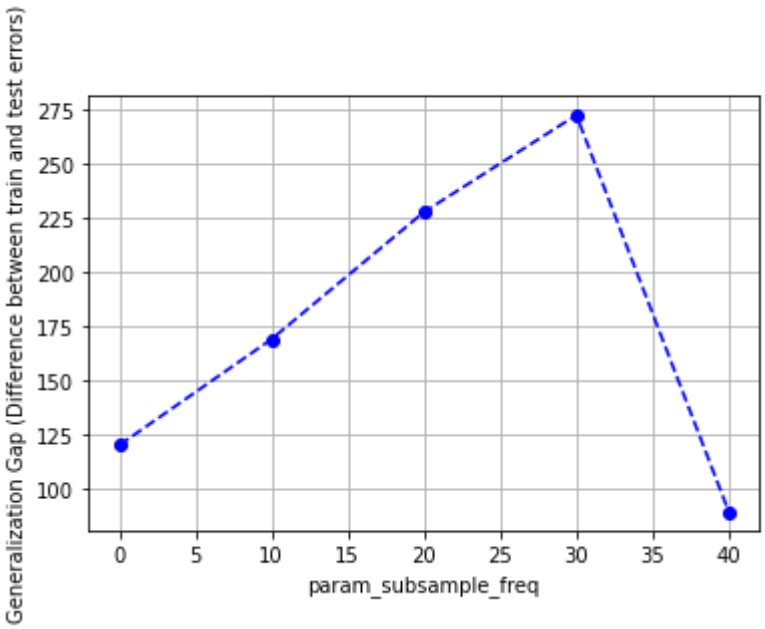

Train error
Test error
Average RMSE for 10 folds
param_reg_lambda


Generalization Gap (Difference between train and test errors)
param_reg_lambda


Train error
Test error
Average RMSE for 10 folds
param_subsample



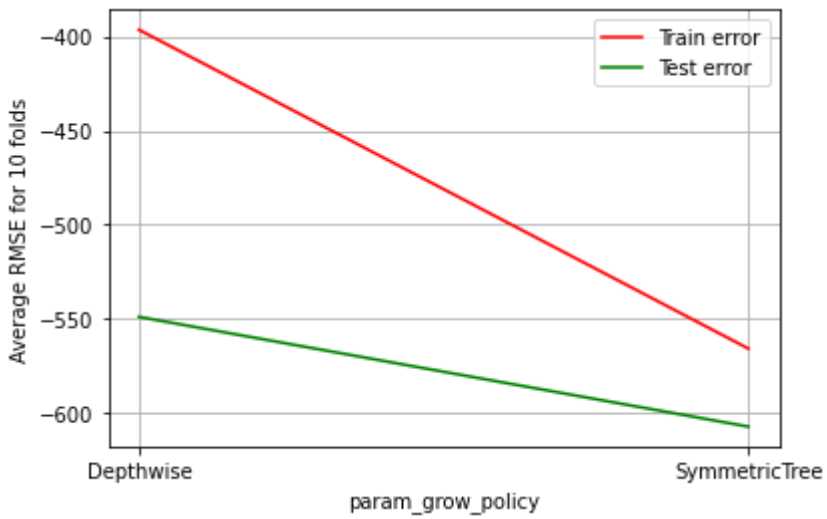


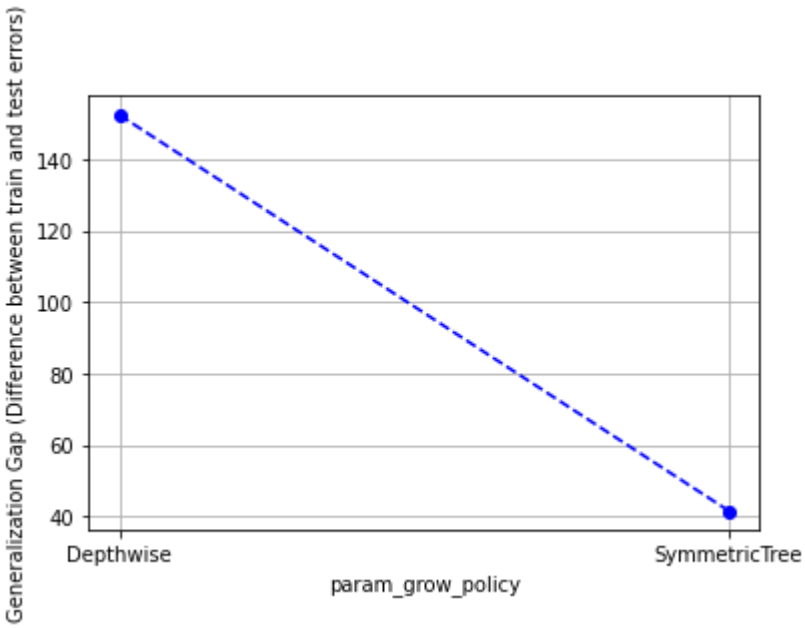


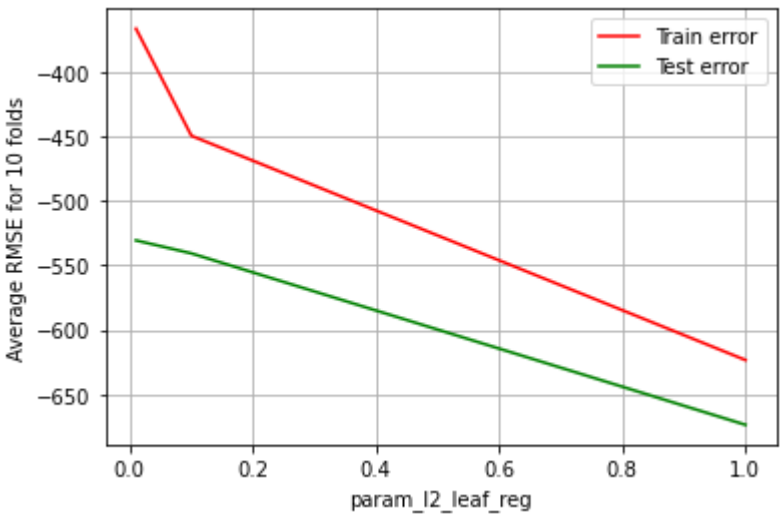
Analysing the effects of parameters of CatBoost Regressor

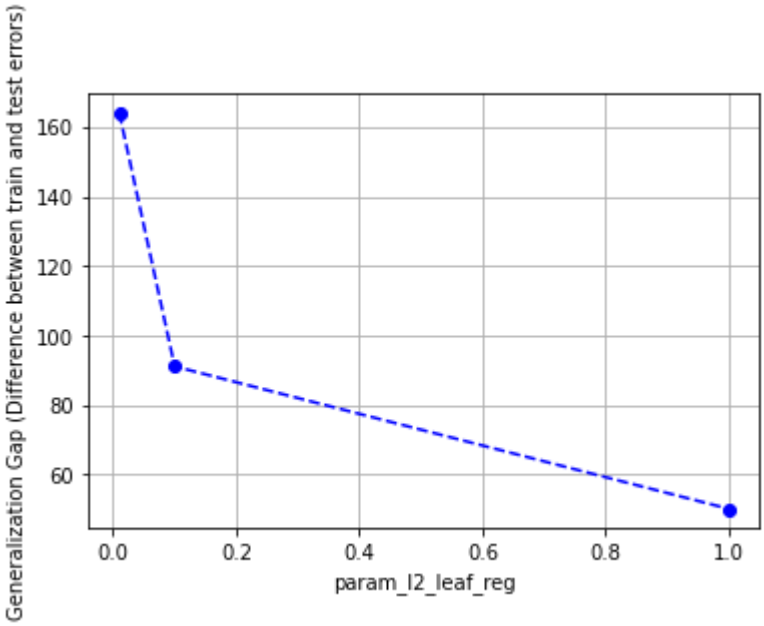
In [222]:

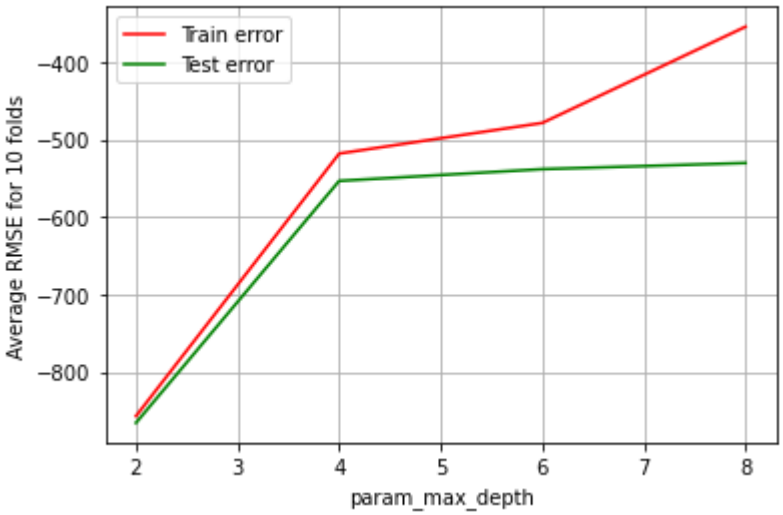
```
catParams = ['param_grow_policy', 'param_l2_leaf_reg',
       'param_max_depth', 'param_num_trees', 'param_score_function']

i = 1
for param in catParams:
    values = np.sort(catBoostResults[param].unique().tolist())

    mean_train_error = []
    mean_test_error = []

    for val in values:
        mean_train_error.append(catBoostResults[catBoostResults[param] == val]['mean
        mean_test_error.append(catBoostResults[catBoostResults[param] == val]['mean_

    difference = np.abs(np.array(mean_train_error) - np.array(mean_test_error))

    plt.figure(i)
    plt.grid()
    plt.plot(values, mean_train_error, 'r')
    plt.plot(values, mean_test_error, 'g')
    plt.xlabel(str(param))
    plt.ylabel("Average RMSE for 10 folds")
    plt.legend(['Train error', 'Test error'])
    i+=1
    plt.show()

    plt.figure(i)
    plt.grid()
    plt.plot(values, difference, 'b--', marker='o')
    plt.ylabel("Generalization Gap (Difference between train and test errors)")
    plt.xlabel(str(param))
    plt.show()
    i+=1
```

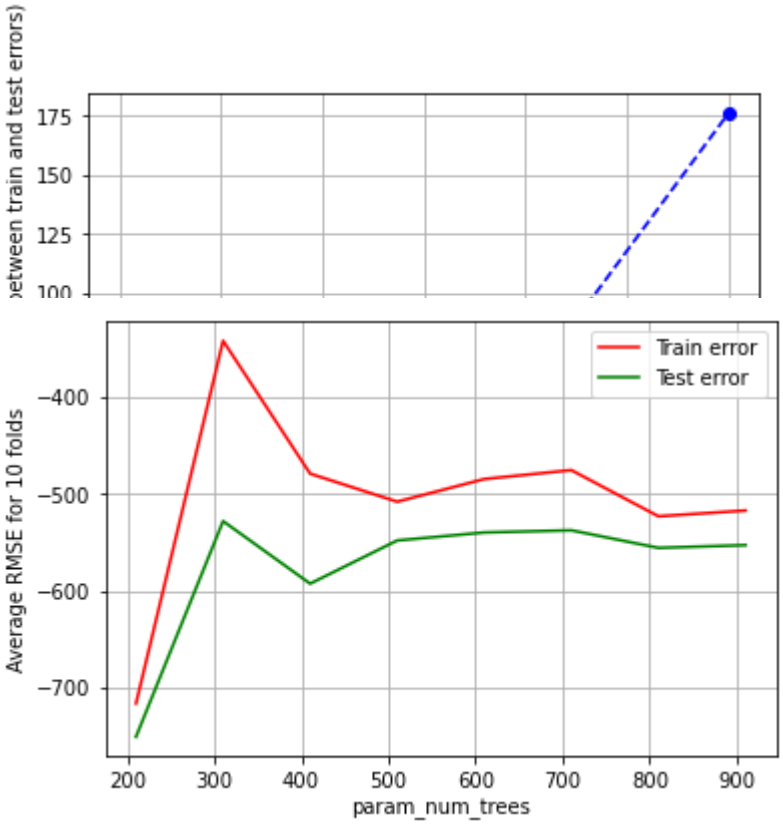


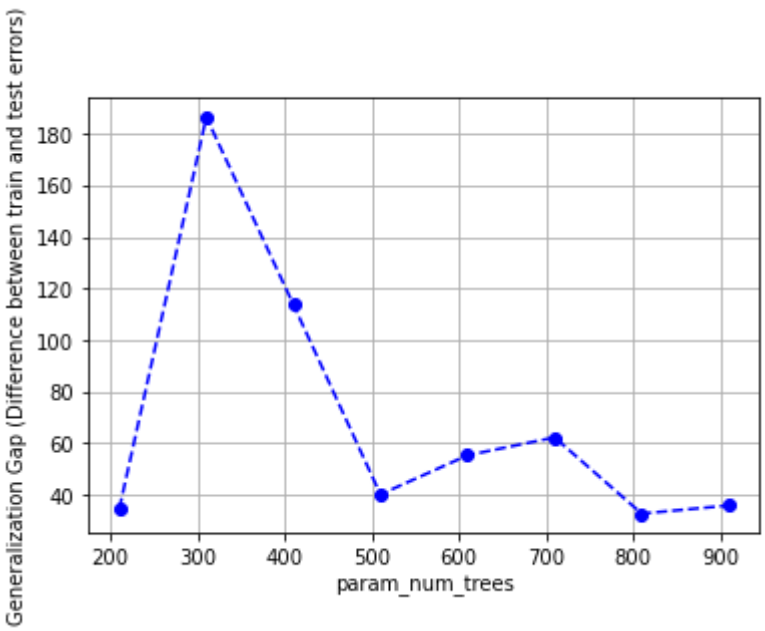

Generalization Gap (Difference between train and test errors)
140
120
100
80
60
40
Depthwise
SymmetricTree
param_grow_policy

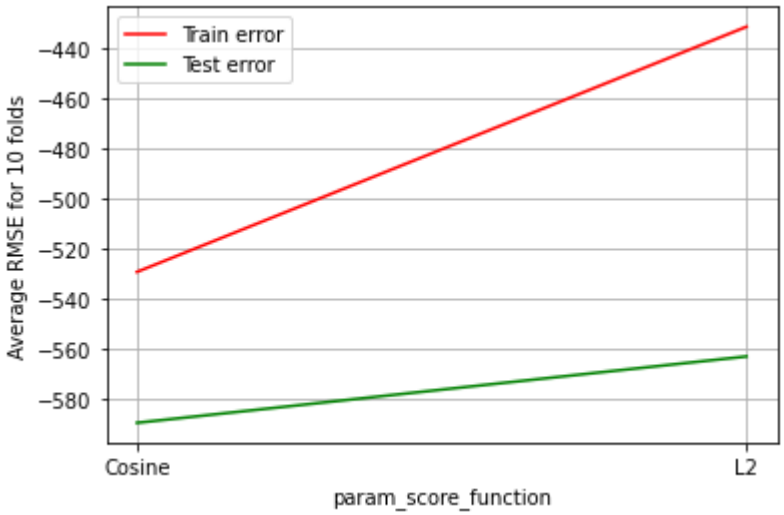

Train error
Test error
Average RMSE for 10 folds
−400
−450
−500
−550
−600
−650
0.0
0.2
0.4
0.6
0.8
1.0
param_l2_leaf_reg

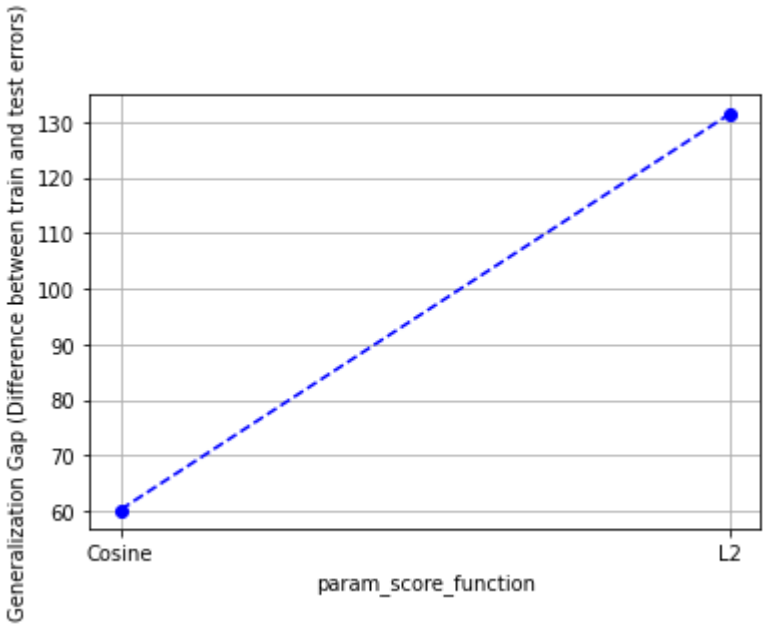

Generalization Gap (Difference between train and test errors)
160
140
120
100
80
60
0.0
0.2
0.4
0.6
0.8
1.0
param_l2_leaf_reg


Train error
Test error
Average RMSE for 10 folds
−400
−500
−600
−700
−800
2
3
4
5
6
7
8
param_max_depth


etween train and test errors)
175
150
125
100
Train error
Test error
−400
−500
−600
−700
Average RMSE for 10 folds
200
300
400
500
600
700
800
900
param_num_trees


Generalization Gap (Difference between train and test errors)
180
160
140
120
100
80
60
40
200
300
400
500
600
700
800
900
param_num_trees





Answer 24

The effects of hyperparameters for LightGBM and CatBoost on the average train and validation errors over K folds are shown above. Also difference between the two is plotted to measure the generalization gap. Lesser the gap the better that hyperparameter helps in generalization.

For LightGBM: All results are based on analysis of graph plotted above.

BoostingType:
GBDT has the most generalization gap, RF helps in more generalization than DART as seen from the graph.

Max depth:
As max depth is increased, train error is decreasing but test error is not increasing with same rate. Also the generalization gap is increasing. This shows with max depth increase overfitting occurs.

Min split gain:
We see that as the min split gain value is increased both the train and test errors are decreasing but the generalization gap is increasing as well. This can be associated to underfitting.

number of trees:
Increasing number of trees doesnt have much effect on the generalization beyond a certain point. It helps in regularization though.

Number of leaves:
Increasing the number of leaves beyond a certain point doesnt change much in the training and testing errors. It helps a bit in regularization.

alpha (L1):
we observe that both train and test RMSE initially improves with finite values of Lasso regularization, but performance drops when aggressive regularization is performed. Too low values is same as no regularization and too high value leads to underfitting

Lmbda (L2):
we observe that both train and test RMSE initially improves with finite values of Tikhonov regularization, but performance drops when aggressive regularization is performed. Too low values is same as no regularization and too high value leads to underfitting

Subsampling ratio:
If ratio is too high the generalization gap increases. Thus one has to find the optimal point since too low values lead to underfitting and too high values leads to overfitting. With increase in ratio, the train error keeps on decreasing but generalization gap is increasing.

subsample_freq:
we see that as subsampling is conducted more often, the performance drops. It attains its optimum at low value (10).

For CatBoost: All results are based on analysis of graph plotted above.

Grow Policy:
Depthwise policy has lower train and test errors compared to symmetric tree.

L2 leaf regularization:
As this increases too much, the performance drops as seen. This is because of underfitting.

Max depth:
With increase in max depth, the train error decreases, test error decrease in beginning but then it saturates. The generalization performance decreases with more depth increase possibly due to overfitting.

Number of trees:
High number of trees dont cause overfitting and also as seen help in reducing train errors and testing errors. Just that the computation time increases. Generalization improves as well.

Param Score function:
L2 reduces the train and test errors but the generalization is poor compared to Cosine which has high train and test errors but good generalization.

Answer 25

The results for this question are already provided above along with each regression model. 10 fold cross validation is applied for evaluation.

Consolidating the results here:

For Diamond Dataset :

Linear Regression:

Linear Regression Ordinary LS train time error: 0.008420395851135253
Linear Regression Ordinary LS validation error: -1220.4102585663093

Best Regularization Scheme:
Validation RMSE: -1220.0209647539734
Train RMSE: -1219.751337929666

Best Linear Regression Model on testset:
Mean squared Error for Diamonds dataset Scaled: 1228.5647060894441

Neural Network:

Best NN validation score for Diamonds dataset: -674.3019330168225
Best NN train score for Diamonds dataset: -575.1597739765449

Best NN on testset:
Mean NN squared Error for Diamonds dataset: 596.5803439579738

Random Forest Regressor:
Best Random Forest test score for Diamonds dataset: -587.6956607066389
Best Random Forest train score for Diamonds dataset: -518.1760164960453

Best RF on testset:
Mean Random Forest squared Error for Diamonds dataset: 580.9118090841522

LightGBM:
Validation RMSE: -540.7715904210484
Train RMSE: -1071.0585848019841

CatBoost:

Validation RMSE: -527.3136022252144
Train RMSE: -958.5509871349407

For emission Dataset:

Linear Regression:
Linear Regression Ordinary LS train time error: 0.0027782440185546873
Linear Regression Ordinary LS validation error: -1.4314262802990454

Best Regularized Regression:
Validation RMSE: -1.4314248070422952
Train RMSE: -1.4413075208468773

Best Linear Regression on testset:
Mean squared Error for Emissions dataset Scaled: 1.5187751212181595

Neural Network:
Best NN validation score for emissions dataset: -1.0858278843127869
Best NN train score for emissions dataset: -0.9199761645723751

Best NN on testset:
Mean NN squared Error for Emissions dataset: 1.1254830951737709

Random Forest Regressor:
Best Random Forest test score for emission dataset: -1.1476879503125526
Best Random Forest train score for emission dataset: -0.9499263129891538

Best RF on testset:
Mean Random Forest squared Error for Emissions dataset: 1.1615851541664983

We see that the Train absolute RMSE values are lower as compared to absolute RMSE values in validation set. This is because complex models generally overfit on the training set as they have seen the actual values corresponding to that and they try to minimize the loss related to training set. When tested on a new data from same distribution a model usually will have higher errors due to some unseen characteristics in the test data.

Answer 26

Out of the bag errors for Best Random forest regressor:

OOB Score for Best RF Regressor for Diamonds dataset: 0.978412855846239
OOB Score for Best RF Regressor for Emissions dataset: 0.7295101756380225

The OOB score is computed as the number of correctly predicted rows from the out-of-bag sample. OOB error is the mean prediction error on each training sample, using only the trees that did not have that sample in their bootstrap sample.Since OOB is being calculated on unseen test data, it serves as a form of validation score for the ensembling trees.

R^2 score is defined as coefficient of determination. It is the proportion of the variation in the target variable that is predictable from the features in the dataset. Higher R-squared values represent smaller differences between the observed data and the fitted values.

0% represents a model that does not explain any of the variation in the target variable around its mean.

100% represents a model that explains all the variation in the target variable around its mean.

$$R^2 = \frac{\text{Variance explained by model}}{\text{Total Variance}}$$

Twitter Data analysis

Project 4 - Part 2
Gaurav Singh
UID: 305353434

In [537]:

```
t pandas as pd
t numpy as np
t matplotlib.pyplot as plt
t os
t datetime
t pytz
t gc
t json
textblob import TextBlob
datetime import date

m allennlp.predictors.predictor import Predictor
ort allennlp_models.tagging


vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer


sklearn.feature_extraction.text import CountVectorizer
sklearn.feature_extraction.text import TfidfTransformer
nltk.stem import WordNetLemmatizer
nltk.stem import PorterStemmer
nltk.corpus import wordnet
nltk.corpus import stopwords
sklearn.feature_extraction import text
t nltk
t re
scipy.signal import find_peaks

summarizers import Summarizers

t geonamescache
sklearn.decomposition import TruncatedSVD
sklearn.pipeline import Pipeline
sklearn.model_selection import train_test_split

sklearn.linear_model import LogisticRegression
sklearn import svm
sklearn import metrics
sklearn.model_selection import KFold
sklearn.linear_model import LogisticRegression
sklearn.naive_bayes import GaussianNB
sklearn.model_selection import GridSearchCV
sklearn.svm import LinearSVC
sklearn.ensemble import RandomForestClassifier
sklearn.metrics import plot_confusion_matrix, f1_score, accuracy_score, precision_sc
```

In [261]:

```
summ = Summarizers('normal')

# https://github.com/salesforce/ctrl-sum
```

```
The tokenizer class you load from this checkpoint is not the same type
as the class this function is called from. It may result in unexpected
tokenization.
The tokenizer class you load from this checkpoint is 'BartTokenizer'.
The class this function is called from is 'PreTrainedTokenizerFast'.
```

Reading only relevant columns of data to conserve RAM as the files are huge

In [2]:

```
basePath = './ECE219_tweet_data/'
```

In [3]:

```
pst_tz = pytz.timezone('America/Los_Angeles')
```

In [4]:

```
def getAvgTweetsPerHour(data):
    df = pd.DataFrame(data, columns = ['timestamp'])
    grp = df.groupby(pd.Grouper(key = 'timestamp', freq = '60min'))
    return len(data)/ len(grp)

def getAvgRetweet(data):
    return np.mean(data)

def getAvgFollowers(data):
    return np.mean(data)

def getStats(filename):
    date = []
    retweets = []
    followers = []
    for line in open(filename):
        tweet = json.loads(line)
        date.append(datetime.datetime.fromtimestamp(tweet['citation_date']))
        retweets.append(tweet['metrics']['citations']['total'])
        followers.append(tweet['author']['followers'])
    return getAvgTweetsPerHour(date), getAvgFollowers(followers), getAvgRetweet(retw
```

In [5]:

```
fileGoHawks = os.path.join(basePath, 'tweets_#gohawks.txt')

first, second, third = getStats(fileGoHawks)

print("Average tweets per hour for #goHwaks: ", first)
print("Average no. of followers for user posting tweet #goHwaks: ", second)
print("Average no. of retweets per tweet #goHwaks: ", third)
```

```
Average tweets per hour for #goHwaks:  292.09326424870466
Average no. of followers for user posting tweet #goHwaks:  2217.923735
5281984
Average no. of retweets per tweet #goHwaks:  2.0132093991319877
```

In [6]:

```
fileGopatriots = os.path.join(basePath, 'tweets_#gopatriots.txt')

first, second, third = getStats(fileGopatriots)

print("Average tweets per hour for #gopatriots: ", first)
print("Average no. of followers for user posting tweet #gopatriots: ", second)
print("Average no. of retweets per tweet #gopatriots: ", third)
```

```
Average tweets per hour for #gopatriots:  40.888695652173915
Average no. of followers for user posting tweet #gopatriots:  1427.252
6051635405
Average no. of retweets per tweet #gopatriots:  1.4081919101697078
```

In [7]:

```
filenfl = os.path.join(basePath, 'tweets_#nfl.txt')

first, second, third = getStats(filenfl)

print("Average tweets per hour for #nfl: ", first)
print("Average no. of followers for user posting tweet #nfl: ", second)
print("Average no. of retweets per tweet #nfl: ", third)
```

```
Average tweets per hour for #nfl:  396.97103918228277
Average no. of followers for user posting tweet #nfl:  4662.3754452369
3
Average no. of retweets per tweet #nfl:  1.5344602655543254
```

In [8]:

```
filepatriots = os.path.join(basePath, 'tweets_#patriots.txt')

first, second, third = getStats(filepatriots)

print("Average tweets per hour for #patriots: ", first)
print("Average no. of followers for user posting tweet #patriots: ", second)
print("Average no. of retweets per tweet #patriots: ", third)
```

```
Average tweets per hour for #patriots:  750.6320272572402
Average no. of followers for user posting tweet #patriots:  3280.46356
16550277
Average no. of retweets per tweet #patriots:  1.7852871288476946
```

In [9]:

```
filesb49 = os.path.join(basePath, 'tweets_#sb49.txt')

first, second, third = getStats(filesb49)

print("Average tweets per hour for #sb49: ", first)
print("Average no. of followers for user posting tweet #sb49: ", second)
print("Average no. of retweets per tweet #sb49: ", third)
```

```
Average tweets per hour for #sb49:  1275.5557461406518
Average no. of followers for user posting tweet #sb49:  10374.16029201
9487
Average no. of retweets per tweet #sb49:  2.52713444111402
```

In [10]:

```
filesuperbowl = os.path.join(basePath, 'tweets_#superbowl.txt')

first, second, third = getStats(filesuperbowl)

print("Average tweets per hour for #superbowl: ", first)
print("Average no. of followers for user posting tweet #superbowl: ", second)
print("Average no. of retweets per tweet #superbowl: ", third)
```

```
Average tweets per hour for #superbowl:  2067.824531516184
Average no. of followers for user posting tweet #superbowl:  8814.9679
9424623
Average no. of retweets per tweet #superbowl:  2.3911895819207736
```

Answer 27

Average tweets per hour for #goHwaks: 292.09326424870466
Average no. of followers for user posting tweet #goHwaks: 2217.9237355281984
Average no. of retweets per tweet #goHwaks: 2.0132093991319877

Average tweets per hour for #gopatriots: 40.888695652173915
Average no. of followers for user posting tweet #gopatriots: 1427.2526051635405
Average no. of retweets per tweet #gopatriots: 1.4081919101697078

Average tweets per hour for #nfl: 396.97103918228277
Average no. of followers for user posting tweet #nfl: 4662.37544523693
Average no. of retweets per tweet #nfl: 1.5344602655543254

Average tweets per hour for #patriots: 750.6320272572402
Average no. of followers for user posting tweet #patriots: 3280.4635616550277
Average no. of retweets per tweet #patriots: 1.7852871288476946

Average tweets per hour for #sb49: 1275.5557461406518
Average no. of followers for user posting tweet #sb49: 10374.160292019487
Average no. of retweets per tweet #sb49: 2.52713444111402

Average tweets per hour for #superbowl: 2067.824531516184
Average no. of followers for user posting tweet #superbowl: 8814.96799424623
Average no. of retweets per tweet #superbowl: 2.3911895819207736

Tweets in hour

- Superbowl
- NFL

Answer 28

In [63]:

```
datesSB = []
datesNFL = []

for line in open(filesuperbowl):
    tweet = json.loads(line)
    datesSB.append(tweet['citation_date'])

for line in open(filenfl):
    tweet = json.loads(line)
    datesNFL.append(tweet['citation_date'])
```

In [104]:

```
datesSBFrame = pd.DataFrame(sorted(datesSB), columns = ['date'])
datesNFLFrame = pd.DataFrame(sorted(datesNFL), columns = ['date'])
```

In [105]:

```
datesSBFrame['date'] = datesSBFrame['date'].apply(lambda x: datetime.datetime.fromti
datesNFLFrame['date'] = datesNFLFrame['date'].apply(lambda x: datetime.datetime.from
```

In [109]:

```
datesSBFrame['day'] = datesSBFrame['date'].apply(lambda x: str(x).split(' ')[0].spli

datesSBFrame['hour'] = datesSBFrame['date'].apply(lambda x: str(x).split(' ')[1].spl

datesNFLFrame['day'] = datesNFLFrame['date'].apply(lambda x: str(x).split(' ')[0].sp

datesNFLFrame['hour'] = datesNFLFrame['date'].apply(lambda x: str(x).split(' ')[1].s
```

In [117]:

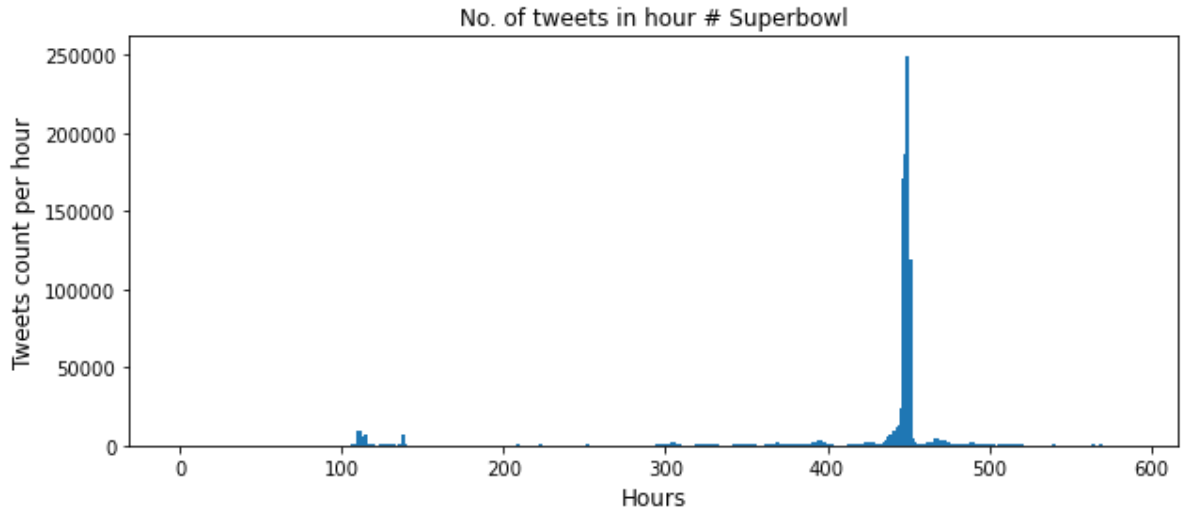
```
# For SB

days = datesSBFrame['day'].to_numpy()
hour = datesSBFrame['hour'].to_numpy()

hrCounts = []
k = 0
while True:
    if k >= len(days):
        break

    j = k

    count = 0
    currH = hour[j]
    currD = days[j]
    while j < len(days) and currH == hour[j] and currD == days[j]:
        count+= 1
        j+= 1

    k = j

    hrCounts.append(count)

    if k >= len(days):
        break
```

In [147]:

```
f = plt.figure()
f.set_figwidth(10)
f.set_figheight(4)

plt.bar(np.arange(len(hrCounts)), hrCounts, width=3)
plt.title("No. of tweets in hour # Superbowl", size = 12)
plt.xlabel("Hours", size = 12)
plt.ylabel("Tweets count per hour", size = 12)
```

Out[147]:

```
Text(0, 0.5, 'Tweets count per hour')
```



In [148]:

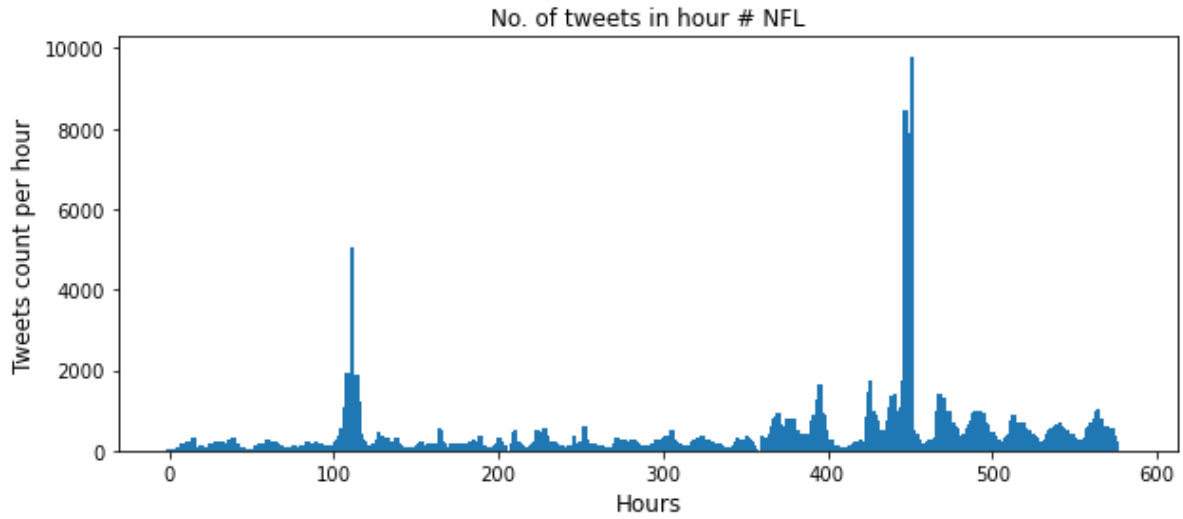
```
# For NFL

days = datesNFLFrame['day'].to_numpy()
hour = datesNFLFrame['hour'].to_numpy()

hrCounts = []
k = 0
while True:
    if k >= len(days):
        break

    j = k

    count = 0
    currH = hour[j]
    currD = days[j]
    while j < len(days) and currH == hour[j] and currD == days[j]:
        count+= 1
        j+= 1

    k = j

    hrCounts.append(count)

    if k >= len(days):
        break
```

In [149]:

```
f = plt.figure()
f.set_figwidth(10)
f.set_figheight(4)

plt.bar(np.arange(len(hrCounts)), hrCounts, width=3)
plt.title("No. of tweets in hour # NFL", size = 12)
plt.xlabel("Hours", size = 12)
plt.ylabel("Tweets count per hour", size = 12)
```

Out[149]:

```
Text(0, 0.5, 'Tweets count per hour')
```



Answer 29

With the given twitter data in hand, I have performed following tasks of interests.

1) Data exploration in terms of Number of tweets by each fanbase before, after and during the super bowl game.

2) Average sentiment trends of each fan base during the game.

3) Finding the important events during the game.

4) Built an overall summary of the game with players contributions and impacts.

5) Fanbase Prediction i.e. given a tweet predicted the state (Washington / Massachusetts) for the fan posting that tweet.

Patriots Roster

In [5]:

```
PatriotsRoster = ['Tom Brady', 'Jimmy Garoppolo', 'Shane Vereen', 'LeGarrette Blount'
                  'Jonas Gray', 'James White', 'James Develin', 'Rob Gronkowski', 'Mi
                  'Tim Wright', 'Julian Edelman', 'Brandon LaFell', 'Danny Amendola',
                  'Matthew Slater', 'Brian Tyms', 'Nate Solder', 'Sebastian Vollmer',
                  'Cameron Fleming',
                  'Dan Connolly', 'Marcus Cannon', 'Josh Fline', 'Bryan Stork', 'Ryar
                  'Rob Ninkovich',
                  'Alan Branch', 'Zach Moore', 'Joe Vellano', 'Vince Wilfork', 'Chris
                  'Jonathan Casillas', 'Jamie Collins', 'Darius Fleming', "Donta High
                  'Akeem Ayers', 'Darrelle Revis', 'Malcolm Butler', 'Brandon Browner
                  'Kyle Arrington', 'Logan Ryan',
                  'Patrick Chung', 'Devin McCourty', 'Nate Ebner', 'Duron Harmon', 'T
                  'Ryan Allen', 'Danny Aiken']

hawksRoster = ['Russell Wilson', 'Tarvaris Jackson', 'BJ Daniels', 'Marshawn Lynch',
               'Christine Michael',
              'Will Tukuafu', 'Luke Wilson', 'Tony Moeaki', 'Cooper Helfet', 'Doug Ba
               'Ricardo Lockette',
               'Chris Matthews', 'Kevin Norwood', 'Bryan Walters', 'Alvin Bailey', 'J
               'Lemuel Jeanpierre', 'Keavon Milton', 'JR Sweezy', 'James Carpenter',
               'Cliff Avril', 'Michael Bennett', 'Demarcus Dobbs', 'David King', "OBr
               'Kevin Williams', 'Tony McDaniel', 'Landon Cohen',
               'Bruce Irvin', 'KJ Wright', 'Bobby Wagner', 'Malcolm Smith', 'Mike Mor
               'Richard Sherman', 'Byron Maxwell', 'Jeremy Lane', 'DeShawn Shead', 'T
               'Earl Thomas', 'Kam Chancellor', 'Steven Terrell', 'Jeron Johnson',
               'Steven Hauschka', 'Jon Ryan', 'Clint Gresham'
              ]
```

In [6]:

```
def cleanTweet(tweet):
    return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", t

def getSentiment(tweet):
    analysis = TextBlob(tweet)
    pol = analysis.sentiment.polarity
    sub = analysis.sentiment.subjectivity
    if pol > 0:
        return 1 , pol, sub
    elif pol == 0:
        return 0, pol, sub
    else:
        return -1, pol, sub


def getSentimentVader(tweet):
    sentiment_dict = sid_obj.polarity_scores(tweet)
    if sentiment_dict['compound'] >= 0.05 :
        return 1
    elif sentiment_dict['compound'] <= - 0.05 :
        return -1
    else :
        return 0
```

In [10]:

```
sid_obj = SentimentIntensityAnalyzer()

def createData(filename):
    temp = []
    tweets = []
    regx = r'(?<![@\w])@(\w{1,25})'
    linkreg1 = r"http\S+"
    linkreg2 = r"www.\S+"

    columns = ['citation_date', 'user', 'tweet', 'sentiment', 'retweeted', 'follower
               'friends_count', 'location', 'possibly_sensitive',
              'lang', 'filter_level', 'retweets', 'ranking_score', 'impressions', 'q
               'mentions', 'source', 'hasLink']

    f = open(os.path.join(basePath, filename))
    for line in f:
        t = json.loads(line)

        temp = np.append(temp, t['citation_date'])
        temp = np.append(temp, t['tweet']['user']['id'])

        rawTweet = t['tweet']['text']
        cleaned = cleanTweet(rawTweet)

        temp = np.append(temp, cleaned)

        level, pol, sub = getSentiment(cleaned)

#         temp = np.append(temp, level)
#         temp = np.append(temp, pol)
#         temp = np.append(temp, sub)

#         senti = getSentimentVader(cleaned)
        temp = np.append(temp, level)

        temp = np.append(temp, t['tweet']['retweeted'])
        temp = np.append(temp, t['tweet']['user']['followers_count'])
        temp = np.append(temp, t['tweet']['user']['friends_count'])

        location = '0'
        if t['tweet']['user']['location']:
            location = t['tweet']['user']['location']

        temp = np.append(temp, location)


        temp = np.append(temp, t['tweet']['possibly_sensitive'])
        temp = np.append(temp, t['tweet']['lang'])
        temp = np.append(temp, t['tweet']['filter_level'])
        temp = np.append(temp, t['metrics']['citations']['total'])
        temp = np.append(temp, t['metrics']['ranking_score'])
        temp = np.append(temp, t['metrics']['impressions'])
        temp = np.append(temp, t['tweet']['geo'])

        lat = None
        long = None
        if t['tweet']['coordinates']:
            lat = str(t['tweet']['coordinates'][list(t['tweet']['coordinates'].keys(
            long = str(t['tweet']['coordinates'][list(t['tweet']['coordinates'].keys
```

```
        temp = np.append(temp, lat)
        temp = np.append(temp, long)

        menti = re.findall(regx, t['tweet']['text'])
        ment = ','.join(menti)

        temp = np.append(temp, ment)
        src = re.sub("<[^>]*>", "", t['tweet']['source'])
        temp = np.append(temp, src)

        haslink1 = re.findall(linkreg1, t['tweet']['text'])
        haslink2 = re.findall(linkreg2, t['tweet']['text'])
        hasLink = 0

        if len(haslink1) or len(haslink2):
            hasLink = 1

        temp = np.append(temp, hasLink)


        tweets.append(temp)
        temp = []

    f.close()
    return pd.DataFrame(tweets, columns=columns)
```

In [11]:

```
files = ['tweets_#gohawks.txt', 'tweets_#gopatriots.txt', 'tweets_#nfl.txt', 'tweets
        'tweets_#sb49.txt', 'tweets_#superbowl.txt']


goHawks = createData('tweets_#gohawks.txt')
print("GoHawks")
gopatriots = createData('tweets_#gopatriots.txt')
print("gopatriots")
nfl = createData('tweets_#nfl.txt')
print("nfl")
patriots = createData('tweets_#patriots.txt')
print("patriots")
sb49 = createData('tweets_#sb49.txt')
print("sb49")
superbowl = createData('tweets_#superbowl.txt')
```

```
GoHawks
gopatriots
nfl
patriots
sb49
```

Roberta sentiment analysis takes a lot of time to predict the sentiment

In [12]:

```
print(len(gopatriots))
```

```
23511
```

In [13]:

```
print(len(goHawks), len(patriots), len(nfl), len(sb49), len(superbowl))
```

```
169122 440621 233022 743649 1213813
```

In [14]:

```
# Sorting the dataframes by timestamp

goHawks['citation_date'] = pd.to_numeric(goHawks['citation_date']).astype('int')
goHawks = goHawks.sort_values(by = ['citation_date'], ignore_index=True)
goHawks['citation_date'] = goHawks['citation_date'].apply(lambda x: datetime.datetim


gopatriots['citation_date'] = pd.to_numeric(gopatriots['citation_date']).astype('int
gopatriots = gopatriots.sort_values(by = ['citation_date'], ignore_index=True)
gopatriots['citation_date'] = gopatriots['citation_date'].apply(lambda x: datetime.c


nfl['citation_date'] = pd.to_numeric(nfl['citation_date']).astype('int')
nfl = nfl.sort_values(by = ['citation_date'], ignore_index=True)
nfl['citation_date'] = nfl['citation_date'].apply(lambda x: datetime.datetime.fromti

patriots['citation_date'] = pd.to_numeric(patriots['citation_date']).astype('int')
patriots = patriots.sort_values(by = ['citation_date'], ignore_index=True)
patriots['citation_date'] = patriots['citation_date'].apply(lambda x: datetime.datet

sb49['citation_date'] = pd.to_numeric(sb49['citation_date']).astype('int')
sb49 = sb49.sort_values(by = ['citation_date'], ignore_index=True)
sb49['citation_date'] = sb49['citation_date'].apply(lambda x: datetime.datetime.from

superbowl['citation_date'] = pd.to_numeric(superbowl['citation_date']).astype('int')
superbowl = superbowl.sort_values(by = ['citation_date'], ignore_index=True)
superbowl['citation_date'] = superbowl['citation_date'].apply(lambda x: datetime.dat
```

In [22]:

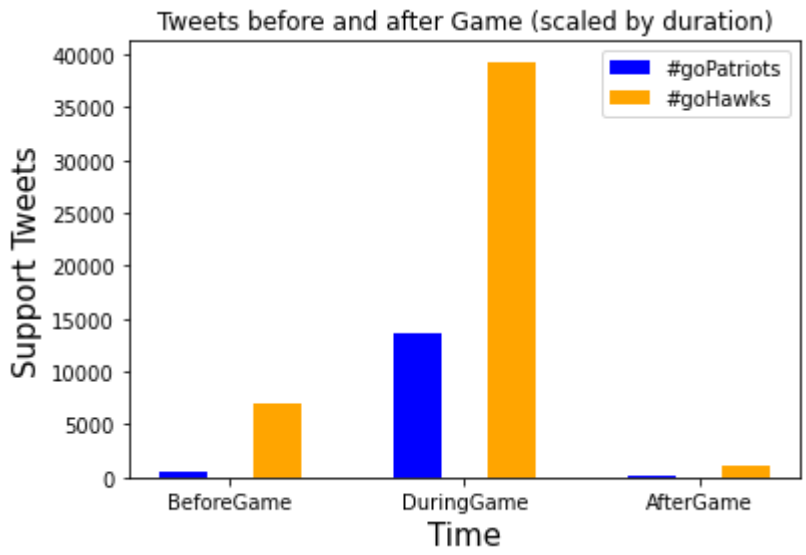
```
def getDataSplits(data, kickOffTime, endTime):
    firstTweet = data.head(1)['citation_date']
    lastTweet = data.tail(1)['citation_date']

    before = int(str(datetime.datetime.fromisoformat(kickOffTime) - firstTweet[0]).s
    after = int(str(lastTweet[len(data) - 1] - datetime.datetime.fromisoformat(endTi

    inGame = data[(data['citation_date'] >= kickOffTime) & (data['citation_date'] <=

    beforeGame = data[(data['citation_date'] < kickOffTime)]

    afterGame = data[(data['citation_date'] > endTime)]

    return inGame, beforeGame, afterGame, before, after
```

In [23]:

```
kickOffTime = '2015-02-01 15:00:00-08:00'
endTime = '2015-02-01 20:00:00-08:00'

inGamegopatriots, beforeGamegopatriots, afterGamegopatriots, b, a = getDataSplits(go
inGamegohawks, beforeGamegohawks, afterGamegohawks, bb, aa = getDataSplits(goHawks,
inGamepatriots, beforeGamepatriots, afterGamepatriots,_,_ = getDataSplits(patriots,
inGamenfl, beforeGamenfl, afterGamenfl,_,_ = getDataSplits(nfl, kickOffTime, endTime
inGamesb49, beforeGamesb49, afterGamesb49,_,_ = getDataSplits(sb49, kickOffTime, end
inGameSB, beforeGameSB, afterGameSB,_,_ = getDataSplits(superbowl, kickOffTime, endT
```

In [24]:

```
# Comparison of no. of tweets from fanBases

labels = ['BeforeGame', 'DuringGame', 'AfterGame']

width = 0.2
x = np.arange(0, 3)
plt.bar(x - 0.2, [len(beforeGamegopatriots)/ b ,
                  len(inGamegopatriots), len(afterGamegopatriots)/a], width, color='
plt.bar(x + 0.2, [len(beforeGamegohawks)/ bb,
                  len(inGamegohawks), len(afterGamegohawks)/ aa], width, color='orar

plt.xticks(x, labels)
plt.xlabel("Time", size = 15)
plt.ylabel("Support Tweets", size = 15)
plt.legend(["#goPatriots", "#goHawks"])
plt.title("Tweets before and after Game (scaled by duration)")
plt.show()
```



Part 1: Fanbase volume estimation

For this task I was interested in knowing which team had better fanbase in terms of volume.

For this two files were read:

tweets_#gohawks.txt and tweets_#gopatriots.txt

These files contained tweets which were sent in direct support of respective teams.

The files are quite big and json storage takes a lot of space, thus the files were read one line at a time and only the important fields were processed and stored in corresponding dataframes.

We know that super bowl games started at **'2015-02-01 15:00:00-08:00' and ended at '2015-02-01 20:00:00-08:00'**

While the data is being loaded into dataframes, each tweet is processed by removing the hyperlinks, web artifacts, special characters.

After getting the dataframes, the whole data was divided into three parts.

a). Tweets before game i.e all tweets before '2015-02-01 15:00:00-08:00'
b). Tweets during the game i.e all tweets in beteen '2015-02-01 15:00:00-08:00' and '2015-03-01 20:00:00-08:00'
c). Tweets after the game i.e all tweets after '2015-02-01 20:00:00-08:00'

The timestamps were convereted from UNIX timestamps to PST datetime format for convenience of interpretation.

The counts of tweets during the above three time periods are plotted in graph above and the (a) and (b) subgroups are scaled by the number of total days over which those tweets were received for comparison.

Its evident that on game day the total number of tweets surpassed the before game and after game tweets by a huge margin.

Also from the graph we can see that Patriots had a smaller fanbase compared to the hawks fanbase.

In [25]:

```
def getPlayerTweets(playerName, data):
    tweets = []
    time = []
    senti = []

    for index, row in data.iterrows():
        tweet = row['tweet']
        tm = row['citation_date']
        sentiment = row['sentiment']

        name = playerName.split(' ')
        a = (' ' + name[0] + ' ').lower()
        b = (' ' + name[0] + ' ').lower()
        c = (' ' + name[1] + ' ').lower()
        d = (' ' + name[1] + ' ').lower()
        e = (name[0]+name[1]).lower()
        if (playerName.lower() in tweet) or (e in tweet):
            tweets.append(tweet)
            time.append(tm)
            senti.append(sentiment)
        else:
            if (a in tweet) or (b in tweet):
                tweets.append(tweet)
                time.append(tm)
                senti.append(sentiment)
            elif (c in tweet) or (d in tweet):
                tweets.append(tweet)
                time.append(tm)
                senti.append(sentiment)

    df = pd.DataFrame(
            {'citation_date': time,
             'tweets': tweets,
             'sentiment': senti
            })

    return df
```

Vader sentiment analyser and textblob are not working well for this data given that support messages are kept neutral and sports terminologies and sentiments are not understood by it.

Finding key points during game

In [26]:

```
pos = ['great', 'goat', 'touchdown', 'love', 'tackle', 'goal', 'amazing', 'score', '
       'beat', 'yes', 'excited', 'rock', 'nice', 'well', 'play', 'strike', 'good', 'h
        'enjoy', 'wow', 'beastmode', 'history', 'mvp', 'pass']

neg = ['loser', 'lost', 'defeat', 'bad', 'suck', 'boo', 'need', 'damn', 'lose', 'fai
       'poor', 'worst', 'angry', 'no', 'please', 'help', 'mad', 'cry', 'shit', 'fuck'
```

In [27]:

```
lemmatizer = WordNetLemmatizer()
ps = PorterStemmer()
```

In [430]:

```
stopWords = set.union(set(stopwords.words('English')), set(text.ENGLISH_STOP_WORDS))

def get_pos_tags(nltkTag):
    firstChar = nltkTag[0]
    if firstChar == 'J':
        return wordnet.ADJ
    if firstChar == 'S':
        return wordnet.ADJ_SAT
    if firstChar == 'V':
        return wordnet.VERB
    if firstChar == 'N':
        return wordnet.NOUN
    if firstChar == 'R':
        return wordnet.ADV
    return wordnet.NOUN


def lemmatize(text, stopWords=stopWords):
    tokens = nltk.word_tokenize(text)

    temp = [token for token in tokens if token not in stopWords]

    tokens = temp

    posTags = nltk.pos_tag(tokens)
    tags = [get_pos_tags(w[1]) for w in posTags]
    lemmas = [lemmatizer.lemmatize(tokens[i], tags[i]) for i in range(0, len(tokens)
    lemmas = ' '.join(lemmas)
    lemmas = re.sub(r'\w*\d\w*', '', lemmas)
    lemmas = re.sub('[.]', '', lemmas)
    lemmas = re.sub(' +', ' ', lemmas).strip()
    return lemmas

def lemmatize2(text, stopWords=stopWords):
    tokens = nltk.word_tokenize(text)

    temp = [token for token in tokens if token not in stopWords and not token.isdigi

    tokens = temp

    posTags = nltk.pos_tag(tokens)
    tags = [get_pos_tags(w[1]) for w in posTags]
    lemmas = [lemmatizer.lemmatize(tokens[i], tags[i]) for i in range(0, len(tokens)
    lemmas = ' '.join(lemmas)
    lemmas = re.sub(r'\w*\d\w*', '', lemmas)
    lemmas = re.sub('[.]', '', lemmas)
    lemmas = re.sub(' +', ' ', lemmas).strip()
    return lemmas


def preprocessNC(sample):
    sample = sample.split('.')
    sample = [lemmatize(sentence) for sentence in sample]
    sample = '.'.join(sample)
    return sample

def preprocess(sample):
    sample = sample.split('.')
    sample = [lemmatize2(sentence) for sentence in sample]
```

```
    sample = '.'.join(sample)
    return sample

def vectorizer(min_df = 3):
    return CountVectorizer(preprocessor=preprocessNC, stop_words='english', min_df=m

def vectorizer2(min_df = 3):
    return CountVectorizer(preprocessor=preprocess, stop_words='english', min_df=mir

def get_sentiment(data, min_df=3):
    CV = vectorizer(min_df=min_df)

    try:
        counts = CV.fit_transform(data).toarray()

        word_dict = dict(enumerate(CV.get_feature_names_out().flatten(), 0))
        word_dict = dict((v, k) for k, v in word_dict.items())
        scores = []
        for i in range(len(data)):
            p = 0
            n = 0
            for pw in pos:
                if pw in word_dict:
                    p+= counts[i][word_dict[pw]]
            for ng in neg:
                if ng in word_dict:
                    n+= counts[i][word_dict[ng]]
            scores.append(np.sign(p - n))

        return np.mean(scores)
    except:
        return 0
```

Taking only english tweets.

In [29]:

```
inGamegopatriots = inGamegopatriots[inGamegopatriots['lang'] == 'en']
inGamegohawks = inGamegohawks[inGamegohawks['lang'] == 'en']
inGamegopatriots['tweet'] = inGamegopatriots['tweet'].apply(lambda x: x.lower())
inGamegohawks['tweet'] = inGamegohawks['tweet'].apply(lambda x: x.lower())
inGamegopatriots['sentiment'] = inGamegopatriots['sentiment'].astype('int')
inGamegohawks['sentiment'] = inGamegohawks['sentiment'].astype('int')

inGamepatriots = inGamepatriots[inGamepatriots['lang'] == 'en']
inGamenfl = inGamenfl[inGamenfl['lang'] == 'en']
inGamepatriots['tweet'] = inGamepatriots['tweet'].apply(lambda x: x.lower())
inGamenfl['tweet'] = inGamenfl['tweet'].apply(lambda x: x.lower())
inGamepatriots['sentiment'] = inGamepatriots['sentiment'].astype('int')
inGamenfl['sentiment'] = inGamenfl['sentiment'].astype('int')

inGamesb49 = inGamesb49[inGamesb49['lang'] == 'en']
inGameSB = inGameSB[inGameSB['lang'] == 'en']
inGamesb49['tweet'] = inGamesb49['tweet'].apply(lambda x: x.lower())
inGameSB['tweet'] = inGameSB['tweet'].apply(lambda x: x.lower())
inGamesb49['sentiment'] = inGamesb49['sentiment'].astype('int')
inGameSB['sentiment'] = inGameSB['sentiment'].astype('int')
```

Using custom sentiment scores from CountVectorizer

Using custom sentiment scores from CountVectorizer

In [30]:

```
grp1 = inGamegopatriots.groupby(pd.Grouper(key = 'citation_date',
                                           freq = '3min')).apply(lambda x: get_senti
```

In [31]:

```
grp2 = inGamegohawks.groupby(pd.Grouper(key = 'citation_date',
                                        freq = '3min')).apply(lambda x: get_sentimer
```

In [32]:

```
duration = ((datetime.datetime.fromisoformat(kickOffTime) - datetime.datetime.fromis
```

In [35]:

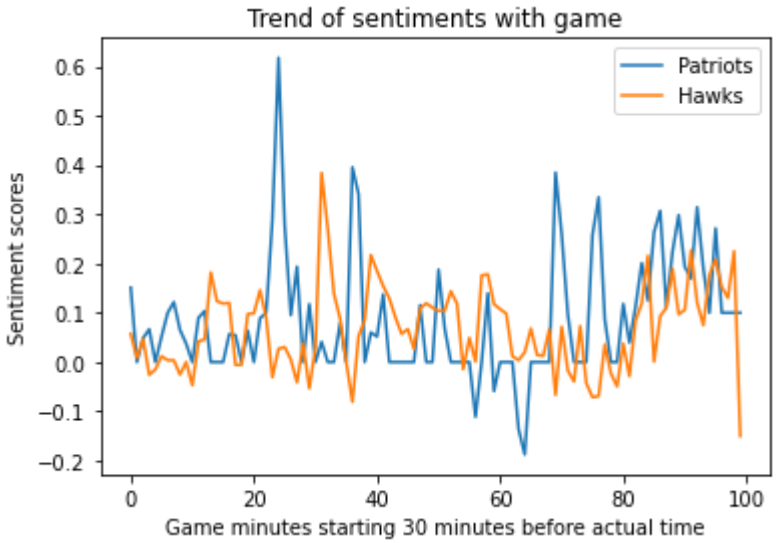
```
a = grp1.values
b = grp2.values
```

In [36]:

```
plt.plot(a, label = 'Patriots')
plt.plot(b, label = 'Hawks')
plt.legend()
plt.title("Trend of sentiments with game")
plt.xlabel("Game minutes starting 30 minutes before actual time")
plt.ylabel("Sentiment scores")
```

Out[36]:

```
Text(0, 0.5, 'Sentiment scores')
```



In [37]:

```
grp1 = inGamegopatriots.groupby(pd.Grouper(key = 'citation_date', freq = '3min')).me
grp2 = inGamegohawks.groupby(pd.Grouper(key = 'citation_date', freq = '3min')).mean(
```

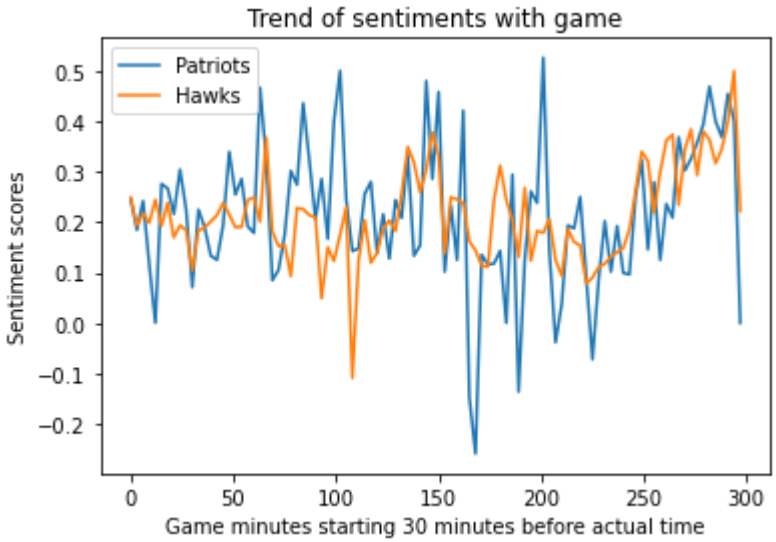
Using TextBlob as sentiment analyser

In [38]:

```
plt.plot(np.arange(0, 300, 3), grp1['sentiment'], label='Patriots')
plt.plot(np.arange(0, 300, 3), grp2['sentiment'], label='Hawks')
plt.legend()
plt.title("Trend of sentiments with game")
plt.xlabel("Game minutes starting 30 minutes before actual time")
plt.ylabel("Sentiment scores")
```

Out[38]:

```
Text(0, 0.5, 'Sentiment scores')
```



Part 2: Average sentiment trends

Sentiments of the fans inferred from the tweets tell a lot about the game and timeline. When there is a sudden drop in the sentiment it means that fans are disappointed due to some event or bad player performance and vice versa.

To analyse the sentiments of the fans, I used two methods one is using TextBlob and other is using a custom built sentiment predictor.

For the custom built sentiment analyser, I creaeted two groups of words one which is a set of all positive words and other is the set of all negative words. Soccer specific and sports specific words and slangs are also included for better performance. The cleaned in game tweets are grouped in timeperiods of 3 mins. The grouped tweets are then passed through CountVetorizer to get the counts of words in each tweet. Then based on how many positive and negative words are contained by each tweet we calculated the sentiment of that sentence as the mean of difference between the positive and negative counts of wors present in that group.

TextBlob is a Lexicon-based sentiment analyzer It has some predefined rules or we can say word and weight dictionary, where it has some scores that help to calculate a sentence's polarity. It works on some predefined rules.Clearly it is not a good analyser as it is not able to give good results on some benchmarks where it achieves only 0.55 accuracy. It is not able to pick up the sarcasms, some random negated words can easily break the rules on which it works. But we are using it to just get a comparison and given that we are averaging the sentiments over a window of 3min it works well.

The results of the sentiment analysis done for two fanbases tweets using goPatriots and goHawks are shown above using the custom sentiment analyser and using TextBlob. We can see out custom analyser is not able to pick that many peaks but overall we are seeing that Hawks dominated the sentiment battle for majority of the

game and then at the end patriots overcame this battle. And this is true because Patriots actually won the game in the end.

In [343]:

```
def getSummary(playerName, data, duration, threshold = 0.50):
    series = getPlayerTweets(playerName, data)
    grouped = series.groupby(pd.Grouper(key = 'citation_date', freq = str(duration)
    means = grouped.mean()

    if threshold > 0:
        peaks = means[means['sentiment'] >= threshold].index.to_numpy()
    else:
        peaks = means[means['sentiment'] <= threshold].index.to_numpy()

    i = 0
    tweets = []
    timeStamps = []
    while i < len(peaks):
        start = peaks[i]
        end = start + datetime.timedelta(minutes = duration)
        i+= 1
        timeStamps.append(start)
        text = series[(series['citation_date'] >= start) & (series['citation_date']
        tweets.append('. '.join(text))

    df = pd.DataFrame(
                {'citation_date': timeStamps,
                 'tweets': tweets,
                 'duration': duration
                })
    return df


def getKeyTweets(data, player):
    tweets = data.split('.')
    player = player.lower()

    text = []
    for tw in tweets:
        name = player.split(' ')
        a = (' ' + name[0] + ' ').lower()
        b = (' ' + name[0] + ' ').lower()
        c = (' ' + name[1] + ' ').lower()
        d = (' ' + name[1] + ' ').lower()
        e = (name[0]+name[1]).lower()
        if (player.lower() in tw) or (e in tw):
            text.append(tw)
        else:
            if (a in tw) or (b in tw):
                text.append(tw)
            elif (c in tw) or (d in tw):
                text.append(tw)

    if len(text) > 30:
        text = np.random.choice(text, 30, replace=False)

    summary = []
    indices = np.arange(0, len(text), 10)
    for i in range(len(indices)):
        start = indices[i]
        end = indices[i] + 10
        content = '. '.join(text[start : end])
        summary.append(summ(content))
```

```
    return '. '.join(summary)
```

Plotting the number of tweets encountered as per given duration of game

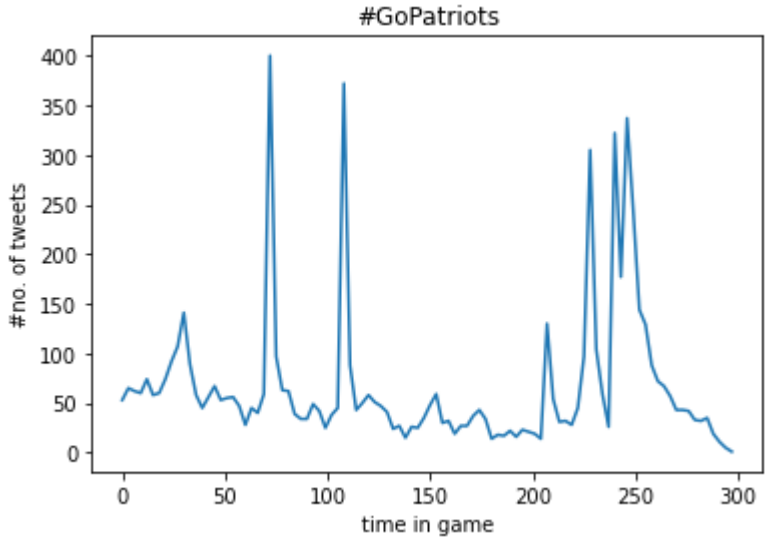
In [294]:

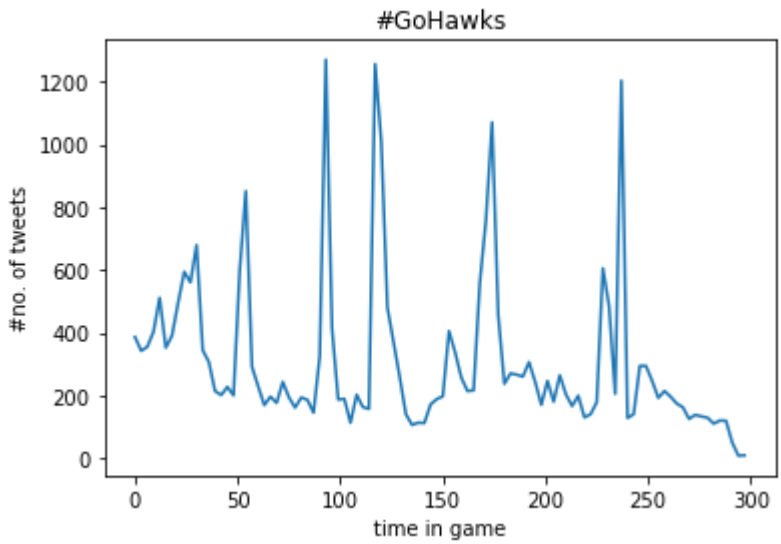
```
def plotVariations(data, duration, title, xlab, ylab):
    frequency = str(duration) + 'min'

    keys = data[['citation_date', 'location']].groupby(
        pd.Grouper(key = 'citation_date', freq = str(duration) + 'min')).count()['lo

    total = (data['citation_date'].to_numpy()[-1] - data['citation_date'].to_numpy()
    rang = (total.seconds / (60))
    x = np.arange(0, rang, duration)
    plt.plot(x, keys)
    plt.title(title)
    plt.xlabel(xlab)
    plt.ylabel(ylab)
    plt.show()

def getPeaks(data, duration, title, xlab, ylab, pThresh):
    frequency = str(duration) + 'min'

    timestamps = data[['citation_date', 'location']].groupby(
        pd.Grouper(key = 'citation_date', freq = str(duration) + 'min')).count().inc

    keys = data[['citation_date', 'location']].groupby(
        pd.Grouper(key = 'citation_date', freq = str(duration) + 'min')).count()['lo

    peaks, _ = find_peaks(keys, height=0, threshold=pThresh)

    total = (data['citation_date'].to_numpy()[-1] - data['citation_date'].to_numpy()
    rang = np.ceil(total.seconds / (60))
    x = np.arange(0, rang, duration)
    plt.plot(x, keys)
    plt.plot(peaks*duration, keys[peaks], "x", color='r')
    plt.title(title)
    plt.xlabel(xlab)
    plt.ylabel(ylab)
    plt.show()

    tweets = []
    for peak in peaks:
        ts = timestamps[peak]

        start = ts
        end = start + datetime.timedelta(minutes = duration)
        text = data[(data['citation_date'] >= start) & (data['citation_date'] <= enc
        tweets.append('. '.join(text))

    return peaks, timestamps[peaks], tweets

def getAssociatedPlayers(keyData, roster):
    keyPoints = len(keyData)

    counts = {}
    for player in roster:
        player = player.lower()
        counts[player] = 0
        name = player.split(' ')
        a = (' ' + name[0] + ' ').lower()
        b = (' ' + name[0] + ' ').lower()
        c = (' ' + name[1] + ' ').lower()
        d = (' ' + name[1] + ' ').lower()
```

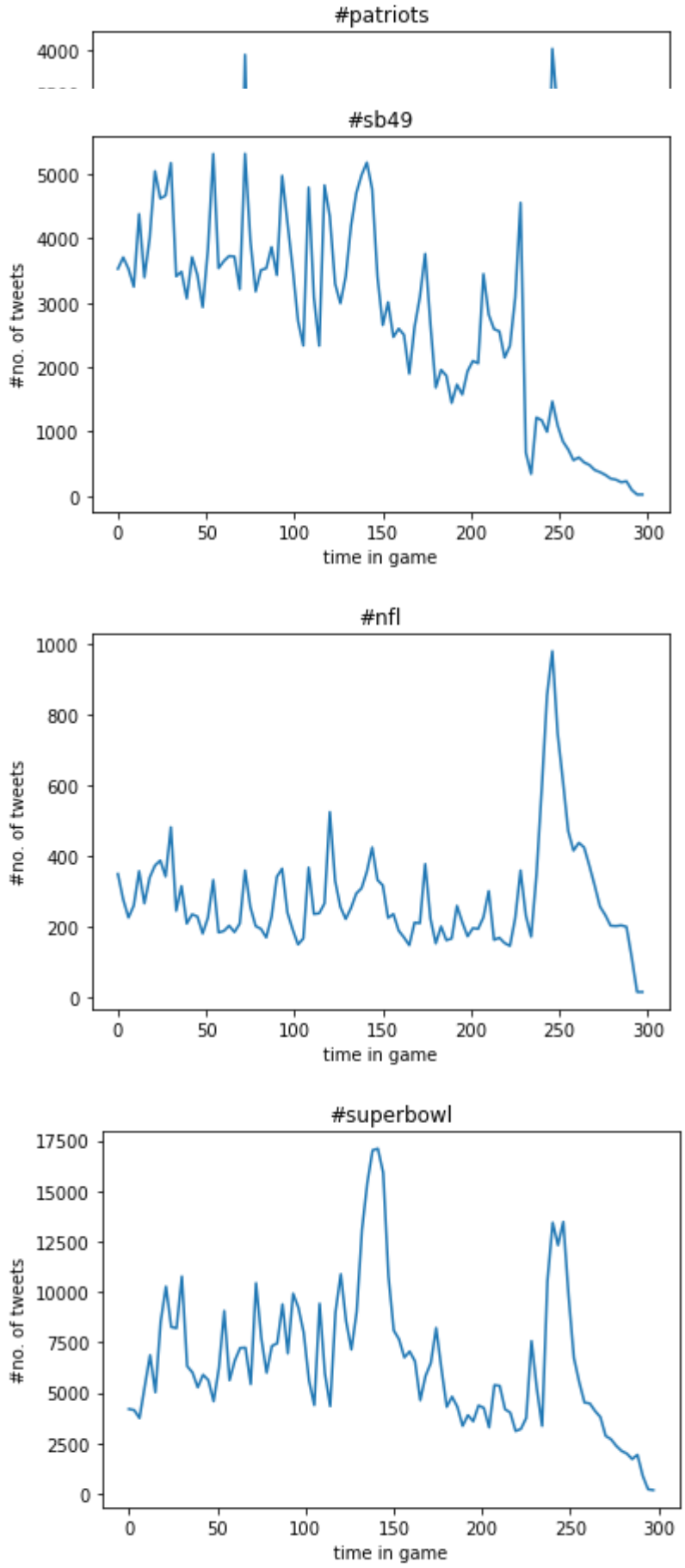
```
        e = (name[0]+name[1]).lower()

        temp = keyData
        counts[player]+= len(re.findall(player, temp))
        temp = re.sub(player, '', temp)

        counts[player]+= len(re.findall(a, temp))
        temp = re.sub(a, '', temp)

        counts[player]+= len(re.findall(b, temp))
        temp = re.sub(b, '', temp)

        counts[player]+= len(re.findall(c, temp))
        temp = re.sub(c, '', temp)

        counts[player]+= len(re.findall(d, temp))
        temp = re.sub(d, '', temp)

        counts[player]+= len(re.findall(e, temp))
        temp = re.sub(e, '', temp)

    counts = dict(sorted(counts.items(), key=lambda item: item[1], reverse=True))
    return counts
```

In [91]:

```
plotVariations(inGamegopatriots, 3, "#GoPatriots", "time in game", "#no. of tweets")

plotVariations(inGamegohawks, 3, "#GoHawks", "time in game", "#no. of tweets")

plotVariations(inGamepatriots, 3, "#patriots", "time in game", "#no. of tweets")

plotVariations(inGamesb49, 3, "#sb49", "time in game", "#no. of tweets")

plotVariations(inGamenfl, 3, "#nfl", "time in game", "#no. of tweets")

plotVariations(inGameSB, 3, "#superbowl", "time in game", "#no. of tweets")
```





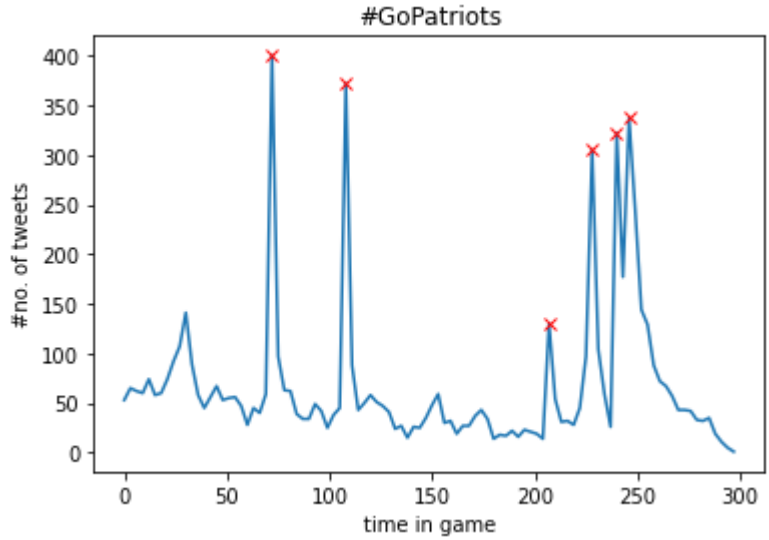


Peak Identification

For goPatriots

In [196]:

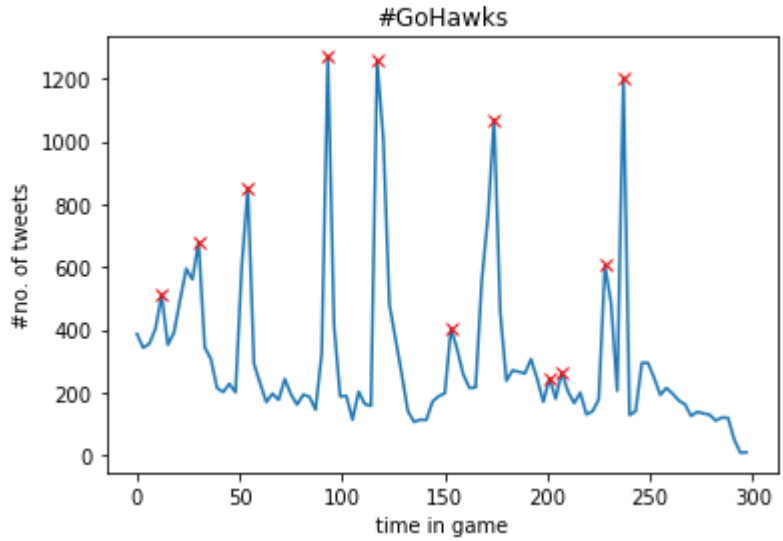
```
peaksgoPatriots, tsgopatriots, twgp = getPeaks(inGamegopatriots, 3,
                                               "#GoPatriots", "time in game", "#nc
```



For goHawks

In [197]:

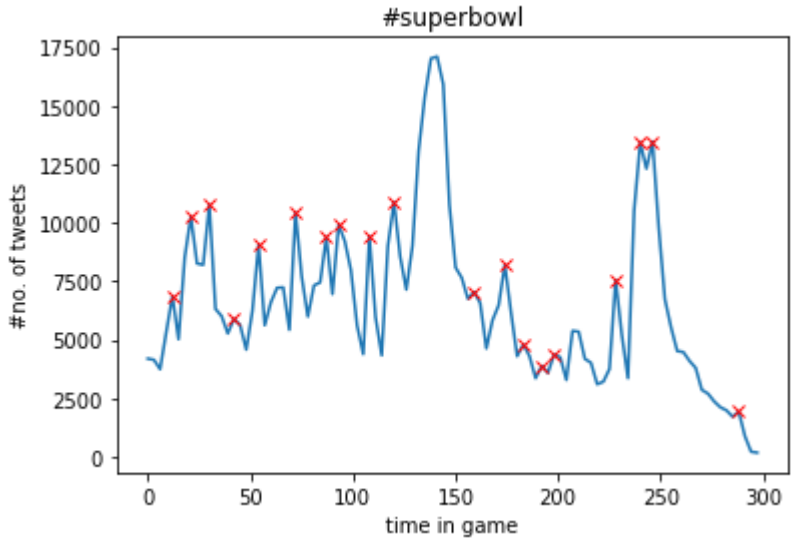
```
peaksgoHawks, tsgohawks, twgh = getPeaks(inGamegohawks, 3, "#GoHawks", "time in game
```



For superbowl

In [198]:

```
peakssb, tssb, twsb = getPeaks(inGameSB, 3, "#superbowl", "time in game", "#no. of t
```



Part 3 : Finding the important events in the game.

The number of tweets during the game gives an idea about the key events. With something interesting happening in game the number of tweets should go up. And thus for all the 6 files(hashtags) I plotted the number of tweets sent in intervals of 3min. The peaks correspond to the key events. The plots are shown above.

One observation is that the peaks of #gopatriots and #gohawks are not occuring at the same time. This means one event which is a positive outcome for one fanbase is a bad event for the other fanbase.

I focussed on three hashtage - #gopatriots, #gohawks and #superbowl. The peaks are identified using scipy's find_peaks from signal processing module. We considered the important events which saw an increase in tweets by a count of 100 compared to its neighbouring time slots. The important events are marked in red as plots above.

A total of 7 keyevents for gopatriots, 11 for gohawks and 25 for superbowl tweets are observed.

Part 4

For this part, I first collected the playing roster for both of the teams from :
https://www.sbnation.com/nfl/2015/2/1/7957703/super-bowl-rosters-2015-seahawks-patriots (https://www.sbnation.com/nfl/2015/2/1/7957703/super-bowl-rosters-2015-seahawks-patriots)

Then, for each of the three hashtags i.e gopatriots, gohawkss and superbowl, corresponding to each key event which is found above, we computed the key players which were mentioned during that event. All the events are of 3 minute duration. To get the mentions a regex on names and its variation is written. The top 5 players with most number of tweets are chosen and the tweets corresponding to that player were stored and processed.

This helps in identifying the players which were in limelight during all the key moments of the game and give us the idea of what they actually did during that time.

To analyse the tweets better, given that each player's tweets can go very high, I used a summarizer from Salesforce.

https://github.com/hyunwoongko/summarizers (https://github.com/hyunwoongko/summarizers)

The summarization is based on https://arxiv.org/abs/2012.04281 (https://arxiv.org/abs/2012.04281) paper . It uses pretrained BART (denoising autoencoder for pretraining sequence-to-sequence models) along with control tokens as inputs to get customised outputs.

The tweets corresponding to each important player in the identified key point is then shortened by randomly choosing 30 tweets which are then grouped into groups of 10 and passed to the summarizer to get a shortened summary which is then appended to the other summaries.

I did this on three hashtags so as to gain different perspectives two supportive(#gohawks, #gopatriots) and one neutral (#superbowl).

The results are quite good as we almost get the whole import events in the game with most important players involded in that key event an their contributions with scorelines and MVP results. All the results are given below.

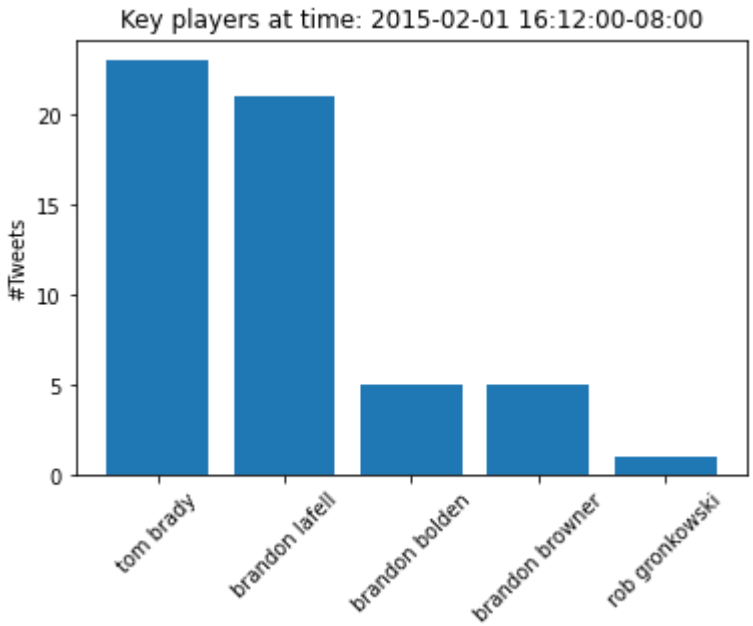
Part 4 - Getting summary of the game using peaks from Patriots fans perspective

Associated players with keypoints

- For #goPatriots

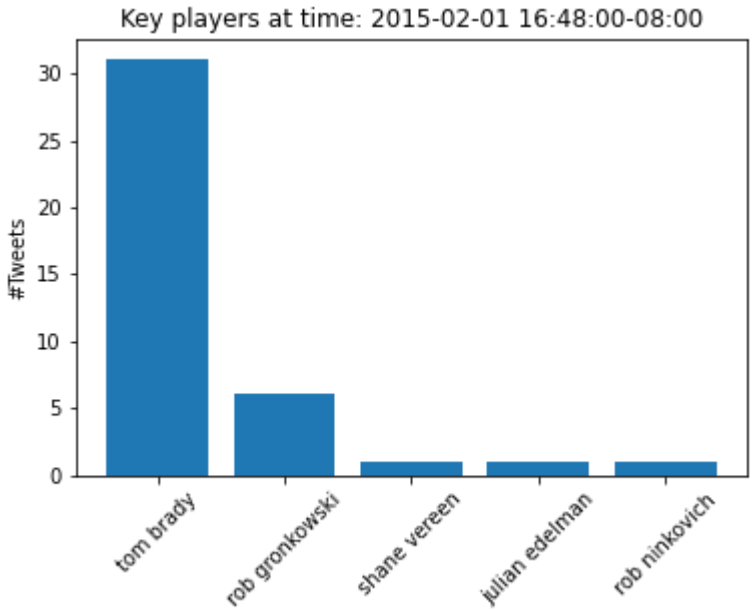
In [347]:

```
keyTimesGP = tsgopatriots
keyTimesGH = tsgohawks
keyTimesSB = tssb


for id, tw in enumerate(twgp):
    mp = getAssociatedPlayers(tw, PatriotsRoster)
    top = list(mp.keys())[0:5]

    values = []
    for tp in top:
        values.append(mp[tp])

    plt.bar(top, values)
    plt.xticks(rotation = 45)
    plt.title("Key players at time: {}". format(keyTimesGP[id]))
    plt.ylabel("#Tweets")
    plt.show()

    for tp in top:
        print("\n----- Summary of tweets for given Key moment for {} ---- \n ".forma
        print(getKeyTweets(tw, tp))
```



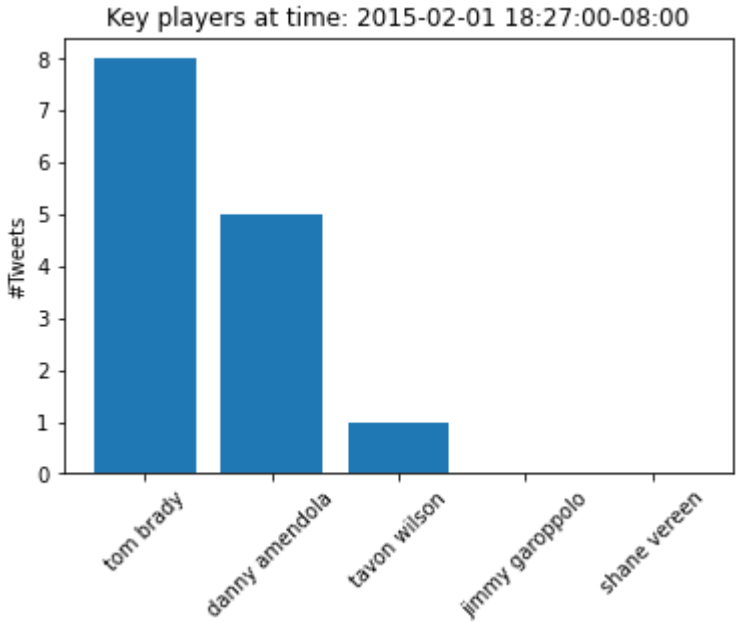

```
----- Summary of tweets for given Key moment for tom brady ----

If you aren t pulling for tom brady tonight you re blind gopatriots..
The difference between tom brady and other nfl quarterbacks is that he
throws a red zone interception and it doesn t phase him at all.. Brady
s interception was only part of his master plan gopatriots.

----- Summary of tweets for given Key moment for brandon lafell ----

touchdown go patriots my boy lafell doyourjob gopatriots.  touchdown b
randon lafell que passe do brady Gopatriots gobrady espntemsuperbowl14
9.. Whoohoo lafell sb49 gopatriots.  pats strike first lafell td 7 0 p
ats gopatriOTS superbowlxlix.. edelman, gronkowski, lafell and brady a
re gonna dominate this game.
```

```
----- Summary of tweets for given Key moment for brandon bolden ----

Gopatriots.  touchdown brandon lafell gopatriots grande tom brady sb4
9.

----- Summary of tweets for given Key moment for brandon browner ----

Gopatriots.  touchdown brandon lafell gopatriots grande tom brady sb4
9.

----- Summary of tweets for given Key moment for rob gronkowski ----

edelman, gronkowski, lafell and brady are gonna dominate this game.
```

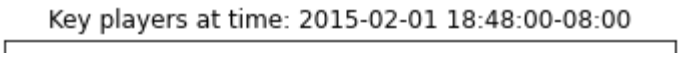



```
----- Summary of tweets for given Key moment for tom brady ----

tombrady brady 12 gopatriots usa.  brady to gronk for the td gopatrio
t.  touchdown again let s go patriots.. Tom brady s doing good gopatri
ots.  awesome pass from tom brady touchfuckingdoooown. patriots 14 7 s
eattle superbowl.. Gopatriots.  touchdown patriots tombrady assure sup
erbowl superbowlxlix.  brady is doting these seahawks up ahah great wo
rk gronk gopatriot.  1 tom brady gopatriots.

----- Summary of tweets for given Key moment for rob gronkowski ----

There he is my boy gronkowski gronkanator spike nfl sb49 superbowlxli
x.

----- Summary of tweets for given Key moment for shane vereen ----

"If you run it it vereen score us a touchdown highschoolflashbacks gop
atriots"

----- Summary of tweets for given Key moment for julian edelman ----

"Gronk, gronk, amp edelman are on fire tonight superbowlxlix gopatriot
s"

----- Summary of tweets for given Key moment for rob ninkovich ----
```

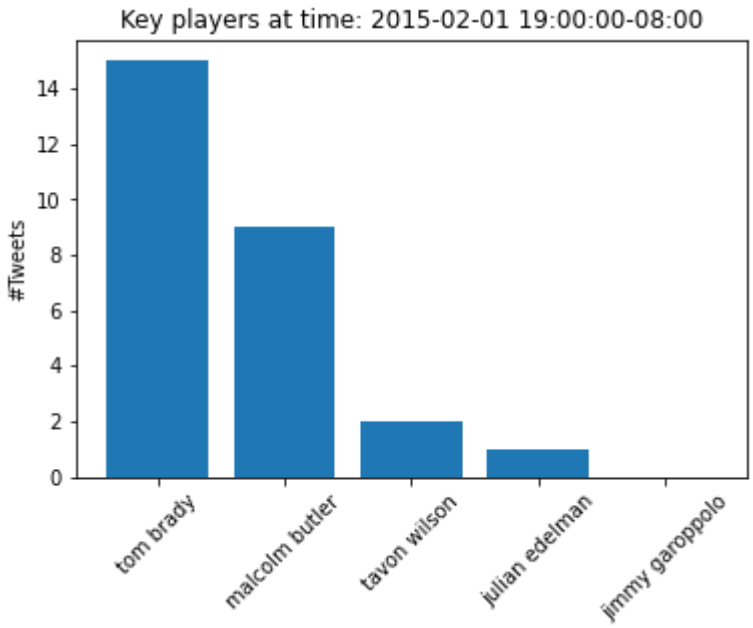
```
rob gronkowski just scored a 22 yard touchdown pass superbowlxlix gopa
triots.
```




```
----- Summary of tweets for given Key moment for tom brady ----

Tom brady gopatriots. be terrific.

----- Summary of tweets for given Key moment for danny amendola ----

Gopatriots.  at last touchdown amendola gopatriots superbowl.  thank c
hrist amendola patriots gopatriot superbowlxlix.

----- Summary of tweets for given Key moment for tavon wilson ----

c'mon pats d bury wilson on this next drive gopatriots.

----- Summary of tweets for given Key moment for jimmy garoppolo ----


----- Summary of tweets for given Key moment for shane vereen ----
```

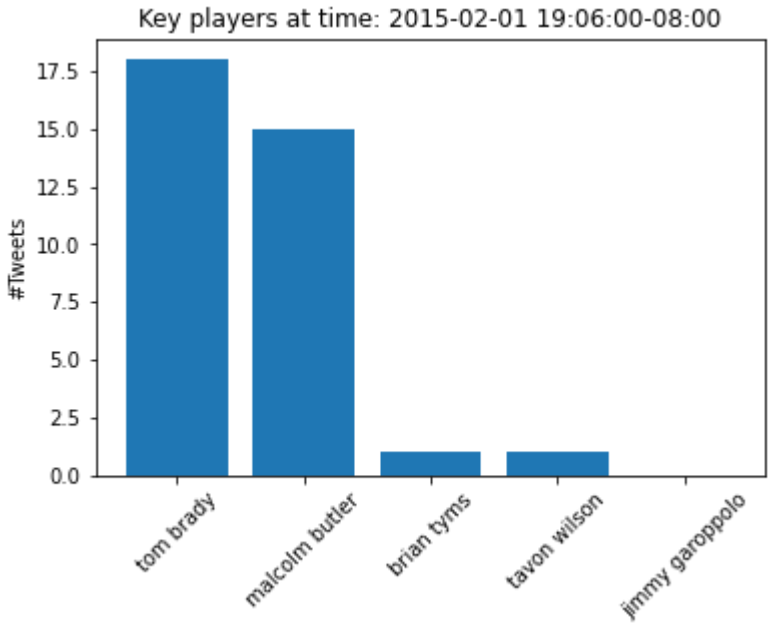



```
----- Summary of tweets for given Key moment for tom brady ----

Gopatriots.  espntemsuperbowl49 big brady boy doyourjob gopatriots gop
ats.. Brady is on top again teambrady gopatriots.  superbowlxlix. touc
hdooooown grande brady gopats gopatriot superbowl.. Sb49: "Go brady bo
y, go brady boy" "I love brady amp edelman"

----- Summary of tweets for given Key moment for julian edelman ----

Edelman scores touchdown gopatriots superbowlxlix.  yes julian with th
e touchdown what concussion gopatriot superbowl.  touchdownnnnnnn yes
brady edelman you legend come on new england you can do it 28 24 Gopat
riots teambrady.. "Brady edelman touchdown. edelman we got the lead"

----- Summary of tweets for given Key moment for tavon wilson ----

i just can t take all the cheesy russell wilson quotes.

----- Summary of tweets for given Key moment for shane vereen ----

"The touchdown gopatriots amendola edelman vereen brady"

----- Summary of tweets for given Key moment for james white ----

yeaaaaaaah gopatriots. White man could jump superbowl patriotsvsseahaw
ks.
```




```
----- Summary of tweets for given Key moment for tom brady ----

tom brady wins 4th super bowl ring gopatriots superbowlchampions. tom
brady shows respect for mvp by winning 4th ring.. There's so much scre
aming go pats finishthejob.

----- Summary of tweets for given Key moment for malcolm butler ----

Gopatriots.  butler intercepting the pass that will ultimatly lead to
the patriots winning superbowl.
```

```
----- Summary of tweets for given Key moment for tavon wilson ----

"That s not sweat running down your face wilson that s tears sad day s
eachickens thank you butler gopatriots patriotsvsseahawks"

----- Summary of tweets for given Key moment for julian edelman ----

julianedelman gopats gopatriots newengland. My husband and I live in N
ew England.

----- Summary of tweets for given Key moment for jimmy garoppolo ----
```

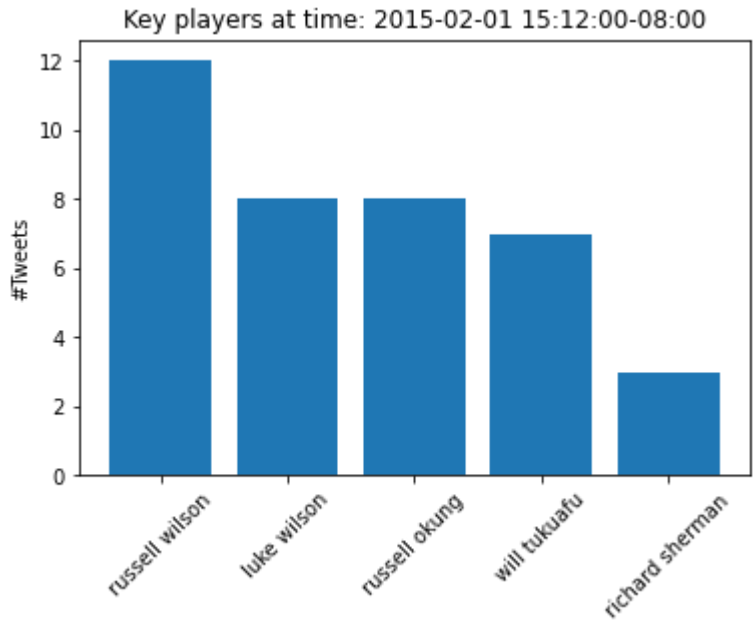



```
----- Summary of tweets for given Key moment for tom brady ----

Gopatriots. go pats pats go pats. amen brady. patriots4thring.. yeeeee
eeah brady! superbowl49 gopatriots champions.

----- Summary of tweets for given Key moment for malcolm butler ----

Butler with the clutch interception holyballsermergerd go patriotssss
ss superbowl champions sbxlix gopatriots patriotsnation.  butler for t
he win superbowlxlix. 28-24.. malcolm butler is adorable i had a visio
n that i was gunna make a big play correction the biggest play gopatri
ots superbowi. so mal Malcolm butler is a psychic cool beans hehadavis
ion gopatriot superbowl.

----- Summary of tweets for given Key moment for brian tyms ----

i know who my prophet is and it ain't lolololol prophet brian its over g
opatriots.

----- Summary of tweets for given Key moment for tavon wilson ----

Why wilson never gave lynch the ball for seattle i will never know.

----- Summary of tweets for given Key moment for jimmy garoppolo ----
```

Part 4 - Getting summary of the game using peaks from Hawks fans perspective

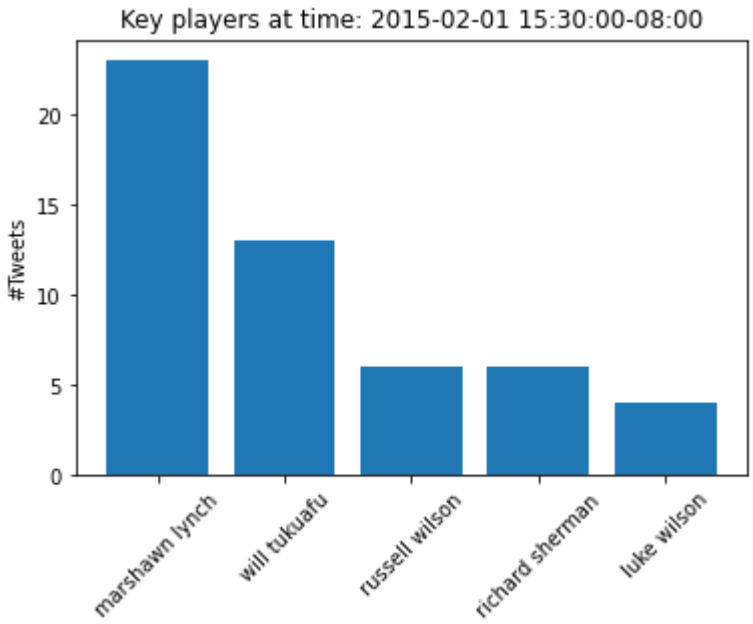
In [348]:

```
keyTimesGP = tsgopatriots
keyTimesGH = tsgohawks
keyTimesSB = tssb


for id, tw in enumerate(twgh):
    mp = getAssociatedPlayers(tw, hawksRoster)
    top = list(mp.keys())[0:5]

    values = []
    for tp in top:
        values.append(mp[tp])

    plt.bar(top, values)
    plt.xticks(rotation = 45)
    plt.title("Key players at time: {}". format(keyTimesGH[id]))
    plt.ylabel("#Tweets")
    plt.show()

    for tp in top:
        print("\n----- Summary of tweets for given Key moment for {} ---- \n ".forma
        print(getKeyTweets(tw, tp))
```

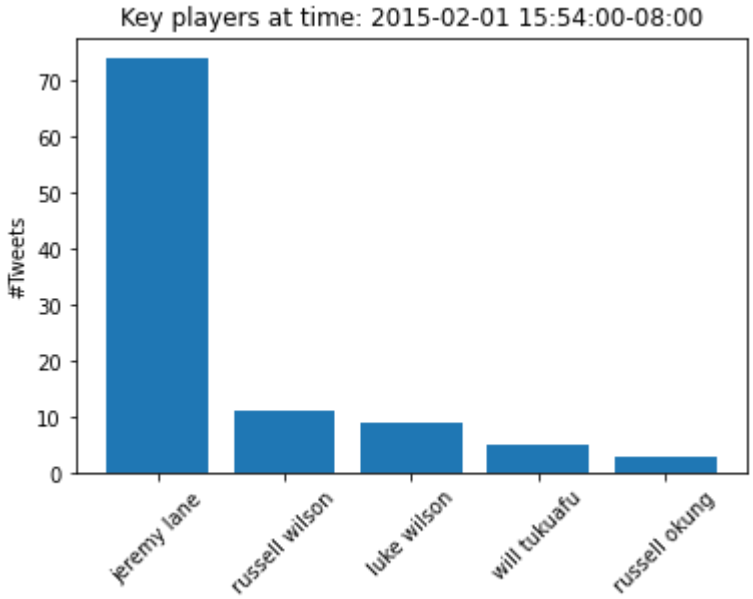



```
----- Summary of tweets for given Key moment for russell wilson ----

i m so glad that they didn t try to replicate the kurt russell intro f
rom last year that was so good so special superbowlxlix gohawks.. goha
wks sb49 is my pick to win the championship. I pick russell mvp and pa
triots 2 turnovers with fully inflated balls gohawks championshipminds
et.

----- Summary of tweets for given Key moment for luke wilson ----

i don t like either teams billsmafia but russell wilson is the fricken
man gohawks.  if brady win this one he s the truth automatic hall of f
ame.
```

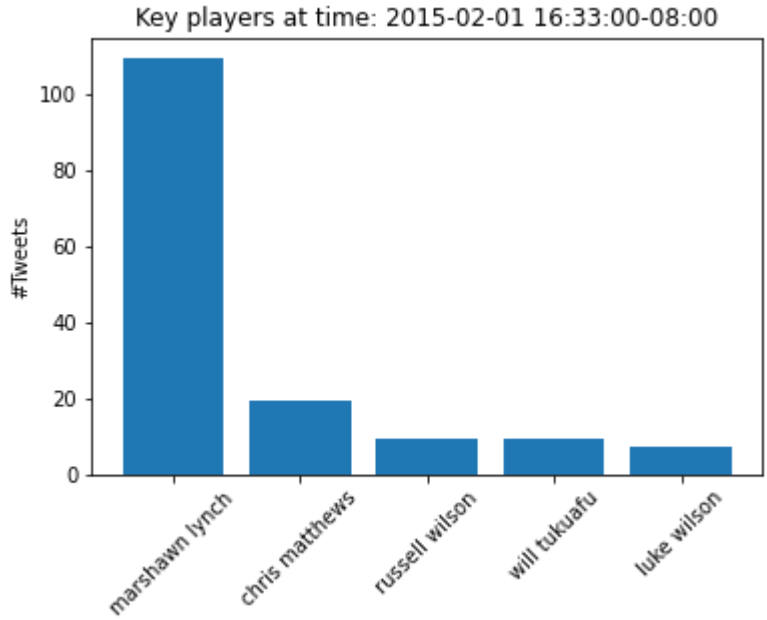
```
----- Summary of tweets for given Key moment for russell okung ----

i m so glad that they didn t try to replicate the kurt russell intro f
rom last year that was so good so special superbowlxlix gohawks.

----- Summary of tweets for given Key moment for will tukuafu ----

My man russell wilson will win the gohawks superbowl49.

----- Summary of tweets for given Key moment for richard sherman ----

i m wearing my richard sherman jersey if things start going wrong remi
nd me to change into the largent jersey i wore last year gohawks.
```



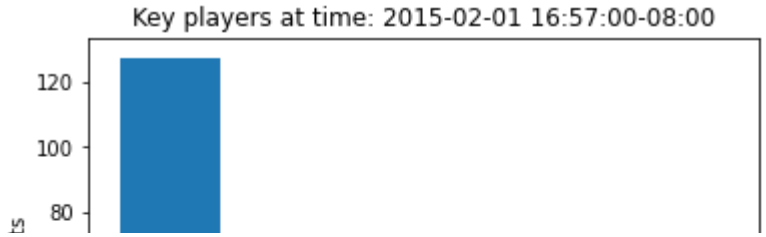

```
----- Summary of tweets for given Key moment for marshawn lynch ----

marshawn lynch eating skittles currently gohawks sb49.. You ain't gott
a score to grab your nuts marshawn lynch gohawks.. How many super fans
send marshawn lynch skittles superbowlxlix gohawks.

----- Summary of tweets for given Key moment for will tukuafu ----

Show us your hawks pride post your pic and we will pick a winner for a
hawksbouquet gohawks.  as long as i have a pulse the nationalanthem wi
ll choke me up superbowlxlix. i will tweet touchdown a lot even when i
m entirely wrong don t care touchdown gohawks.. The coin has been toss
ed superbowlcroatia will go 2 coach carroll gohawks superbowl hr.

----- Summary of tweets for given Key moment for russell wilson ----

"Let s go lob gohawks.  hustle like russell gohawks"

----- Summary of tweets for given Key moment for richard sherman ----

It s time for marshawn and sherman to feast gohawks.

----- Summary of tweets for given Key moment for luke wilson ----

Let s go lob gohawks.  let s do this general sherman leading president
wilson gohawks.
```



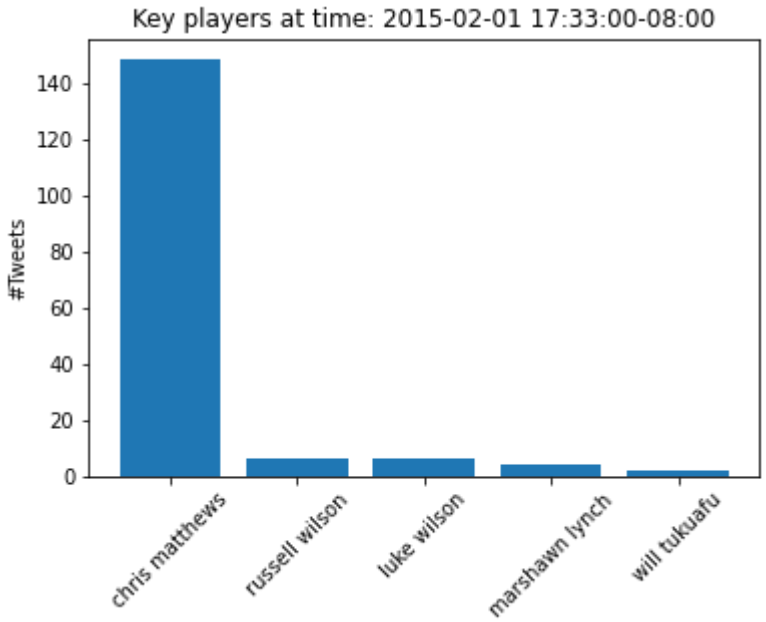

```
----- Summary of tweets for given Key moment for jeremy lane ----

A great pick by lane hopefully he s not hurt too bad.. oh hi tom have
you met jeremy lane yea that s him the one with your ball touchdown ho
pes deflated lob.. Huge interception in the end zone jeremy lane picks
off brady in the endzone gohawks lob sb49.

----- Summary of tweets for given Key moment for russell wilson ----

i can t root against russell wilson gohawks.  yes interception gohawks
wilson superbowlxlix cheering from browning montana seavsne.

----- Summary of tweets for given Key moment for luke wilson ----

i can t root against russell wilson gohawks.  yes interception gohawks
wilson superbowlxlix cheering from browning montana seavsne.

----- Summary of tweets for given Key moment for will tukuafu ----

tom brady will throw at least three more interceptions tonight gohawk
s.

----- Summary of tweets for given Key moment for russell okung ----

i can t root against russell wilson gohawks.  honestly i am totally co
ol with russell running his little heart out today.
```



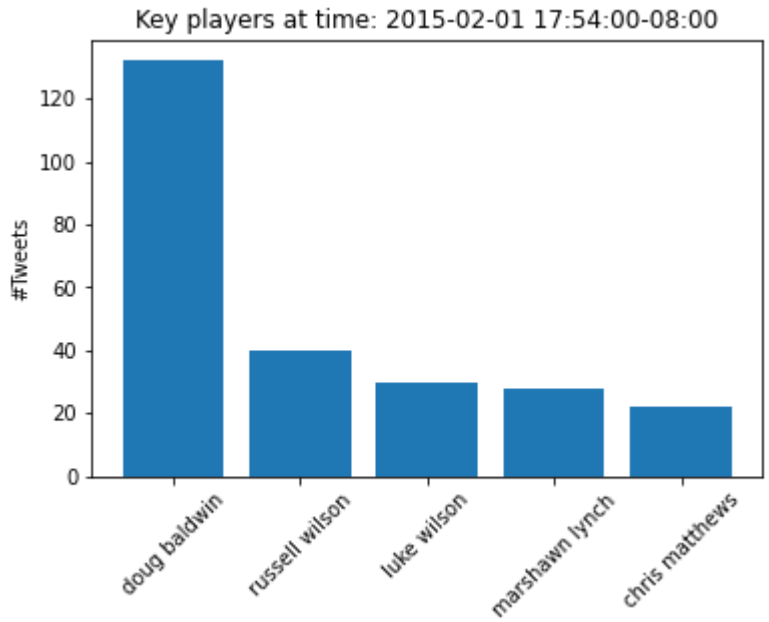

```
----- Summary of tweets for given Key moment for marshawn lynch ----

marshawn lynch with the 3 yard td to tie it 7 7 beastmode marshawnlync
h gohawks.. marshawn lynch with the tuddy superbowl wherestheskittles
12s gohawks.  touchdown for thx lynch beastmode gohawks.. The Seahawks
opened the skittles and lynch takes the ball in the endzone. lynch goh
awks sb49.

----- Summary of tweets for given Key moment for chris matthews ----

matthews makes a huge catch in the second quarter against Green Bay..
"Yeah baby you know why i'm here superbowlxlix gohawks touchdown eat t
hat fuckthepatriots.  way to go chris matthews bbn gohawks sb49.  russ
ellwilson chrismatthews beastmode yeah baby"

----- Summary of tweets for given Key moment for russell wilson ----

Got off work in time and now rocking my wilson jersey gohawks.

----- Summary of tweets for given Key moment for will tukuafu ----

feed the beast and he will score gohawks.

----- Summary of tweets for given Key moment for luke wilson ----

got off work in time and now rocking my wilson jersey gohawks.  gohawk
s wilson beast.
```




```
----- Summary of tweets for given Key moment for chris matthews ----

chris matthews is amazing gohawks sb49.  matthews damn this guy right
gohawks.  touchdown.. Chris matthews is our hero great call b3lieve 12
s gohawks.. chris matthews for the touchdown with 6 seconds left in th
e half.

----- Summary of tweets for given Key moment for russell wilson ----

russell wilson is the truth gohawks russellwilson superbowlxlix.. lync
h wilson answers brady touuuuuuuchdown seaaaaaahawks gohawks sb49.  wi
lson to matthews td seahawks.  lol russell wilson has like 4 completio
ns all game and he throws a td at an almost impossible moment gohawk
s.. Russel wilson to mathews again but a td this time.

----- Summary of tweets for given Key moment for luke wilson ----

here s hoping but not liking this call gohawks wilson superbowi.. russ
ell wilson is a baller prime time awesome gohawks.  wilson to matthews
for 6 tied at 14 tied at 14. wilson to mathews again but a td this tim
e gohawks. gohawks. sb49. "F*** yes, yes, fuck yes, yes". lets got pet
e carroll and russ wilson gohawks superbowlxlix.

----- Summary of tweets for given Key moment for russell okung ----

" russell wilson is the truth gohawks russellwilson superbowlxlix.  th
at s why russel wilson is so damn inspiring"

----- Summary of tweets for given Key moment for will tukuafu ----

In ten years my dad will most likely be coaching the seahawks gohawks
twins.
```

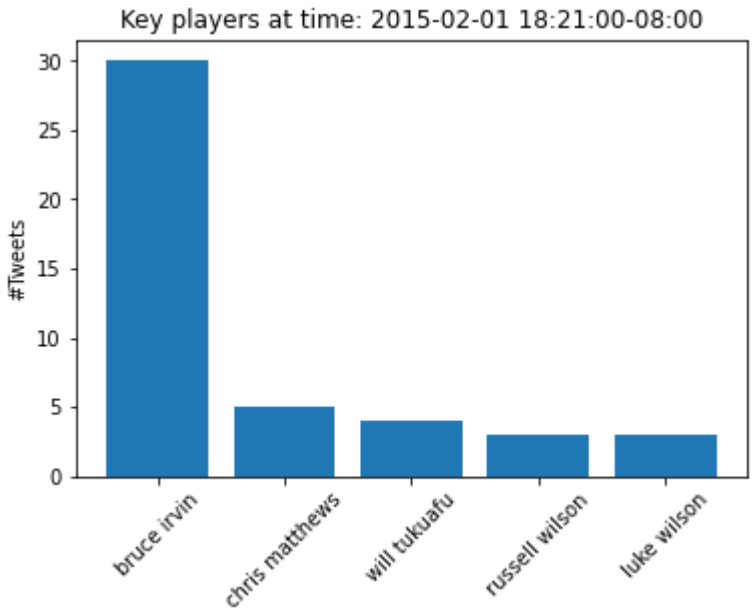


```
----- Summary of tweets for given Key moment for chris matthews ----

sb49. are you kidding me chris matthews the unsung hero with another m
arvelous catch.. chris matthews where have you been all my life gohawk
s.. chris matthews where were you before holy shit gohawks nevssea  pe
r bill bel Belichick it s a players game that s right.

----- Summary of tweets for given Key moment for russell wilson ----

Did russell wilson actually shower at halftime not a bad idea gohawks.

----- Summary of tweets for given Key moment for luke wilson ----

Did russell wilson actually shower at halftime not a bad idea gohawks.

----- Summary of tweets for given Key moment for marshawn lynch ----

marshawn lynch 3 yard touchdown run.

----- Summary of tweets for given Key moment for will tukuafu ----

patriots may have struck first but seahawks will strike last.
```

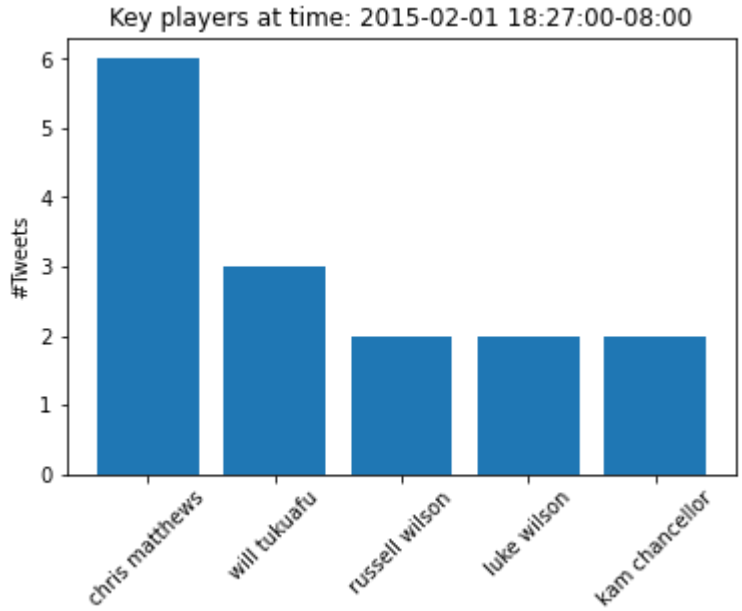



```
----- Summary of tweets for given Key moment for doug baldwin ----

"I guess doug baldwin is iight huh deon". Gohawks legionofboom.  touch
down doug baldwin gohawks sb49.  my favorite baldwin sbxlix gohawks..
baldwin capitalizes on the pick superbowlxlix superbowl sb49 nfl gohaw
ks goseahawks.  touchdown seahawks wilson to baldwin for 6 24 14 in th
e 3rd gohawks. angry doug baldwin gohawks sb49.  touchdown doug hawks
gohawks 89

----- Summary of tweets for given Key moment for russell wilson ----

russell wilson just playing leap frogger like a damn pro.. yeeesss dou
g baldwin has a great run amp a lovely little pass from wilson gohawk
s.. "Boom boom boom russell wilson you sir are a legend"

----- Summary of tweets for given Key moment for luke wilson ----
```

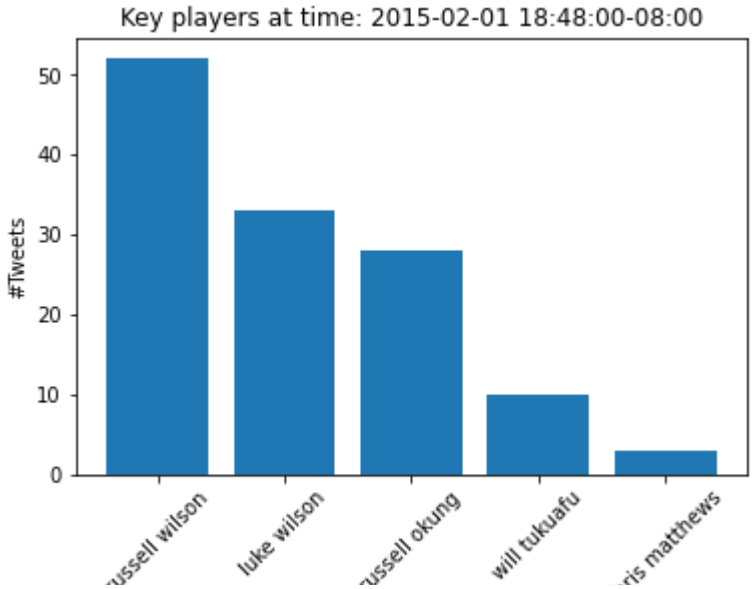
```
everydayhesrusselin rt first name russell last name wilson.. russell w
ilson to doug baldwin sb49 superbowl gohawks.  wilson to baldwin anoth
er hawks touchdown due to lynch s running beastmode ultimatesb go Hawk
s.  nice play by wilson but stupid penalty on lob weare12 gohawks.. wi
lson matthews lynch baldwin touchdown iloveyouseguys gohawks. wilson t
o baldwin for 6 24 14 in the 3rd.

----- Summary of tweets for given Key moment for marshawn lynch ----

marshawn lynch is a beast gohawks 12thman.. Never bet against russell
wilson and marshawn they will leave you sick i tell you sick.. Gohawks
beastmode marshawn lynch is doing work on dem patriots superbowlxlix.

----- Summary of tweets for given Key moment for chris matthews ----

chris collinsworth loves him some tommy brady.. gohawks.  superbowl mv
p chris matthews or russell wilson gohawks sb49.. marshawn lynch busti
ng through tackles and chris mathews coming up big they can t stop the
beast gohawks seahawks.
```



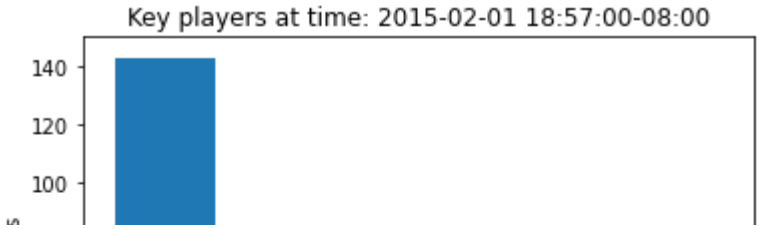

```
----- Summary of tweets for given Key moment for bruce irvin ----

What up bruce gohawks.  irvin with the sackarooskie gohawks superbowl2
015.. irvin is the first to sack brady gohawks sb49.. bruce irvin with
twerked before that sack gohawks sbxlix.  irvin has shown up so big th
is year just another time on the biggest stage superbowl gohawks.

----- Summary of tweets for given Key moment for chris matthews ----

Check out chris matthews for the seahawks gohawks sb49.

----- Summary of tweets for given Key moment for will tukuafu ----

nflrookie will you unleash beastmode this monday raw sb49 gohawks.

----- Summary of tweets for given Key moment for russell wilson ----

i would like to see a 99 wilson or baldwin ultimatesb wilson gohawks.

----- Summary of tweets for given Key moment for luke wilson ----
```




```
----- Summary of tweets for given Key moment for chris matthews ----

msnbc s chris matthews praises seahawks via gohawks sb49.

----- Summary of tweets for given Key moment for will tukuafu ----

i would like to order an interception now please i will take it to go
gohawks superbowl.

----- Summary of tweets for given Key moment for russell wilson ----

It's up to russell wilson to win this for the seahawks now gohawks.

----- Summary of tweets for given Key moment for luke wilson ----

It's up to russell wilson to win this for the seahawks now gohawks.

----- Summary of tweets for given Key moment for kam chancellor ----

kam almost jumped there gohawks.  chancellor get after it gohawks sb4
9.
```




```
----- Summary of tweets for given Key moment for russell wilson ----

russell wilson is made for this gohawks.. A solid drive and the take t
he lead who s ready for some russell wilson late game magic.. Russell
wilson has a chance to win his new contract with the Seattle Seahawks.

----- Summary of tweets for given Key moment for luke wilson ----

"I ain t even worried bout it my boy wilson bout to get nasty on um go
hawks.  alright wilson time to win this game gohawks nevssea hawks wil
son sb49smchat gohawks". cmon wilson gohawks.  time for wilson to work
his magical blessings.. My dad i don t think he s got it about wilson
ma he better go take a dump and get back in it.

----- Summary of tweets for given Key moment for russell okung ----

russell wilson visits seattle children s hospital every tuesday to mee
t patients.. in russell we trust sb49 q13fox.. russell wilson any rela
tion to bridgitte superbowl gohawks.

----- Summary of tweets for given Key moment for will tukuafu ----

i still believe we can do it we ve done it before we will do it again
lets go louder letsgo gohawks seahawks lob imin.

----- Summary of tweets for given Key moment for chris matthews ----

i don t want to see walters on the field ahead of matthews gohawks inr
usswetrust.
```

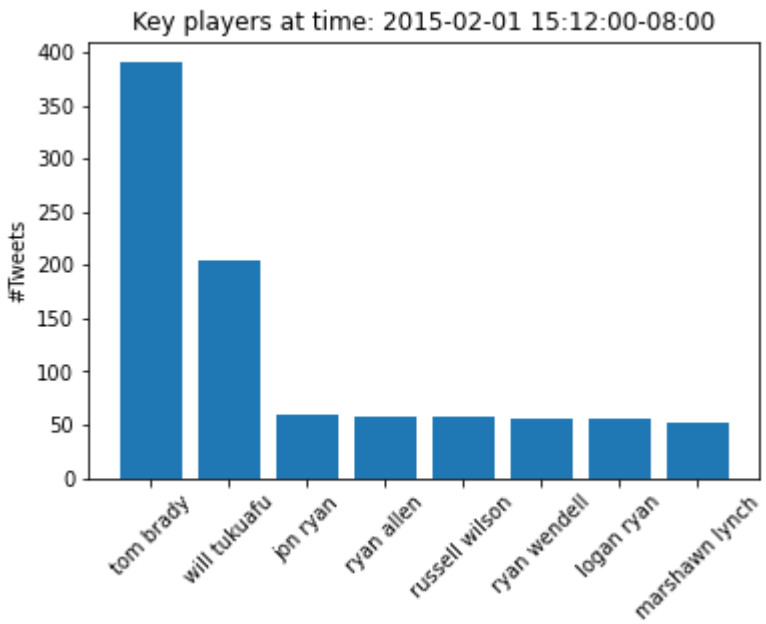



```
----- Summary of tweets for given Key moment for jermaine kearse ----

" kearse for president concetration gohawks.  kearse what a catch". "
kearse gohawks.  that catch by kearse will go down in history gohawks
superbowlxlix.  wow kearse got hands". kearse will you marry me 1 5 un
believable gohawks.  superbowlxlix.

----- Summary of tweets for given Key moment for will tukuafu ----

That catch by kearse will go down in history gohawks superbowlxlix.

----- Summary of tweets for given Key moment for russell wilson ----

gohawks baby12 goseahawks superbowl49 liveonkmtr.

----- Summary of tweets for given Key moment for russell okung ----

come on russell you got this first down gohawks baby12 goseahawks supe
rbowl49 liveonkmtr.  ok now get in there russell gohawks.

----- Summary of tweets for given Key moment for david king ----

i just shit my pants sick catch lmao david tyree manningham kearse pat
riots are sick.
```

Part 4 - Getting overall summary of the game at different keypoints

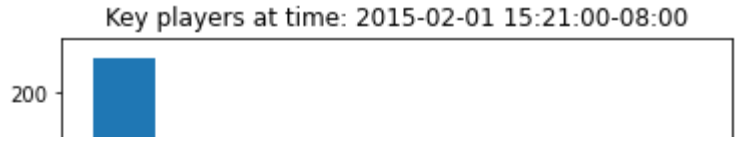
In [349]:

```
keyTimesGP = tsgopatriots
keyTimesGH = tsgohawks
keyTimesSB = tssb

fullroster = list(set.union(set(PatriotsRoster), set(hawksRoster)))
for id, tw in enumerate(twsb):
    mp = getAssociatedPlayers(tw, fullroster)
    top = list(mp.keys())[0:8]

    values = []
    for tp in top:
        values.append(mp[tp])

    plt.bar(top, values)
    plt.xticks(rotation = 45)
    plt.title("Key players at time: {}". format(keyTimesSB[id]))
    plt.ylabel("#Tweets")
    plt.show()

    for tp in top:
        print("\n----- Summary of tweets for given Key moment for {} ---- \n ".forma
        print(getKeyTweets(tw, tp))
```



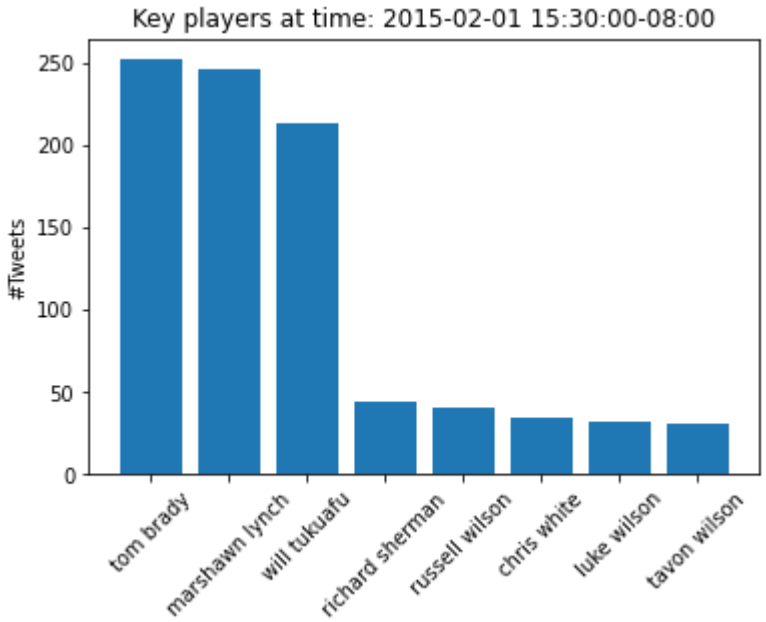

```
----- Summary of tweets for given Key moment for tom brady ----

The Patriots take on the Falcons in Sunday night's superbowl.. 4th rin
g for brady on the way patsnation superbowlxlix.. i hate tom brady sup
erbowl superbowlxlix sb49 patriotswin.

----- Summary of tweets for given Key moment for will tukuafu ----

i think tonight the seahawks will still rule superbowl iwontstayuplat
e.. superbowlxlix will donate 1 to madd for every trip taken when user
s enter the promo code thinkandride today drivesafe. patriotsnation.co
m predicts patriots will win.. Will marshawn lynch join the long list
of famous golden moments superbowlxlix goldenmoments.

----- Summary of tweets for given Key moment for jon ryan ----
```

```
Rex ryan and tony romo finally made it to the superbowl pizzahutcommer
cial.. Apparently john ryan is from the universityofvagina superbowl s
uperbowlxlix. did jon ryan just say he went to the university of vagin
a.. hahahaha bittersweet symphony is the seahawks theme song thinking
of ryan philippe doing bad things isn t very inspiring superbowl.

----- Summary of tweets for given Key moment for ryan allen ----

it just sounded like jon ryan from the seahawks said the university of
vagina superbowl superbowlxlix sb49.. hahahaha bittersweet symphony is
the seahawks theme song thinking of ryan philippe doing bad things isn
t very inspiring superbowl.. Rex ryan isn t much better as a commercia
l actor superbowlxlix.

----- Summary of tweets for given Key moment for russell wilson ----

Seattle Seahawks defeated Seattle Seahawks 28-21. russell wilson mvp s
uperbowl 12s. russel wilson m lynch r sherman k chancellor e thomas se
ahawks superbowlxlix.. The reigning super bowl champions seattle seaha
wks led by russell wilson.. russle wilson seattleseahawks superbowl.
let s go seahawks sb49 lynch sherman wilson.

----- Summary of tweets for given Key moment for ryan wendell ----

At least rex ryan has a better job than coaching the jets superbowlxli
x.. I'm pretty sure john ryan just said university of vagina superbow
l.. Did john ryan from seattle special teams just say he was from the
university of vagina superbowl.  johnryan superbowlxlix superbowl49 bu
ffalobills pizzahut rex ryan s favorite pie is feetza.

----- Summary of tweets for given Key moment for logan ryan ----

I'm pretty sure john ryan just said university of vagina superbowl.. j
ohn ryan university of vagina superbowlxlix.  did jon ryan say univers
ity of vagina if not sure did sound like it superbowl.. Rex ryan throw
s a red flag and nails a guy in the nuts superbowlcommercials.

----- Summary of tweets for given Key moment for marshawn lynch ----

i just want marshawnlynch s victory presser superbowl.. Seahawks to wi
n by 6 points marshawn lynch will be the difference superbowlxlix. vil
la bolo ready for the night superbowlxLix go seahawks marshawnlynch be
astmode.. I am firmly in the patriots camp tonight cause of brady but
can t wait to see lynch in beast mode absolute brute superbowl patriot
svsseahawks.
```



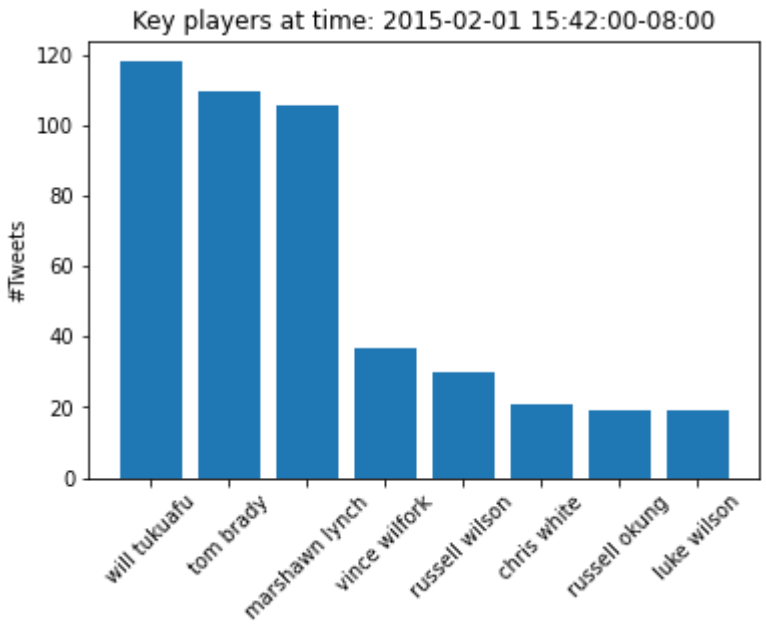

```
----- Summary of tweets for given Key moment for will tukuafu ----

Vote now for your favorite team in superbowlxlix.. superbowl 2015 is o
n tonight at 8pm ET.. i predict will cash in his briefcase in tonight
superbowl.

----- Summary of tweets for given Key moment for tom brady ----

tom brady is the captain of the patriots. The newenglandpatriots are t
he champions of the new england clam chowder.. New England Patriots ta
ke on seattle seahawks in superbowl. Fans supporting the patriots beca
use Tom brady was in a family guy episode superbowlxlix.. The game is
on patriots to win really close game seahawks defence to be unbelievab
le but brady makes his chances count superbowlxlix.

----- Summary of tweets for given Key moment for russell wilson ----

It's all from houston ne danny amendola brandon lafell michael bennett
cameron fleming sea russell okung david king superbowl sb49 nfl.. russ
ell wilson is super cute. I'm thirsty for superbowl superbowlxlix.. th
e social media geek in me loves this time of year tweet away tweeps su
perbowl goseahawks wilson bandwagon.

----- Summary of tweets for given Key moment for luke wilson ----

russell wilson and the seattleseahawks will win superbowlxlix superbow
lsunday seavsne. patriotsvsseahawks. seattle seahawks is winning this
lynch and wilson have to lead the team so that everyone is playing at
110 superbowl seattleforthewin. mVP mvp kam kam chancellor or russel w
ilson will win the superbowl. patriotsvseahawks will win by 10 point
s.. i love russel wilson that much more because he put his hand on is
heart icried iloveamerica nationalanthem superbowl gohawks.. Fans are
backing russell wilson and the seahawks to take out superbowl xlix.

----- Summary of tweets for given Key moment for tavon wilson ----

i can accept russell wilson getting another ring superbowlxlix.. super
bowl patriots vs seahawks is on tonight at 8 p.m. ET.. 8 mins to go se
ahawks let s make history superbowl seahawkswin seavsne wilson brady q
uarterback.

----- Summary of tweets for given Key moment for russell okung ----

i just wanna run my fingers through russell wilson s luxurious locks s
uperbowlxlix.. If the seahawks win tonight russell wilson will be mvp
trophy superbowlxlix.. i can accept russell wilson getting another rin
g superbowlxlix.

----- Summary of tweets for given Key moment for marshawn lynch ----

i got the seahawks 27 patriots 26 superbowl mVP marshawnlynch.. marsha
wn lynch to go absolutely h a m superbowlxlix. seattle seahawks lets a
nnihilate tom brady and his patriots.. The patriots outmatch almost al
l players for the seahawks.

----- Summary of tweets for given Key moment for richard sherman ----
```

```
What s with these close ups of richard sherman superbowlxlix.. i like
richard sherman i like anyone who talks the talk and then walks the wa
lk i can t stand those who talk and then waddle superbowl.. Richard sh
erman's rendition of the national anthem was reminiscent of little ric
hard in mystery alaska superbowlxlix. hey idiots there s plenty of opp
ortunity to boo sherman and belichek during the national anthem isn t
the time superbowl.
```



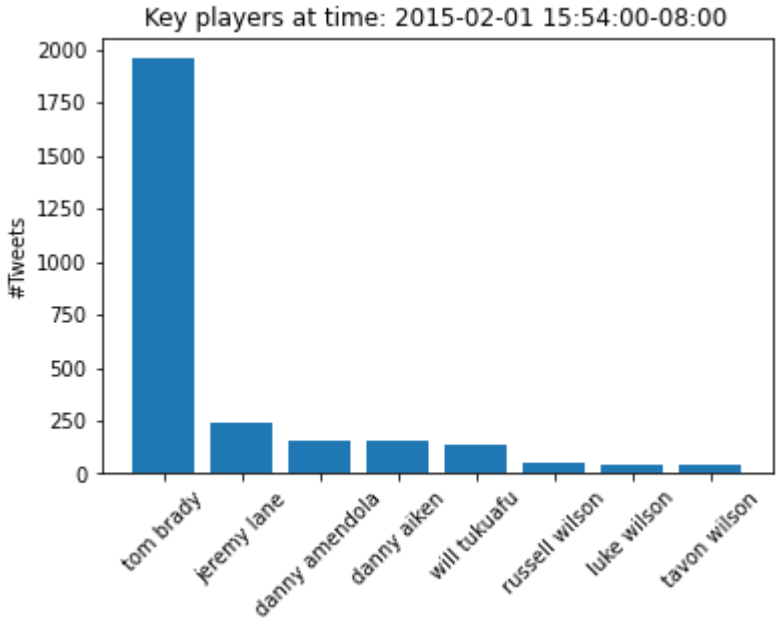

```
----- Summary of tweets for given Key moment for tom brady ----

How much of a steal at 199 6th round was tom brady?. am i the only one
that thinks tom brady is not at all attractive his kinda basic superbo
wlxlix.. tom brady is a great leader who could still win even if spud
webb was his only wide receiver superbowl sb49.

----- Summary of tweets for given Key moment for marshawn lynch ----

marshawn lynch is eating his skittles superbowlxlix.. marshawnlynch ea
ting before superbowl. marshawn lynch beastmode powered by a merchant
superbowl.. My favorite part about marshawn lynch when not trucking du
des on the field he looks like a guy pretending to be marshawn Lynch s
uperbowl.

----- Summary of tweets for given Key moment for will tukuafu ----

superbowlxlix will be a nail biting commentary on the footballs deflat
egate.. i know absolutely nothing about nfl but seattle seahawks will
win superbowl seavsne superbowlxlix.. It is estimated over 110 million
people will be watching the super bowl game on superbowl49.

----- Summary of tweets for given Key moment for richard sherman ----

Sherman is just an absolute tank superbowlxlix.. i already fancy the a
rse off richard sherman superbowlxlix.. The Patriots take on the Seaha
wks in Super Bowl XLIX. The game will be played in Seattle's home stad
ium, CenturyLink Field. The game is set to kick off at 8:30 p.m. ET on
Sunday night. Richard sherman is expected to start for the Patriots.

----- Summary of tweets for given Key moment for russell wilson ----
```

```
Rt prediction seahawks to win by more than 7pts russell wilson to be m
VP superbowl.. seattle seahawks and russell wilson patriotsvsseahawks
superbowl.. Watching the superbowl and rooting for russell wilson beca
use he s teamjesus.

----- Summary of tweets for given Key moment for chris white ----

Don't care what anyone says jurassicworld superbowlxlix.. i'm not exci
ted to hear chris collinsworth s voice the whole game terrible annoyin
g ithinkiknoweveryone sb49 superbowl.. chris pratt trained the velocir
aptors in jurassicworld trailer that could be awesome superbowl movie
s.

----- Summary of tweets for given Key moment for luke wilson ----

i want the patriots but russell wilson is so sexy superbowlxlix.  pred
iction seahawks to win by more than 7pts.. mysticmorgan rt prediction
seahawks to win by more than 7pts russell wilson to be mVP superbowl..
i like the confidence and spunk of wilson over the poise and experienc
e of brady seattle 30 new england 20 superbowl.

----- Summary of tweets for given Key moment for tavon wilson ----

The new england patriots 35 19 calledit superbowlxlix. I'm happy to ma
ke black history month because it's super bowl time. I'm rooting for r
ussell wilson because he s teamjesus.. i want the patriots but russell
wilson is so sexy superbowlxlix.  i must inquire wilson can you still
have fun duhduhdu hduh phish superbowl.  confirmation that russel wils
on s beard is not prosthetic superbowlXlix.. Tom brady vs russell wils
on superbowlxlix. wilson is a class act amp i like i m cheering for be
gr8 if wilson beat manning amp then brady in superbowl.
```



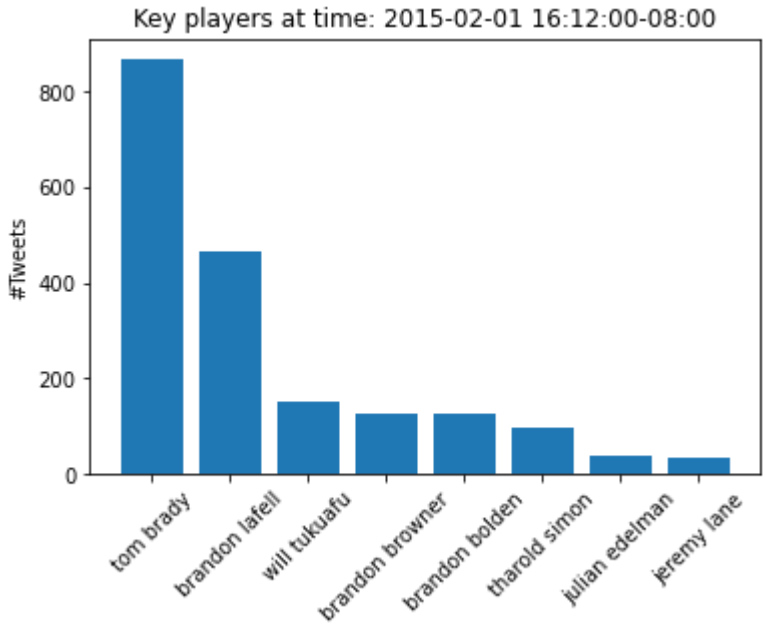

```
----- Summary of tweets for given Key moment for will tukuafu ----

Tom Brady in a swerve will leg drop ron gronkowski and form the new fo
otball order with roger goodell superbowl.. If they spelt defence wron
g heads will roll for that one superbowl.. After 5 minutes play litera
lly nothing has happened will anything actually happen superbowl hande
gg.
```

```
----- Summary of tweets for given Key moment for tom brady ----

Tom brady will leg drop ron gronkowski and form the new football order
with roger goodell superbowl.. Tom brady patriots superbowl.  tom brad
y isajerk superbowlxlix freehernandez imonlyhereforthecommercials.  i
don t follow football but i do love the superbowl seattle ftw cuz i li
ke their coach s vibe more and cuz brady is too good looking.  fuck yo
u wilson especially tom hanks volleyball superbowl.. Let s go brady er
r again superbowlxlix. Let s go pats.

----- Summary of tweets for given Key moment for marshawn lynch ----

i wonder how 2000 team would have handled marshall lynch superbowlxlix
superbowl.  rt 3 lynch runs no first down your thoughts marshawn super
bowl.. It took the entire patriots defensive line to tackle lynch beas
t mode was charging until that 3rd down massive pats defense.. marshaw
n lynch runs for 3 and out for the seahawks.

----- Summary of tweets for given Key moment for vince wilfork ----

vince wilfork makes an early impression in the superbowl for the patri
ots.. you can t block vince wilfork with one man superbowl.. Vince wil
fork is a wrecking ball superbowlxlix.

----- Summary of tweets for given Key moment for russell wilson ----

The hawks trying to establish the running game early good for wilson o
ption plays. If the patriots can continue to stop the run they will wi
n. If lynch stopped but football season clock ticking superbowl.. wow
russell wilson is 10 0 vs qb s who previously won a superbowl gohawk
s.. russell wilson is a legend killer 10 0 against super bowl qbs supe
rbowl triplethreatsports.com.

----- Summary of tweets for given Key moment for chris white ----

i'm not gonna lie i kind of wanted the national anthem to start out wi
th the snow glows white on the mountain tonight superbowl.. i can t fo
llow the plot of this game are the bad guys the fellas in the white co
stumes or the blue outfits goteam superbowl.  i wear a sharp tie and a
crisp white shirt and my wife doesn t feed me pizza like that budlight
living superbowl superbowlad.  a white return man screams fair catch s
uperbowl.. "When he asked that white bitch can you twerk?"

----- Summary of tweets for given Key moment for russell okung ----

The last pick of the draft, russell wilson will be the mvp superbowl..
The patriots defense is bringing everyone on marshawn lynch plays russ
ell wilson will have to beat them with his arm.

----- Summary of tweets for given Key moment for luke wilson ----

The patriots defense bringing everyone on marshawn lynch plays russell
wilson will have to beat them with his arm superbowl.. i hope russell
wilson jon ryan have a good game topblokes superbowlxlix.
```



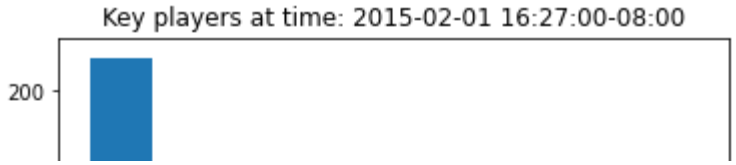

```
----- Summary of tweets for given Key moment for tom brady ----

tom brady threw that last one too low too much air in the ball perhaps
superbowlxlix deflatriots.. New England hasn't scored a point in the 1
st quarter in any of brady s super bowls superbowlxlix. brady cant thr
ow nuttin with those little girl hands and a properly inflated ball su
perbowl tombrady deflategate seahawkswin.. the ball must not feel righ
t to brady superbowl nevssea.

----- Summary of tweets for given Key moment for jeremy lane ----

jeremy lane picks off tom brady at the goal line sb49 superbowlxlix..
Jeremy lanes just won superbowl mvp. well played rt.. Great play by je
remy lane superbowlxlix.

----- Summary of tweets for given Key moment for danny amendola ----

danny trejo and steve buscemi marshamarshamarsha superbowl.. Snickers
commercials are the bees knees superbowlxlix snickers.. The snickers c
ommercial w danny trejo and steve buscemi is hilarious hewitt superbow
l.

----- Summary of tweets for given Key moment for danny aiken ----

Can t hate on a commercial with danny trejo and steve buscemi in it su
perbowl.. i don t even know what was just being hawked but danny trejo
on the brady bunch was worth the price of admission superbowlcommercia
ls.  love the snickers commercial with steve buscemi and danny Trejo b
rilliant superbowlxlix.  omg snickers with dannytrejo i m dying superb
owl superbowl commercials.. danny trejo meets the brady bunch it s abo
ut time superbowlcommercials.

----- Summary of tweets for given Key moment for will tukuafu ----

i don t like good will hunting because i consider it part of the patri
ots organization superbowl.. Since my team isn t playing tonigh i have
to cheer for some hawks so the seahawks will do superbowlxlix ihatetom
brady.. patriots vs seahawks who will win gooo patriots superbowlxlix.

----- Summary of tweets for given Key moment for russell wilson ----
```

```
russel already breakin ankles ooooo superbowlxlix sb49.  russell shook
the shit outta him superbowl.  wilson should have tried to run more he
could have done it superbowl.. russel wilson breaks for a run 2 amp 3
superbowl.. Russell wilson is 10 0 against super bowl winning qbs his
defense is one of the best in nfl history superbowl.  fly wilson fly s
uperbowl.

----- Summary of tweets for given Key moment for luke wilson ----

russell wilson is a danger with the feet work superbowlxlix.. russell
wilson got lucky but good run superbowlxlix.. Russell wilson is one sl
ippery scrambler does so well superbowl.

----- Summary of tweets for given Key moment for tavon wilson ----

russell wilson's footwork nice asf superbowlxlix. dat nigga can t give
wilson that much time superbowl.. u can not sack wilson superbowl.. ru
ssell wilson out here breaking ankles and 1 mixtape style superbowlxli
x.
```



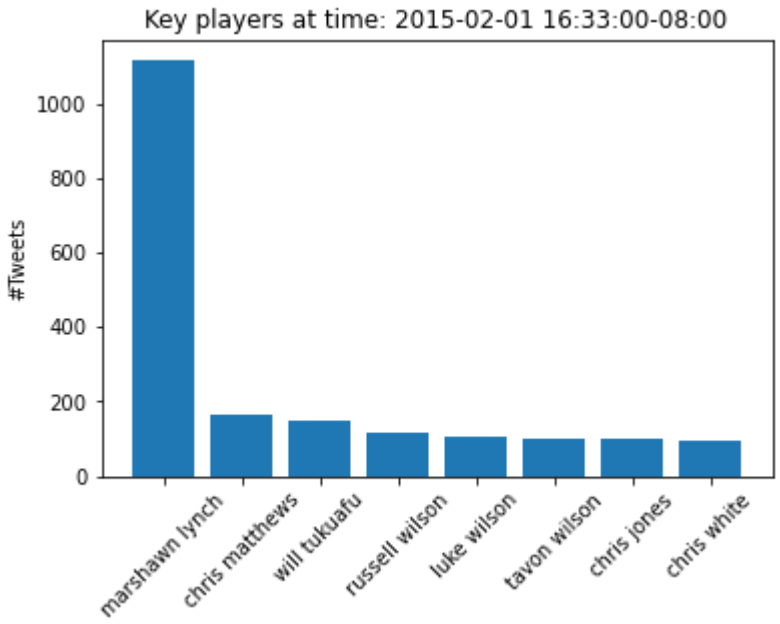

```
----- Summary of tweets for given Key moment for tom brady ----

Tom brady and the patriots superbowl.  touchdown on a pass by brady to
lafell superbowl patsnation.. This one is in my top 3 right now snicke
rs the brady bunch ads superbowl brandbowl.. i quite fancy tom brady s
uperbowl.  touchdown new england brady to lafell thats what i am talki
ng about brady lafell newenglandpatriots superbowl superbowlxlix sb49.

----- Summary of tweets for given Key moment for brandon lafell ----

brady gt lafell touchdown pats lead 7 0 midway through q2 superbowl.
on lafell td great route hard stick settles down before reaching safet
y perfection superbowlxlix.. Lsu alumni brandon lafell got the first t
ouchdown in the superbowl.. patriots strike first brady lafell 7 0 sup
erbowlxlix. The pats take a 7 0 lead over the seahawks.

----- Summary of tweets for given Key moment for will tukuafu ----

It's great now technically illiterate people will try to improve their
internet by pouring coke on their modems.. This superbowl is all about
```

```
precision football amp deadly accurate qb play. Two great defenses coa
ching amp execution will decide this game.. The patriots score first t
ouchdown will now punt to england red coats superbowl. Will seattle sc
ore today superbowl49 superbowl.  so high fructose corn syrup water am
p caramel coloring will bring happiness to the people of cocacola supe
rbowlxlix.  will somehow end cyber bullying apparently superbowl comme
rcials.

----- Summary of tweets for given Key moment for brandon browner ----

Brandon lafell has the first score of the day pats are up ne 7 0 sea 2
nd qtr superbowlxlix.  and the sec is back on the board in a big game
lsu s brandon Lafell scores td go give patriots the lead superbowlxLi
x.. Brandon lafell scores a touchdown for the patriots against the sea
hawks.. The patriots go up 7 0 with a touchdown from brandon lafell.

----- Summary of tweets for given Key moment for brandon bolden ----

patriots lead 7 0 with 9 47 left in the 2q. Get in brandon lafell grea
t td for the patriots 7 0 gopats superbowl superbowlxlix.. Brandon laf
ell scores the first touchdown of the game for the patriots.. brandon
lafell with the touchdown new england patriots with 7 on the board. sb
49 superbowlxlix superbowl.

----- Summary of tweets for given Key moment for tharold simon ----

It s not a good day to be named simon i guess pens superbowl.. superbo
wl seahawks take on the Patriots in Sunday's NFL season opener.. brady
to simon: "The pats don't found a weak link in that d that simon dude
smh superbowl.  i blame simon s terrible tackling for that td superbow
l"

----- Summary of tweets for given Key moment for julian edelman ----

i hope edelman gets injured because of what he did to jeremy lane supe
rbowlxlix sb49.. Julian edelman is going to be a thorn in seattle s si
de all night superbowlxlix.. A crossbreed of tom brady and julian edel
man would be the fucking coolest bastard ever superbowlxlix.

----- Summary of tweets for given Key moment for jeremy lane ----

Edelman is a beast superbowlxlix simon lane are sorry corners no doub
t. that interception looks important already but only because lane got
taken out as a result superbowl.. ouch rt gruesome photo from collisio
n between jeremy lane and julian edelman superbowl.. We need lane seah
awks superbowlxlix. She was so excited right after lane s play she was
so excited.
```



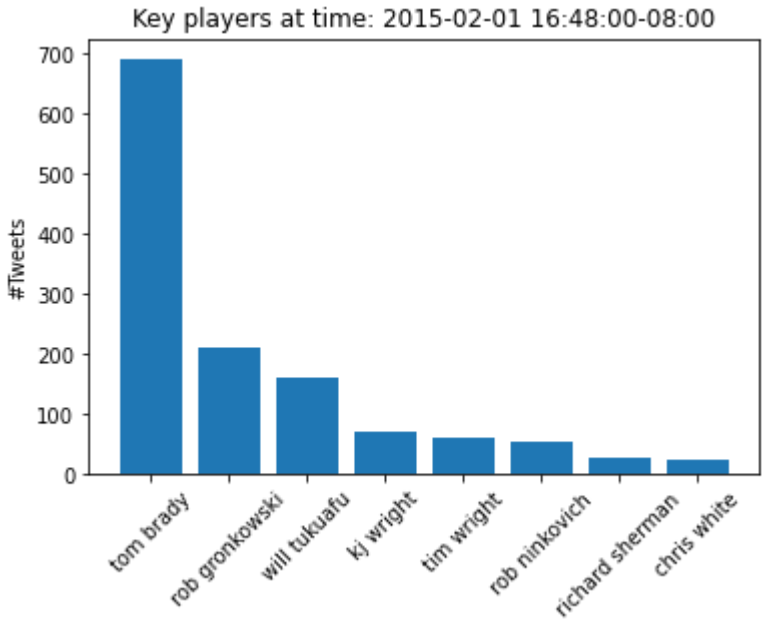

```
----- Summary of tweets for given Key moment for will tukuafu ----

everything will kill you nothing is safe except coca cola superbowl..
i will scream if i see one more depressing insurance comp commercial s
b49 superbowlxlix. that comercial about the kid who will never do stuf
f is dark as hell superbowl wtf why.. Super Bowl commercials will feat
ure funny ads for anti depressants well played big pharma.

----- Summary of tweets for given Key moment for tom brady ----

Tom Brady is in his 6th super bowl and win or lose he s sleeping next
to gisele tonight.. tom brady is the man patriotswin superbowlxlix. to
m brady can t throw these properly inflated balls superbowl sb49 sb49s
mchat. go england bora poxa tom brady first td e seahawkes so olhando
cade wilson.. tom brady throws the same short passes man up no zone ne
eded superbowlxlix.

----- Summary of tweets for given Key moment for steven hauschka ----

What died on steven tyler s face superbowl.. i m glad steven tyler is
enjoying the superbowl meanwhile i m still recovering from the bleak f
uture nationwide has painted for us.. Trash performing at halftime sup
erbowl.

----- Summary of tweets for given Key moment for steven terrell ----

The skeleton of steven tyler is in attendance at this game superbowlxl
ix.. sir paul mccartney and steven tyler why can't they perform this y
ear superbowl.. steven tyler got a mustache you guys superbowl.

----- Summary of tweets for given Key moment for russell wilson ----

If i had a dollar for every time russell wilson completed a pass i wou
ld have 1 dollar superbowl.. i think russellwilson is really a trolldo
ll superbowlxlix.. russel wilson hasn t got into the game seahawks loo
k flat superbowl.

----- Summary of tweets for given Key moment for luke wilson ----

superbowl superbowlxlix sb49. russel wilson hasn t got into the game s
eahawks look flat superbowl. seattleseahawks newenglandpatriots footba
ll americanfootball superbowl.. i can t believe russell wilson doesn t
have 1 completion shocked superbowl superbowlsunday.  come on wilson u
se your magic superbowl seahawks.. rt rt if you have as many completio
ns as russell wilson seahawks patriots superbowl.

----- Summary of tweets for given Key moment for tavon wilson ----

i can t believe russell wilson doesn t have 1 completion superbowl.
I'm shocked at how well superbowl superbowlsunday is going.. i bet tha
t brady and wilson both wish they could use echo1612 on the sidelines
superbowl echo1612 tomorrow sadjustmentstoday.. superbowl superbowlxli
x sb49. superbowlsunday literalrape. come on wilson use your magic sup
erbowl seahawks.
```

```
----- Summary of tweets for given Key moment for kevin williams ----

kevin hart sitting next to will ferrell lots of jokes in that suite i
bet guffaws might even perform a sketch or two superbowlxlix.. The sta
rs are aligning for superbowlxlix. kevin hart will ferrell paul mccart
ney mark wahlberg amp kenny chesney.. What do you think john travolta
and kevin hart are discussing up there awkward superbowlxlix.
```



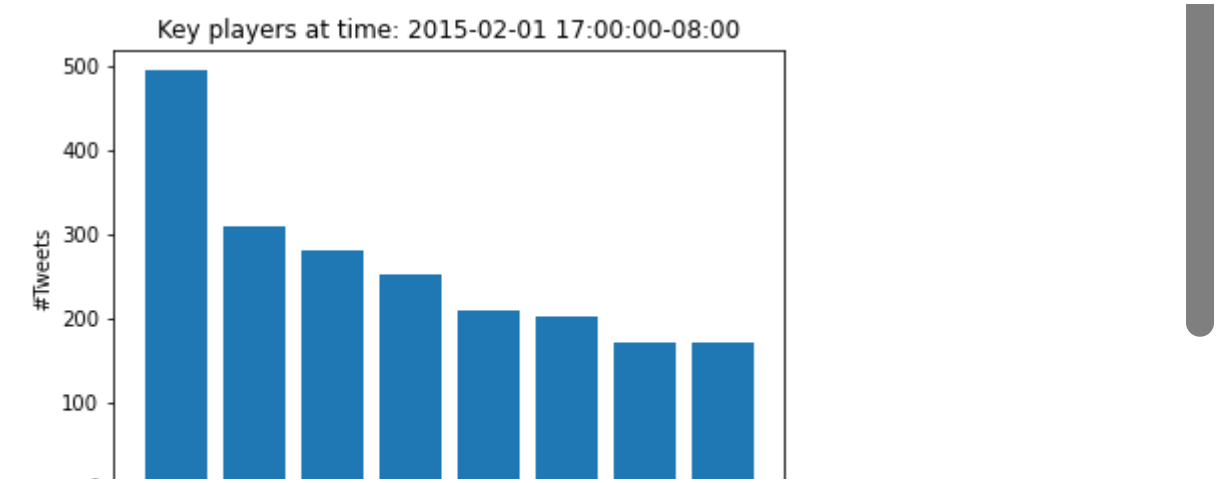

```
----- Summary of tweets for given Key moment for marshawn lynch ----

lynch leads the NFL in touchdowns with 51.. patriots and seahawks tied
7 7 late in the 2nd quarter of superbowl xlix. lynch td 7 7 superbowlx
lix nesea. marshawn lynch for seattle score 24-24.. lynch is a beast b
eastmode superbowlxlix.

----- Summary of tweets for given Key moment for chris matthews ----

chris matthews catches his first career catch in superbowl.. superbowl
xlix. how did matthews catch that under so much pressure.. yeah you lo
ve man on man don't you chris collinsworth superbowlxlix.

----- Summary of tweets for given Key moment for will tukuafu ----

This year will be known as the boohoobowl sadcommercials superbowlxli
x.. 2nd half will be intense blood is boiling superbowlxlix.. i honest
ly want the patriots to win just so will perform a song naked at a sho
w superbowl.

----- Summary of tweets for given Key moment for russell wilson ----

superbowlxlix.  out of nowhere seattle has a td wilson s big play to m
atthews made that why he s only attempted 2 passes so far is beyond m
e.. russell wilson wants to repeat his nfc championship performance su
perbowlxlix. The seahawks answer backin a big way wilson to lynch to t
ie it up 7 to 7 superbowl sb49.. yeahhhh wilson to lynch touchdown sea
hawks it s 7 7 superbowl.  pass was fire wilson wow superbowl.

----- Summary of tweets for given Key moment for luke wilson ----

russell wilson just knows how to turn it on all of a sudden.. russell
wilson is the second flukiest qb i ve ever seen three completions in a
```

```
while half overrated.. okay russell wilson and chris matthews that was
pretty superbowlxlix.

----- Summary of tweets for given Key moment for tavon wilson ----

The game is tied at 7.. A great job by russell wilson as the seahawks
come from behind to beat the superbowl.. news flash new england when w
ilson rolls out he throws deep oh and marshawn lynch is an absolute be
ast.

----- Summary of tweets for given Key moment for chris jones ----

chris matthews is superman seahawks superbowlsunday.  chris collinswor
th loves his man on man superbowl.. chris collinsworth is like an fan
he likes whoever is winning superbowl.. for a first catch chris mathew
s you made it a good one superbowlxlix.

----- Summary of tweets for given Key moment for chris white ----

Former wr chris matthews making a big play superbowl seahawks.. chris
matthews with the huge catch bbn superbowlxlix. chris pratt is 1000 fu
ckable in this jurassic park preview superbowl.. chris matthews with a
44 yard catch on this drive hardball superbowlxlix sb49 nevssea snf.
```



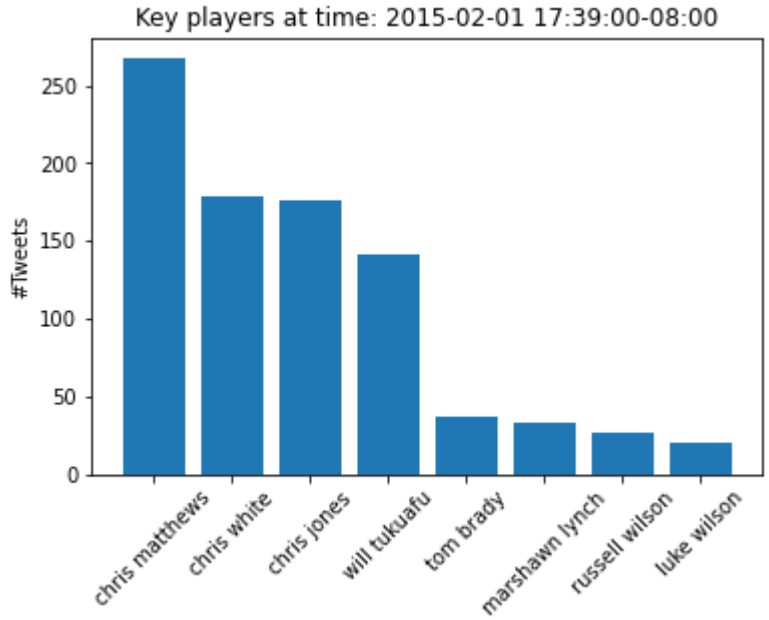

```
----- Summary of tweets for given Key moment for tom brady ----

Brady to gronk for another touchdown earthquakespike to boot 14 7 patr
iotsvsseahawks superbowlxlix.. superbowl. patriotswin patriotsvsseahaw
ks superbowlxlix superbowl2015.. superbowl super pass by brady touchdo
wn 14 7 pats lead.  brady to gronk finally touchdown superbowlxlix.

----- Summary of tweets for given Key moment for rob gronkowski ----

rob gronkowski puts the pats into the lead just before ht ne 14 7 seat
tle gronknation superbowlxlix.  i think gronkowski s arm is the same s
ize as my leg superbowl.  gronk robgronkowski nice touchdown.. Brady t
o gronkowski touchdown superbowl.  touchdown of gronkowski ne patriots
14 7 seattle seahawks 2nd quarter 0 31 superbowlxlix.  14 7 new englan
d patriots robgronkowski superbowl.. Gronkowski with the touchdown gro
nknation superbowl.  tom amp rob a deathly combination of boom and boo
m touchdown superbowlxlix patsnation.  and there s gronkowski doing wh
```

```
at he does patriots superbowl.

----- Summary of tweets for given Key moment for will tukuafu ----

Sb49 superbowlxlix singers.. microsoft will build your kid s prostheti
c legs and then the legs will crash superbowlcommercials.. The seahawk
s will win gohawks superbowlxlix.

----- Summary of tweets for given Key moment for kj wright ----

Why was kj wright on him lmaooo kam should have been out there superbo
wlxlix.. kj wright burned superbowlxlix.  huge mismatch no way wright
can cover gronk big mistake by seahawks superbowl49.. Why was wright c
overing gronk superbowlxlix superbowi.  why do you have kj wright cove
ringgronk that s what happens superbowl.

----- Summary of tweets for given Key moment for tim wright ----

superbowlxlix patriotsvsseahawks usmelltoast. Gronk beats kj wright 14
-7 in new england. Kj wright throws two touchdown passes to gronk.. No
t sure that kj wright was the best choice to be covering gronk there c
hancellor or even bobby wagoner might have been better superbowl.. Wha
t stupid defense to put k j wright on gronk with no help over top supe
rbowlxlix.

----- Summary of tweets for given Key moment for rob ninkovich ----

touchdown rob superbowlxlix.  touchdown patriots tom brady to rob gron
kowski with less than 30 seconds remaining in the half to go up 14 7 o
n seattle. rob gronk spike. patriots 14 7 seahawks gopats superbowl14
9.. i hate the patriots but i ll be damnned if rob gronkowski isn t el
ite af superbowlxlix patriots. qb tom brady connects with te rob gronk
owskj for a td 14 7 pats with under one minute left in q2 superbowl..
superbowlxlix.  touchdown patriots rob gronkowski scored with 31 secon
ds left in the 1st half.

----- Summary of tweets for given Key moment for richard sherman ----

i want richard sherman s baby birth to be the halftimeshow that would
be performanceart enough to get a grant superbowl.. Tom brady is a sma
rt qb he knew sherman wasnt over there so he went deep superbowlxlix.
does anyone else think richardsherman looks like scar nfl sb49 superbo
wl.  right after seahawks move sherman to the slot superbowl.. Richard
sherman covers gronk the whole game putting a linebacker on gronk guar
anteed brady would get him the ball fail superbowlxlix. i think sherma
n is more hurt than advertised can t jam or he d have split out wide i
nstead of kj good cover lb superbowlxLix.

----- Summary of tweets for given Key moment for chris white ----

i don t know who to cheer for in the superbowl so i hope whichever tea
m has the handsomest white quarterback wins.. The patriots scored agai
n in superbowl 2015. The white team wore a red and blue uniform.. walt
er white returns in a superbowl ad breakingbad.  What s with all the o
ld white washed up actors making stupid commercials superbowlads super
bowlcommercials. get it white boy superbowlxlix gronkowski.
```



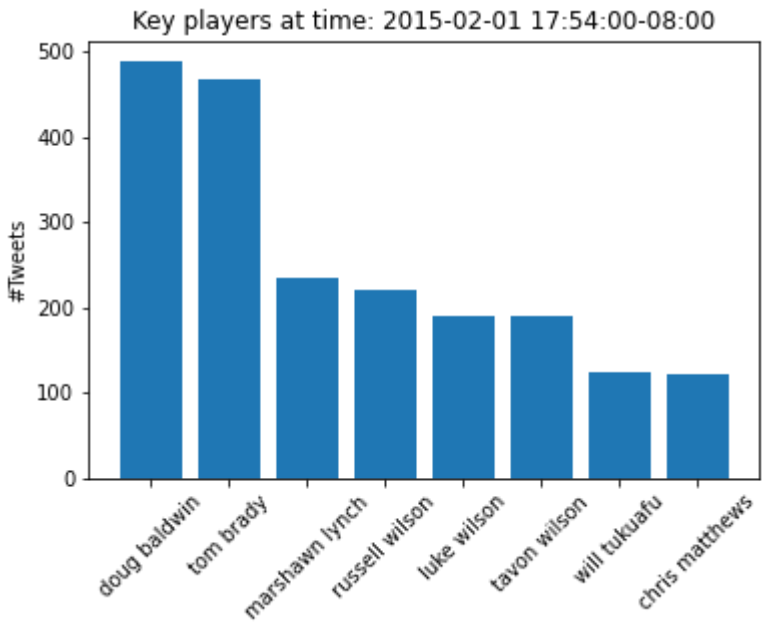

```
----- Summary of tweets for given Key moment for chris matthews ----

chris matthews is playing hardball superbowl sb49.. What a great game
superbowlxlix goseahawks russellwilson chrismatthewsiskillingit.. who
is chris matthews he ll be trending soon enough superbowl seahawks. ma
tthews makes a touchdown that ties it all up 14 14 superbowlxlix.

----- Summary of tweets for given Key moment for chris white ----

ok al and chris pump the breaks on the chris matthews hype superbowl..
chris matthews has been fantastic today superbowl.. chris matthews is
going to be drowning in pussy tonight superbowl. The patriots defense
look like little league of football superbowlxlix seahawks.

----- Summary of tweets for given Key moment for chris jones ----

What are the vegas odds of chris mathews being super bowl mvp superbow
l superbowlxlix.. Raiders have offered a five year deal to chris matth
ews superbowlxlix. If you go to superbowl.com, you can find out more a
bout chris Matthews amp amp.. i m confused does chris collinsworth thi
nk the td wouldn t count if the clock expired superbowl.

----- Summary of tweets for given Key moment for will tukuafu ----

Gents and gents being well groomed will help you be less scary than so
me of these superbowl commercials.. So if i understand the superbowl a
ds right we re all bad parents and children will die unless we douse e
verything in coke.. This game is intense. The patriots will take it in
the end.

----- Summary of tweets for given Key moment for russell wilson ----

russell wilson is clutch superbowlxlix.. My new dream job would be to
follow russell wilson around superbowlxlix.. superbowlxlix.  katy perr
y show everyone why russell brand is a complete idiot for leaving you
halftimeshow. russell wilson throws a touchdown to chris matthews for
seattle tied at 14 at halftime.

----- Summary of tweets for given Key moment for tom brady ----

Old school tom brady vs new school russell wilson superbowl.. The game
is tied at 14-14 after the first half. Tom brady throws two touchdown
passes in the second half. After the game, brady looks deflated.. i m
rooting for the seahawks because i think it would be funny to see tomb
rady cry again superbowl.
```

```
----- Summary of tweets for given Key moment for luke wilson ----

superbowlsunday superbowi superbowlxlix patriotsvsseahawks usn.. patri
ots and seahawks tied at the half 14 14 wilson pulls off an amazing th
row 6s.. russell wilson to matthews with 2 seconds left gutsy call wit
h a big payoff superbowlxlix.  way to go wilson hell of a throw go sea
hawks superbowlxLix.  sooooo i vote russel wilson best passing game to
uchdown superbowl sb49 14 14 going into halftime. seattle que partidaz
o py arg superbowl xlix.

----- Summary of tweets for given Key moment for tavon wilson ----

The patriots and seahawks are tied at the half 14 14 wilson pulls off
an amazing throw 6s patriotsvsseahawks patsnation sb49 superbowl.. i v
ote russell wilson best passing game touchdown superbowl sb49 14 14 go
ing into halftime.. A nice pass from russell wilson to chris matthews
was the highlight of the game for the patriots.
```



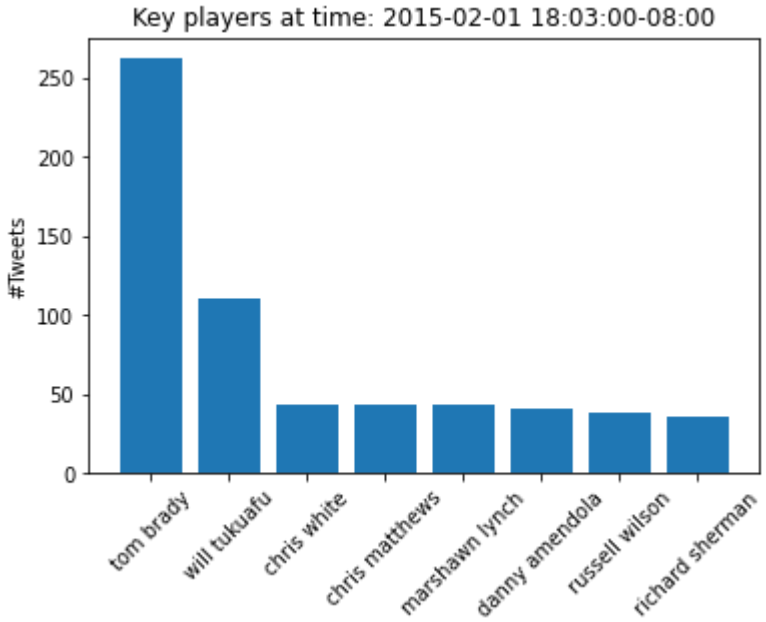

```
----- Summary of tweets for given Key moment for chris matthews ----

chris matthews just proves that seattle s scouting department is unrea
l superbowlxlix.. seahawks up 17 14 thanks to catch by chris matthews
superbowlxlix.. i mean chrismatthews who knew he had hands like that s
uperbowl.

----- Summary of tweets for given Key moment for chris white ----

rt chris matthews last 100 yard receiving game was on october 16 2010
with the kentucky wildcats. Chris matthews deserves everything he s go
t coming towards him rn superbowlxlix superbowlmVP.. Chris matthews is
a beast superbowl seahawkswin superbowlxlix kingvaith. chris matthews
having a david tyree game so far superbowl.. Was chris matthews stint
at foot locker before or after he was named cfl rookie of the year sb4
9 superbowl nfl seahawks patriots.

----- Summary of tweets for given Key moment for chris jones ----

The chris matthews story is coming to a theater near you in 2015 super
bowlxlix.. Chris matthews is on pace to be mVP superbowlxlix. chris ma
tthews was a cat for catch bbn sec ncaa nflfi superbowl seattleseahawk
s mVP.. My fantasyfootball mind is all ready going for next year with
```

```
this chris matthews dude 3 catches for 100 yards and a td superbowlxli
x.

----- Summary of tweets for given Key moment for will tukuafu ----

When will people learn superbowl sb49. when be performing at the super
bowl it will totally worth it.. superbowlxlix will always be my fav no
matter who comes after her.. My sponsor athletic will be giving away a
pair of my revolution combatpro1 shoes tomorrow.

----- Summary of tweets for given Key moment for tom brady ----

tom brady might be the captain of the patriots but i captain the black
bertha superbowl.. Tom brady has an exceptional jaw superbowlxlix.. Ju
st me and the cat watching the super bowl i bet even she doesn t like
tom brady superbowl.

----- Summary of tweets for given Key moment for marshawn lynch ----

Why would u do a draw when you have lynch at 1 yard mrsoapstar huh its
likeaforeignlanguagetome superbowl imhereforthecommercials.. why hasn
t had marshawn lynch in a commercial yet superbowlxlix.. i want the se
ahawks to win just so i can see another lynch interview superbowlxlix.
we don t care we got brady superbowl patriots.

----- Summary of tweets for given Key moment for russell wilson ----

katy perry's influence is being felt everywhere apparently superbowl.
russell wilson is wearing eyeliner.. russell wilson is showing a great
game superbowl. Michael collinsworth now officially has a wilson boner
superbowlxlix.. russell wilson better have sharted himself at halftime
otherwise he s a pussy for showering amp resuiting during halftime sup
erbowl. let s go seattleseahawks seahawks superbowl 2015.

----- Summary of tweets for given Key moment for luke wilson ----

russell wilson has had 183 games in a row where his team held the lead
at some point the unlikely streak continues superbowlxlix. katy perry
s influence is being felt everywhere apparently superbowl.  luke 02 39
am here and very awake superbowl.. russell wilson has now surpassed ku
rt warner for most super bowl passes completed by god superbowl.
```



```
----- Summary of tweets for given Key moment for doug baldwin ----

touchdown seahawks by doug baldwin pat is good 24 14 seattle.  baldwin
anota y seattle se ponen 10 arriba patriots 14 24 seahawks superbowlxl
ix.. doug baldwin scores for seattle with an end zone catch seahawks l
ead 24 14 superbowlxlix. baldwin just did the randy moss moon td celeb
ration joe buck is disgusted superbowl. baldwin got a stupid penalty a
fter the td by baldwin.. The official set a pick on that td to baldwin
no one can convince me that the seahawks earmed this lead superbowlxli
x.

----- Summary of tweets for given Key moment for tom brady ----

tom brady does it again with an interception superbowlxlix.. If your t
om brady now would probably be a good time to deflate some balls super
bowl.. Tom Brady throws two interceptions in Super Bowl XLIX.

----- Summary of tweets for given Key moment for marshawn lynch ----

i m just here to get a second ring seattleseahawks superbowlxlix 12thm
an nfl marshawnlynch. give my man lynch his rushing td.. marshawn lync
h is a beast marshawnlynch superbowlxlix.. marshawn lynch is a complet
e animal superbowlxlix sb49 seattleseahawks.

----- Summary of tweets for given Key moment for russell wilson ----

"Wow les seahawks sont formidables wilson et lynch le bulldozer superb
owlxlix". The seahawks lead the nfl superbowl.  touchdown wilson finds
baldwin in the end zone and the seahawks now lead 24 14.  touchdown se
attle pase de russell wilson para baldwin y los seahawks se adelantan
x 10 newengland 14 seattle 24 superbowlxlix.. russell wilson finds dou
g baldwin in the endzone to widen seattle s lead to 24 14 superbowlxli
x.

----- Summary of tweets for given Key moment for luke wilson ----

russell wilson mvptakeover back2back deflate brady and patriotcheater
s. superbowlxlix.. russell wilson played a game of chicken in the pock
et with chandler jones waited last sec slipped him superbowlxlix seaha
wkspatriots.. russel wilson is the next tom brady of super bowls super
bowl.

----- Summary of tweets for given Key moment for tavon wilson ----

russell wilson is having a great black history moment superbowl.. seah
awks lead 24 14 with 4 54 3q superbowlxlix seahawks.. i love how peopl
e said russell wilson wouldn t do anything in the nfl because he s too
small the guy is unreal superbowl.

----- Summary of tweets for given Key moment for will tukuafu ----

katie perry will ferrell looked great at half time superbowlxlix blade
sofglory.. Will nbc show us what was the celebration penalty for will
the nfl not let you show it superbowl makeithappy.. 13 unanswered is t
he only way i will get this forecast right superbowlxlix sb49.

----- Summary of tweets for given Key moment for chris matthews ----

chris matthews has been very crucial for us tonight keep it up chris s
uperbowlxlix seattleseahawks.. lynch and matthews are amazing superbow
```

lxlix.. After tonight if the seahawks win you know they re gonna put m
atthews in a foot locker commercial superbowlxlix.



----- Summary of tweets for given Key moment for tom brady ----

i m frustrated for tom brady right now cause what the hell superbowlxl
ix. let s discuss brady s ruby red slippers superbowl. waits for tom
Brady s bitchbaby cry spell to happen superbowl.. Tom brady can t seem
to handle his balls superbowlxlix.. tom brady having a shocking third
quarter sb49 superbowlxlix.

----- Summary of tweets for given Key moment for will tukuafu ----

i will be talkin superbowlcommercials ya know real hard hittin stuff..
should matthews win the mVP he will join teammate malcolm smith with t
he who are you award superbowl mVP seahawks chrismatthews nfl.. you wi
ll never have moves like these shark superbowl.

----- Summary of tweets for given Key moment for chris white ----

Chris matthews plays hardball for the seahawks.. i didn t get to see t
he walter white commercial but did see the toe fungus one superbowlxli
x.. Chris Collinsworth is clearly rooting against the pats throughout
the entire playoffs superbowl.

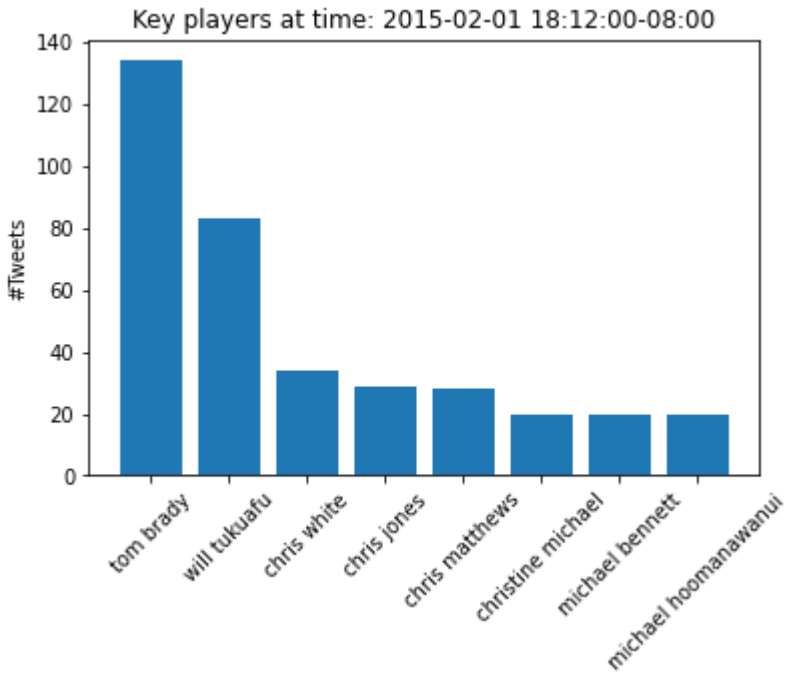
----- Summary of tweets for given Key moment for chris matthews ----

A proven commodity in michael bennett amp america s new superbowl hero
chris matthews a killer 1 2 combo for seahawks superbowl.. i ll be ok
with the seahawks winning if chrismatthews gets mVP superbowl.. i bet
chris matthews got marshawn lynch an employee discount on his gold sho
es superbowlxlix.

----- Summary of tweets for given Key moment for marshawn lynch ----

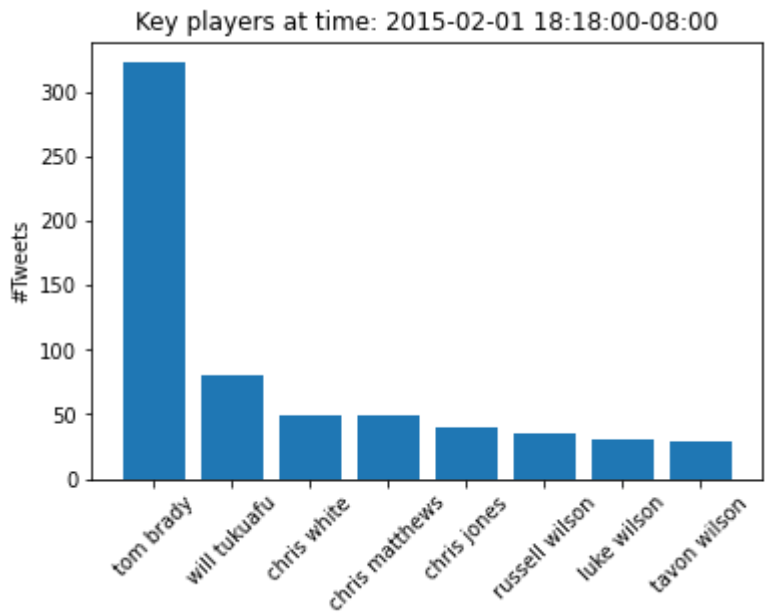
marshawn lynch didn t want to learn the words superbowlxlix.. do you t
hink marshawnlynch will get fined after the game superbowl seahawkswi
n.. here comes a very heavy dose of lynch look out superbowl beastmode

```
seavspats. marshawn lynch left marshawn right superbowlxlix.

----- Summary of tweets for given Key moment for danny amendola ----

There was no way to duck that hit danny lol superbowlxlix.. amendola j
ust dropped it like it was an important pass he needed to catch superb
owlxlix.. A deion branch would have made the catch and taken the hit a
mendola superbowl patsnation.

----- Summary of tweets for given Key moment for russell wilson ----

haaa now russell wilson should make it plain with a touchdown birdienu
mnum superbowl.. russell wilson doesn t need the stats he will do what
he needs to do to win superbowl.. Russell wilson is the future of the
superbowl, writes ravensfan.com. The future is now superbowlxlix. watc
h kearse and wilson make a big play here.

----- Summary of tweets for given Key moment for richard sherman ----

Something about pete carroll reminds me of richard gere superbowlxli
x.. Richard sherman is an absolute wanker superbowlxlix. no way sherma
n put down revis he weakest bragging the scoreboard collins worthy a f
ool.. i m sure richard sherman is a big kings of leon guy superbowl.
```



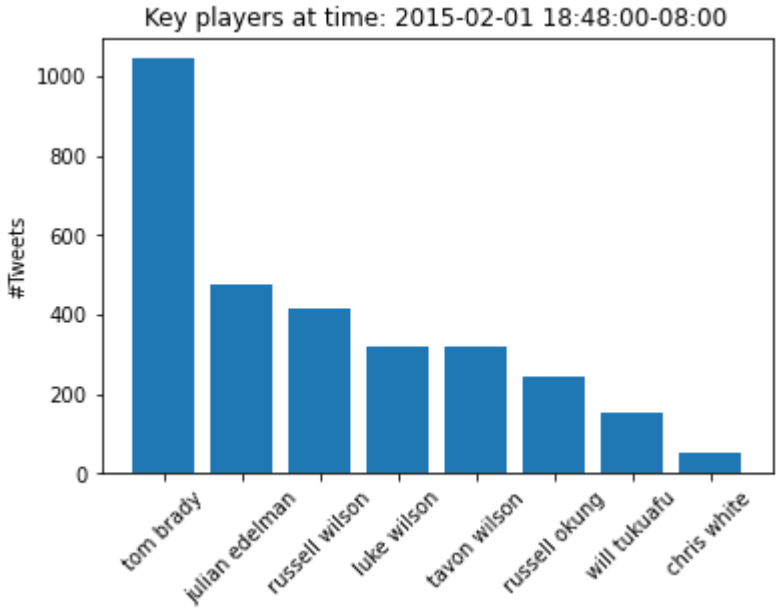

```
----- Summary of tweets for given Key moment for tom brady ----

When do we start the manning gt brady convo again vols superbowl.. tom
brady has no confidence in passing over 20 yards superbowlxlix.. Tom b
rady s crying and the hawks are flying superbowlxlix.

----- Summary of tweets for given Key moment for will tukuafu ----

Will the real katy perry stand up superbowl?. No matter what the resul
t the media will shit itself talking about how accurate tom terrific i
s.. Did superbowl just pay themselves 2 million for that ad and if it
results in a profit will they finally lose their non profit status.

----- Summary of tweets for given Key moment for chris white ----

All football aside i think land sharks cars driving sideways and emoti
```

```
onal white dads are the real winners today.. rt just spent the entire
super bowl halftime explaining to my white friends who missy elliot is
superbowl truedat funnyshit.. i m still going for chris jones and the
bgsu gopatriots superbowlxlix.

----- Summary of tweets for given Key moment for chris jones ----

How stoned is chris collinsworth right now superbowl.. The studio got
turned down by someone better when chris pratt is in a lead role juras
sicworld superbowlxlix.. chris matthews is having a solid game superbo
wlxlix.

----- Summary of tweets for given Key moment for chris matthews ----

How stoned is chris collinsworth right now superbowl.. Show me chris e
vans and chris pratt dear camera guy superbowl.. chris pratt is beatin
g chris evans 24 14 superbowl. chris mathews had 0 career catches befo
re superbowlxlix.

----- Summary of tweets for given Key moment for christine michael ---
-

"Michael amp sons be slayin deflategate superbowlads". michael and son
s best superbowl commercial ever or bestest.

----- Summary of tweets for given Key moment for michael bennett ----

"Michael amp sons be slayin deflategate superbowlads". michael and son
s best superbowl commercial ever or bestest.

----- Summary of tweets for given Key moment for michael hoomanawanui
----

"Michael amp sons be slayin deflategate superbowlads". michael and son
s best superbowl commercial ever or bestest.
```



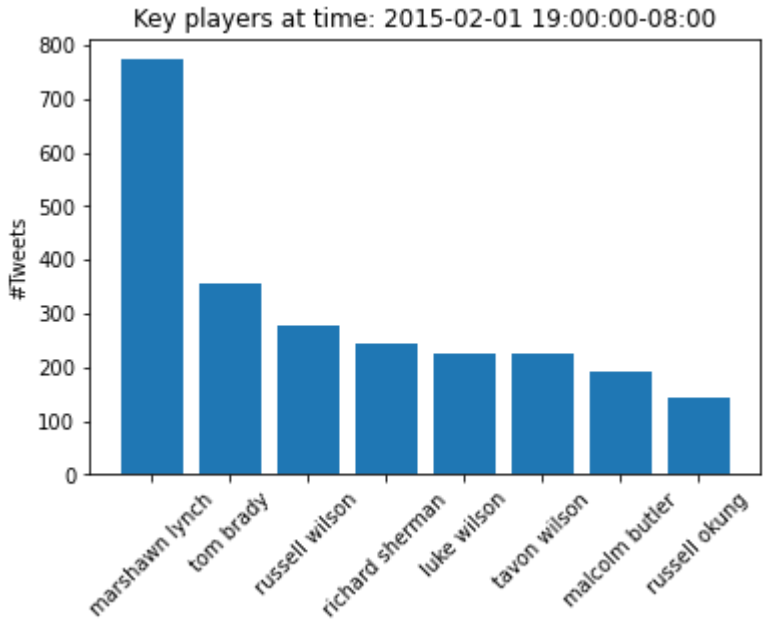

```
----- Summary of tweets for given Key moment for tom brady ----

tom brady is in ted 2 if he loses we know why superbowl superbowlcomme
rcials ted2.. tom brady s best pass in the second half was during that
ted 2 ad superbowl.  ok that ted commercial feat tom brady win superbo
```

```
wl.  maybe tombrady should spend a little less time acting superbowl s
bcommercials adbowl.. So so tom brady is griping about stuffed bears l
ike it was the afc championship game superbowlxlix.

----- Summary of tweets for given Key moment for will tukuafu ----

Russle wilson says the seahawks will win the superbowl.. No halftime s
how will ever top this one superbowl.. How we will remember superbowlx
lix the year the superglue and fanny packs became in style superbowl.

----- Summary of tweets for given Key moment for chris white ----

The bluebombers are getting no credit for this chris matthews guy star
tedfrombottom.. The most successful ad during superbowl hard ball with
chris mathews. walter white is back watch esurance breaking bad super
bowl ad superbowl sbcommercials.. A shout out to crispy chris matthews
from winn Winnipeg go bombers winn Winnipeg bluebombers nfl superbowlx
lix 12thman.

----- Summary of tweets for given Key moment for chris matthews ----

Hardball with chris mathews best line of the night in superbowl sb49..
chris matthews new title the footlocker amazing game so far superbowlx
lix superbowlsunday seahawkswin.. chris matthews just made the seahawk
s active roster 1 month ago.

----- Summary of tweets for given Key moment for chris jones ----

Chris matthews looking to become the 2nd former iowa barnstormer to wi
n super bowl mvp superbowlxlix.  chris matthews new title the footlock
er amazing game so far. thanks greeny i don t know what chris collinsw
orth talking about superbowl.. chris matthews is 3 3 for 100 yards on
passes thrown his way.. Hardball with chris mathews best line of the n
ight in superbowl sb49 rt see gt. sooo if win could chris matthews the
guy no one has heard of before today get mvp superbowlxlix.

----- Summary of tweets for given Key moment for russell wilson ----

The russell wilson brand loses the super bowl headlineoftheday sb49 ha
lftime superbowl.. russell wilson sacked mid throw sb49 superbowl.  da
mn sack wilson superbowlxlix.  if the seahawks were a fut they d be re
al sweaty wilson runs away from people all day superbowl fifa15.. In m
i view so far lynch and wilson are being the best mVP rusellwilson lyn
ch superbowl.

----- Summary of tweets for given Key moment for luke wilson ----

russell wilson taking a play out of christian hackenburg s playbook th
ere superbowlxlix.. russell wilson sacked on 3rd down take over after
the punt lead 24 14 superbowlxlix nevssea.. superbowlxlix.com is a wee
kly, off-the-wall look at the NFL. This week, we look at the latest fr
om the NFL. In the meantime, we take a look at some of the most memora
ble moments from the last 2 seasons of the NFL.

----- Summary of tweets for given Key moment for tavon wilson ----

russell wilson taking a play out of christian hackenburg s playbook th
ere superbowlxlix.. russell wilson sacked on 3rd down take over after
the punt lead 24 14 superbowlxlix nevssea.. If the seahawks were a fut
they d be real sweaty wilson will he beat the patriots yes he will so
n.
```



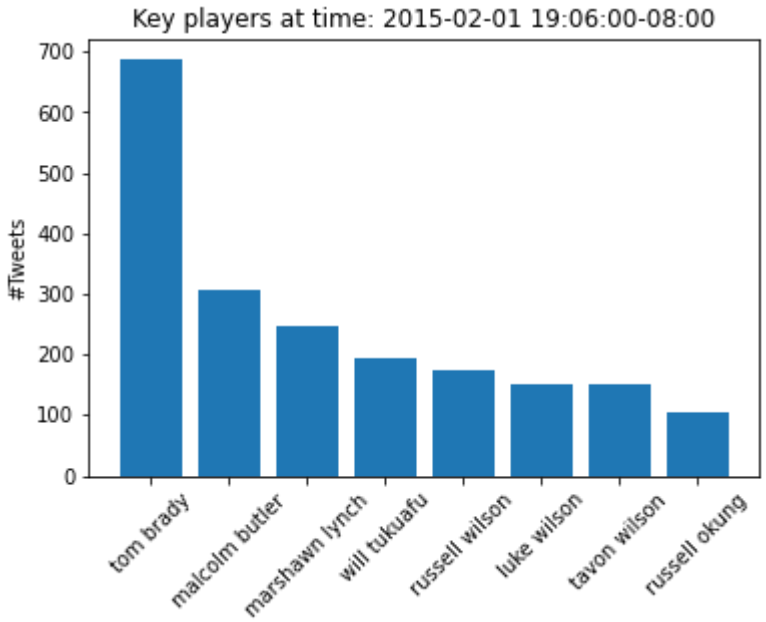

```
----- Summary of tweets for given Key moment for tom brady ----

Tom Brady leads the Patriots to a 28-24 win over the Seahawks. Patriot
s fans are chanting " tom freaking brady superbowlxlix". Julian Edelma
n touchdown puts patriots ahead 28-24. Tom brady throws another clutch
touchdown pass.. Two minutes to the endless glory superbowlxlix.

----- Summary of tweets for given Key moment for julian edelman ----

Julian edelman's touchdown pass to my favorite guy edelman majoranxiet
y letsgopats superbowlxlix.. tharold simon leaden footed as brady find
s edelman to make it 28 24 superbowl.. New England Patriots take 28-24
lead over Seattle Seahawks in 4th quarter. Julian Edelman catches a to
uchdown pass with 2 02 left to give pats the lead.

----- Summary of tweets for given Key moment for russell wilson ----

Super Bowl XLIX is on Sunday night at 8 p.m. ET. Super Bowl XLVIII is
on Monday night at 8:30 p.m., ET.. Can russell wilson throw seahawks d
own the field when he has to throw great game superbowlxlix.. 2 minute
s 2 seconds superbowl 2015superbowl superbowlxlix.

----- Summary of tweets for given Key moment for luke wilson ----

The patriots have to hit wilson if they want to win superbowl.. If bra
dy brady does it again wilson has 2 minutes to respond superbowlxlix..
Can russell wilson pull off a touchdown with 2 02 to go gohawks superb
owlxlix.

----- Summary of tweets for given Key moment for tavon wilson ----

i hope nobody s underestimating wilson here this is gonna be a game su
perbowlxlix.  if russell wilson doesn t pull this off does that mean g
```

```
od was rooting for the patriots superbowl.. time to see what russell w
ilson is made of sb49 superbowlxlix.. russell wilson can win the mvp a
nd the super bowl right here superbowlxlix.

----- Summary of tweets for given Key moment for russell okung ----

time to see what russell wilson is made of sb49 superbowlxlix.. russel
l wilson faces off against Manny Pacquiao in a backyard bout.. Russell
wilson gets the opportunity to go from very good to great right now su
perbowlxlix.

----- Summary of tweets for given Key moment for will tukuafu ----

The Super Bowl will be decided in the last three minutes of the game..
2 minutes to go who will win superbowl.. This will go down as one of t
he most forgettable superbowl games of all time.

----- Summary of tweets for given Key moment for chris white ----

i sense chris matthews is going to make a epic play on this drive shot
calling superbowlxlix gohawks. somewhere chris christie is being denie
d a high five superbowl.. 2-2 with 52 seconds left in superbowl amp ch
ris collinsworth brings up deflategate thats classless amp sad on nb
c.. Never count out the red white and blue also true in football super
bowlxlix.
```



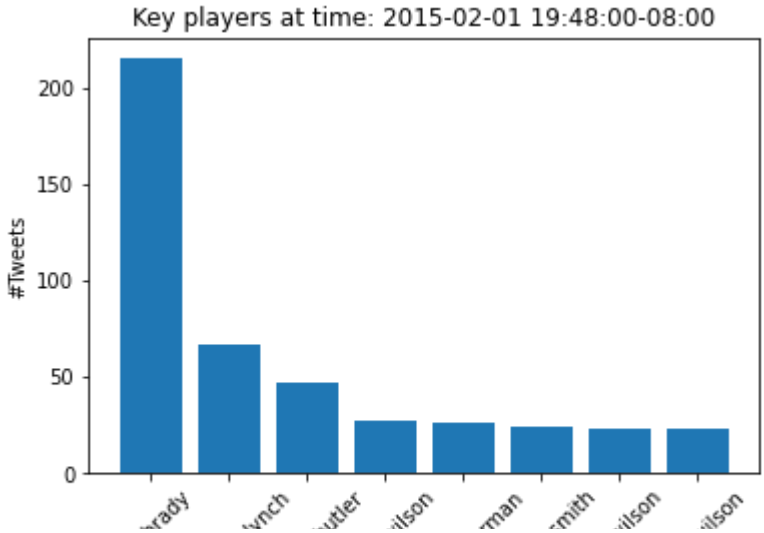

```
----- Summary of tweets for given Key moment for marshawn lynch ----

i can t believe they didn t give that ball to marshawn lynch superbow
l.. Why would you throw it instead of giving it to marshawn lynch supe
rbowl.. lynch is unstoppable brainfart.  i wouldn t have run lynch eit
her stupid call beastmode superbowlxlix.

----- Summary of tweets for given Key moment for tom brady ----

aww what s the matter tom you mad or nah haha superbowlxlix.. Tom brad
y wins his 4th super bowl ring.. Tom brady gets his 4th ring superbowl
xlix superbowlchampions. tom brady deserved that champ superbowlXlix.

----- Summary of tweets for given Key moment for russell wilson ----
```

```
There s a joke here about wilson and his god and where is he now?. daa
aaaaamnnnnnnnn russell wilson is the new tony romo chockedout superbow
lxlix.. If the offensive coordinator made that call he should be fired
if russell wilson made it there s always next year nfl superbowl.

----- Summary of tweets for given Key moment for richard sherman ----

" richard sherman s fave said it all why would you throw it superbowl.
woh seeing sherman s face hurt superbowl" "In the words of richardSher
man against richard Sherman you mad bro gopats superbowl". Richard she
rman is going to cry superbowl.. fuck you sherman fuck you lynch fuck
you wilson fuck you seacocks go superbowlxlix.

----- Summary of tweets for given Key moment for luke wilson ----

russell wilson was betting on the patriots superbowl patriotswin.. The
butler picks off wilson the patriots are gonna win the super bowl supe
rbowlxlix.. That play shows why wilson is not a top qb his team is onl
y in playoffs coz of defense and lynch overrated.

----- Summary of tweets for given Key moment for tavon wilson ----

russell wilson you will be sacrificed superbowlxlix.. Where s your god
now russell wilson superbowl.. wow russell wilson way to be the edith
crawley of superbowlxlix downtonabbey.

----- Summary of tweets for given Key moment for malcolm butler ----

Butler intercepting the pass that will ultimatly lead to the patriots
winning superbowl superbowl49 gopatriots. Butler is from vicksburg ms
superbowl nevssea sb49. Butler you hero superbowl.. Butler's intercept
ion gives patriots the game over.. The butler did it superbowlxlix.

----- Summary of tweets for given Key moment for russell okung ----

russell wilson cost seattle the superbowl.. lmfao that choke by russel
l wilson superbowl seavsne.. i wanna buy a bottle of russell wilson s
tears seahawks patriots seavsne superbowl.
```




```
----- Summary of tweets for given Key moment for tom brady ----
```

```
12 tom brady superbowlxlix 4superbowlrings. tom brady goat.. Sportworl
d and sportworld.com are the official public broadcasters of the europ
ean netherlands network.. "What a gaaame congrats patriots. What a sex
y tombrady"

----- Summary of tweets for given Key moment for malcolm butler ----

Butler won it for the guys man patriots superbowlxlix. malcom butler i
hatethepatriots seahawksaresorelosers superbowl. Butler you re the rea
l mvp superbowl patriots.. Newenglandpatriots win superbowl 49-17 over
patriots. malcom butler is the mvp of the superbowl isn't he? ifmalcom
butler gets mVP i m out superbowl patriots.. There s a ring on revisis
land thanks malcolm butler superbowlxlix nevssea.

----- Summary of tweets for given Key moment for marshawn lynch ----

you all know why we passed the ball marshawn lynch superbowl superbowl
xlix.. i can t believe they called a pass play with lynch in the backf
ield bad move carroll nfl superbowlxlix.. "Dumb seattle should ve rode
on the back of lynch all the way to superbowl victory" "How do you not
give the ball to lynch superbowl?"

----- Summary of tweets for given Key moment for will tukuafu ----

i expect it will be some party wherever rob gronkowsi ends up tonight
superbowl. yo watch it be a fumble lol i will die superbowl sb49.. Sup
er Bowl 49 will be remembered for bad play call superbowlxlix.. This w
ill go down as the worst call in super bowl history superbowlxlix.

----- Summary of tweets for given Key moment for russell wilson ----

Still boggles my mind why would you pass with lynch and wilson s feet
on the 1 superbowl sb49.. A new bold prediction: butler will get laid
by tom brady s wife tonight.. I guess god didn t want russell wilson t
o win this one god superbowl.

----- Summary of tweets for given Key moment for luke wilson ----

russell wilson threw the game away superbowlxlix.  tom brady paid wils
on off confirmed superbowl.  yes luck can only get you so far wilson s
uperbowl.. "Russel wilson com marshall lynch no time fazer um passe a
1 jarda da endzon decep o seahawks superbowl. pra que fazer isso". rus
sell wilson blew the game for seattle that is all superbowlxlix.

----- Summary of tweets for given Key moment for tavon wilson ----

The longest 18 seconds of russell wilson s life superbowl.. that last
play by russell wilson is going to haunt him for the rest of his lif
e.. i don t know why wilson didn t give the ball to lynch superbowl.

----- Summary of tweets for given Key moment for russell okung ----

The call was in your face russell superbowl.  and would you like your
superbowl well done or rare said russell wilson.. russell wilson and p
ete carroll need to be crucified for that garbage call.. tom brady to
russell wilson you mad bro superbowl.
```




```
----- Summary of tweets for given Key moment for tom brady ----

we are the champions come to next superbowl tom brady is a deus.. just
inbieber bullseye tombrady superbowl.. "This tom brady gif is everythi
ng sb49 superbowl" "Justinbieber bullseye tombrady superbowl"

----- Summary of tweets for given Key moment for marshawn lynch ----

We had fun tweeting with y all everybody thinks lynch should have gott
en the ball but hindsight is 20 20.. Had to repost marshawn superbowl2
015 seahawks.. marshawn lynch looking at pete carroll right now like s
eattleseahawks superbowl pats.

----- Summary of tweets for given Key moment for malcolm butler ----

malcolm butler won the superbowl for the patriots. The butler is a cla
ssy player.. Butler's interception won the superbowl for the patriots.
But dez b catch didn t count huh karma bitches patriotswin superbowlxl
ix patriots malcolmbutler.. The butler did it interception of the deca
de superbowl superbowlxlix gopatriots patriotsvsseahawks. destiny call
ed malcolm butler s name and the man did not blink superbowl patriotsw
in.

----- Summary of tweets for given Key moment for russell wilson ----

i wish carroll would keep it 100 and say i wanted wilson to be the her
o instead of lynch the nfl did too superbowlxlix.. i was the one tweet
ing snarky things when pete carroll called that pass and russell wilso
n threw it superbowl sb49.. What a superbowl what a superbowl gotta lo
ve gronk just my opinion.

----- Summary of tweets for given Key moment for richard sherman ----

How many rings does tom have hahaha is sherman bradys sponsor patriots
win?. so a picture is worth a thousand words superbowl madatmebro rich
ardsherman.. Thanks richard nepatriots superbowl. sherman s face tho l
ol superbowl.

----- Summary of tweets for given Key moment for malcolm smith ----

The best superbowl commercial when uwa tiger malcolm butler won the ga
me.. 12th man malcolm butler superbowl.. malcolm butler will never hav
e to pay for a meal in new england again.
```

```
----- Summary of tweets for given Key moment for luke wilson ----

i wish carroll would keep it 100 and say i wanted wilson to be the her
o instead of lynch the nfl did too superbowlxlix.. russell wilson is a
smart guy why not change that play at the line your throwing to ricard
o lockette not a real wr superbowl.. i hear god smitted russell wilson
today he s a fickle piece of work superbowl.

----- Summary of tweets for given Key moment for tavon wilson ----

i wish carroll would keep it 100 and say i wanted wilson to be the her
o instead of lynch the nfl did too superbowlxlix.. russell wilson is a
smart guy why not change that play at the line your throwing to ricard
o lockette not a real wr superbowl.. i hear god smitted russell wilson
today he s a fickle piece of work superbowl.
```

Part 5 Fanbase Prediction

For this part I have predicted the state from which a tweet has been made. Only two states : Washington and Massachusetts are considered. The hashtag data used for this is the #superbowl as it contains all the tweets from supporters, rivals and neutral people.

While loading the tweets and data, tweets which are having location tweet['user']['location'] are only considered. The results were further filtered by taking only tweets with language as 'english'.

The location given is not per any standard norm and thus, to get the states to which a location belongs, I used a python library "geonamecache", from which we got the state corresponding to a city and also have written a simple filter to determine the state given the location.

Only WA and MA states are kept.

We are left with total :

MA - 16574
WA - 15784, datapoints.

Out of these 20% were kept aside for testing and 80% are used for training.

To build the features from the text, I used TFIDF representation of each word in a tweet. A pipeline with stopword removal, lemmatization, digit removal was used, given the tweets have already gone through a level of cleaning earlier as they were loaded from file.

Thus there are 6340 features per data point which is too much given low number of samples.

I used TruncatedSVD to reduce the number of features. Explained variance plot helped in deciding the number to choose. We kept 200 components thus 6340 features are reduced to 200 components and 40 % of variance is retained.

The classes are evenly distributed and there is not such class imbalance.

Two classifiers are used to predict the two classes.

Logistic regression.
Support Vector Classifier.

Search Params for Logistic Regression:
C: [10^-4, 10^4]
penalty: [l1, l2, elasticnet] , elasticnet is a combination of l1 and l2

Search Params for SVC:
C: [1, 10, 100, 1000, 0.1, 0.01, 0.001]
penalty: [l1, l2] , elasticnet is a combination of l1 and l2
loss: ['hinge', 'squared_hinge']

GridSearchCV with a k-fold of 5 is used to search the hyperparameters.

Best Logistic Regression had params: l1, C = 1.0
Best test accuracy: 0.7456736711990112

Scores:
accuracy: 0.7469097651421508
recall: 0.7421999372997606
precision: 0.7719416369220731
f1_score: 0.7383590910127489

Best SVC had params: l2, C = 100, loss = 'squared_hinge'
Best test accuracy: 0.7469097651421508

Scores:

accuracy: 0.7469097651421508
recall: 0.7421999372997606
precision: 0.7719416369220731
f1_score: 0.7383590910127489

The confusion matrices and ROC with AUC for both of the classifiers are plotted below.
The AUC for both of them is 0.84 which is average performance.
Also there are a lot of misclassifications but SVC seems to performs slightly better compared to Logistic Regression.

All #superbowl tweets are loaded but only considering english ones as they can be analysed and segregated.

In [352]:

```
superbowlEN = superbowl[superbowl['lang'] == 'en']
```

In [355]:

```
superbowlEN['lang'].value_counts()
```

Out[355]:

```
en    949540
Name: lang, dtype: int64
```

Total dataset : 949540

In [364]:

```
gc = geonamescache.GeonamesCache()
```

In [385]:

```
# 'countrycode': 'US'
# 'admin1code': 'MA'
#  'admin1code': 'WA'
#  countrycode': 'US'

cityKeys = list(gc.get_cities().keys())

cities = gc.get_cities()

WACities = []
MACities = []

for key in cityKeys:
    city = cities[key]
    if city['countrycode'] == 'US' and city['admin1code'] == 'MA':
        MACities.append(city['name'].lower())
    elif city['countrycode'] == 'US' and city['admin1code'] == 'WA':
        WACities.append(city['name'].lower())
```

In [413]:

```
def isWACity(loc):
    if loc.lower() in WACities:
        return 'WA'
    if " WA" in loc or ("Washington" in loc and "DC" not in loc and "D.C." not in lo
        return 'WA'
    return False

def isMACity(location):
    if location.lower() in MACities:
        return 'MA'
    if " MA" in location or "Massachusetts" in location:
        return 'MA'
    return False

def isWAMA(location):
    x = isWACity(location)
    y = isMACity(location)
    return x or y
```

In [416]:

```
superbowlEN['location'] = superbowlEN['location'].apply(lambda x: isWAMA(x))
```

```
/var/folders/vz/klhnj80n0ldfrc97mb312d9c0000gn/T/ipykernel_99659/40550
58753.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  superbowlEN['location'] = superbowlEN['location'].apply(lambda x: is
WAMA(x))
```

In [423]:

```
superbowlEN = superbowlEN[(superbowlEN['location'] == 'WA') | (superbowlEN['location
```

In [472]:

```
superbowlEN['location'].value_counts()
```

Out[472]:

```
MA    16574
WA    15784
Name: location, dtype: int64
```

In [473]:

```
def getSVD(n_comp, data):
    SVD = TruncatedSVD(n_components=n_comp, random_state=42)
    SVD.fit(data)
    return SVD
```

In [477]:

```
totalData = superbowlEN[['tweet', 'location']]
```

In [517]:

```
totalData['location'].value_counts()
```

Out[517]:

```
MA    16574
WA    15784
Name: location, dtype: int64
```

Classes are evenly distributed. No class imbalance

In [478]:

```
x_train, x_test, y_train, y_test = train_test_split(totalData['tweet'],
                                                    totalData['location'], test_size


print("--------- Train data Size: {}, Test data Size: {} ---------".format(len(x_tra
```

```
--------- Train data Size: 25886, Test data Size: 6472 ---------
```

In [479]:

```
CV = vectorizer2(min_df=3)

featurePipeline = Pipeline([
    ('count', CV),
    ('tfidf', TfidfTransformer(smooth_idf=True, use_idf=True))
]).fit(x_train)
```

In [480]:

```
train_tfidf = featurePipeline.transform(x_train)
test_tfidf = featurePipeline.transform(x_test)
```

In [481]:

```
features = len(featurePipeline['count'].get_feature_names_out())
print("Total features: ", features)
```

```
Total features:  6340
```

In [482]:

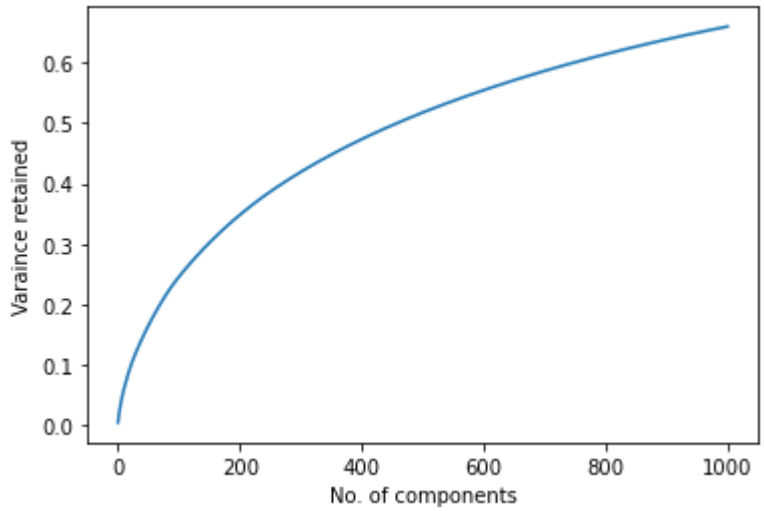
```
n_components = 1000
SVD = getSVD(n_components, train_tfidf)
```

In [483]:

```
plt.plot(np.arange(1000), np.cumsum(SVD.explained_variance_ratio_))
plt.xlabel("No. of components")
plt.ylabel("Varaince retained")
```

Out[483]:

```
Text(0, 0.5, 'Varaince retained')
```



Choosing n_components for SVD = 200

In [484]:

```
n_components = 200
SVD = getSVD(n_components, train_tfidf)
```

In [485]:

```
train_tfidf = SVD.transform(train_tfidf)
test_tfidf = SVD.transform(test_tfidf)
```

In [491]:

```
gridSearchLR = Pipeline([
    ('clf', LogisticRegression(penalty='l1', C = 0.1, random_state=42, solver='saga'
])

params_LR = {
    'clf__C': [10.0**x for x in np.arange(-4,4)],
    'clf__penalty': ['l1', 'l2', 'elasticnet']
}
```

In [492]:

```
grid_LR = GridSearchCV(gridSearchLR, param_grid=params_LR, cv=5, verbose=0, n_jobs=-
                       scoring='accuracy').fit(train_tfidf, y_train)
```

In [495]:

```
LRdf = pd.DataFrame(grid_LR.cv_results_)
```

In [497]:

```
LRdf.sort_values(by=['mean_test_score'], ascending=False).head(5)
```

Out[497]:

2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
0.743288	0.737686	0.750048	0.745886	0.005033	1
0.740583	0.739231	0.748503	0.745190	0.004339	2
0.743867	0.739231	0.748117	0.745113	0.003330	3
0.744253	0.739424	0.746571	0.744611	0.002734	4
0.744060	0.739038	0.746765	0.744495	0.002888	5

Using best logistic regression model

In [505]:

```
clfLR = LogisticRegression(penalty='l1', C = 1.0, random_state=42,
                           solver='saga')

predsLR = clfLR.fit(train_tfidf, y_train).predict(test_tfidf)
```

In [506]:

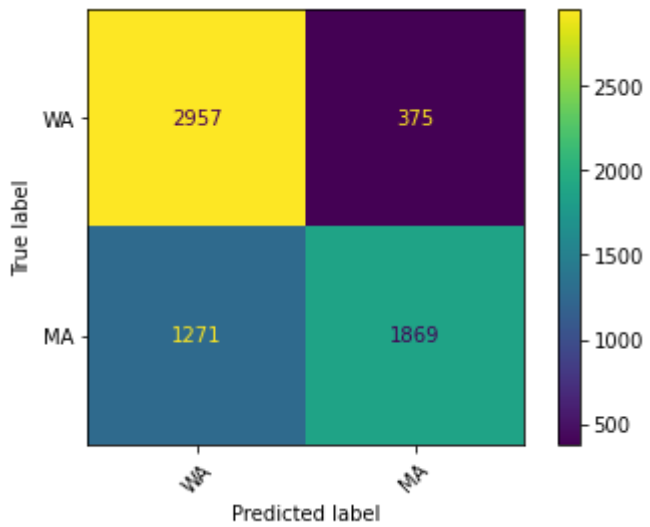
```
print("Test accuracy Logistic regression: ", accuracy_score(y_test, predsLR))
```

```
Test accuracy Logistic regression:  0.7456736711990112
```

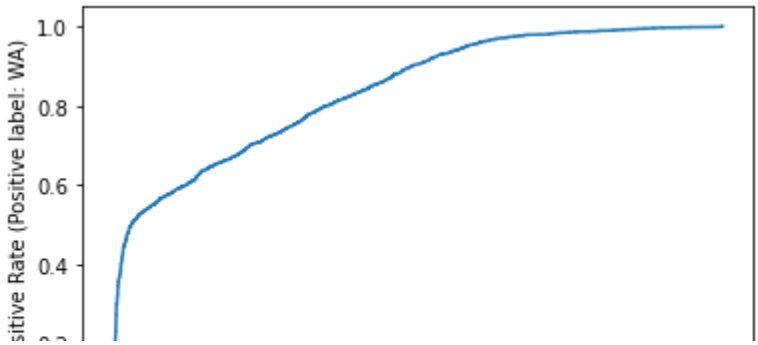
In [514]:

```
print("\n---------Scores for Logistic Classifier Classifier-------")
print('accuracy:', accuracy_score(y_test, predsLR))
print('recall:', recall_score(y_test, predsLR, average='macro'))
print('precision:', precision_score(y_test, predsLR, average='macro'))
print('f1_score:', f1_score(y_test, predsLR, average='macro'))

print("\n---------Confusion Matrix for Logistic Classifier-------")
plot_confusion_matrix(clfLR, test_tfidf, y_test, display_labels=['WA', 'MA'], xticks
plt.show()

metrics.plot_roc_curve(clfLR, test_tfidf, y_test)
plt.show()
```

```
---------Scores for Logistic Classifier Classifier-------
accuracy: 0.7456736711990112
recall: 0.7413389559645515
precision: 0.766136376284409
f1_score: 0.7382772392430639

---------Confusion Matrix for Logistic Classifier-------

/Users/gauravsingh/miniforge3/envs/tf_metal/lib/python3.8/site-package
s/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confus
ion_matrix is deprecated; Function `plot_confusion_matrix` is deprecat
ed in 1.0 and will be removed in 1.2. Use one of the class methods: Co
nfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
```




```
/Users/gauravsingh/miniforge3/envs/tf_metal/lib/python3.8/site-package
s/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_roc_cu
rve is deprecated; Function :func:`plot_roc_curve` is deprecated in 1.
0 and will be removed in 1.2. Use one of the class methods: :meth:`skl
earn.metric.RocCurveDisplay.from_predictions` or :meth:`sklearn.metri
c.RocCurveDisplay.from_estimator`.
  warnings.warn(msg, category=FutureWarning)
```



In [524]:

```
gridSearchSVC = Pipeline([
    ('clf', LinearSVC(penalty='l1', C = 1.0, loss='hinge', random_state=42))
])

params_SVC = {
    'clf__penalty': ['l1', 'l2'],
    'clf__C': [1, 10, 100, 1000, 0.1, 0.01, 0.001],
    'clf__loss': ['hinge', 'squared_hinge']
}
```

In [525]:

```
grid_SVC = GridSearchCV(gridSearchSVC, param_grid=params_SVC, cv=5, verbose=0, n_jok
                        scoring='accuracy').fit(train_tfidf, y_train)
```

```
huggingface/tokenizers: The current process just got forked, after par
allelism has already been used. Disabling parallelism to avoid deadloc
ks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELI
SM=(true | false)
huggingface/tokenizers: The current process just got forked, after par
allelism has already been used. Disabling parallelism to avoid deadloc
ks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELI
SM=(true | false)
huggingface/tokenizers: The current process just got forked, after par
allelism has already been used. Disabling parallelism to avoid deadloc
ks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELI
SM=(true | false)
huggingface/tokenizers: The current process just got forked, after par
allelism has already been used. Disabling parallelism to avoid deadloc
ks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELI
SM=(true | false)
huggingface/tokenizers: The current process just got forked, after par
allelism has already been used. Disabling parallelism to avoid deadloc
ks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELI
SM=(true | false)
huggingface/tokenizers: The current process just got forked, after par
allelism has already been used. Disabling parallelism to avoid deadloc
ks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELI
SM=(true | false)
huggingface/tokenizers: The current process just got forked, after par
allelism has already been used. Disabling parallelism to avoid deadloc
ks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELI
SM=(true | false)
huggingface/tokenizers: The current process just got forked, after par
allelism has already been used. Disabling parallelism to avoid deadloc
ks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELI
SM=(true | false)
```

In [526]:

```
svcDf = pd.DataFrame(grid_SVC.cv_results_)
```

In [533]:

```
svcDf.sort_values(by=['mean_test_score'], ascending=False).head(5)
```

Out[533]:

2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
0.743288	0.738459	0.748503	0.745345	0.004041	1
0.739618	0.739231	0.751014	0.745190	0.004916	2
0.740390	0.739231	0.747537	0.744495	0.003865	3
0.741163	0.739038	0.747151	0.744379	0.003563	4
0.738845	0.739811	0.744833	0.743993	0.004029	5

In [534]:

```
clfSVC = LinearSVC(penalty='l2', C = 100, loss='squared_hinge', random_state=42)

predsSVC = clfSVC.fit(train_tfidf, y_train).predict(test_tfidf)
```

```
/Users/gauravsingh/miniforge3/envs/tf_metal/lib/python3.8/site-package
s/sklearn/svm/_base.py:1206: ConvergenceWarning: Liblinear failed to c
onverge, increase the number of iterations.
  warnings.warn(
```

In [535]:

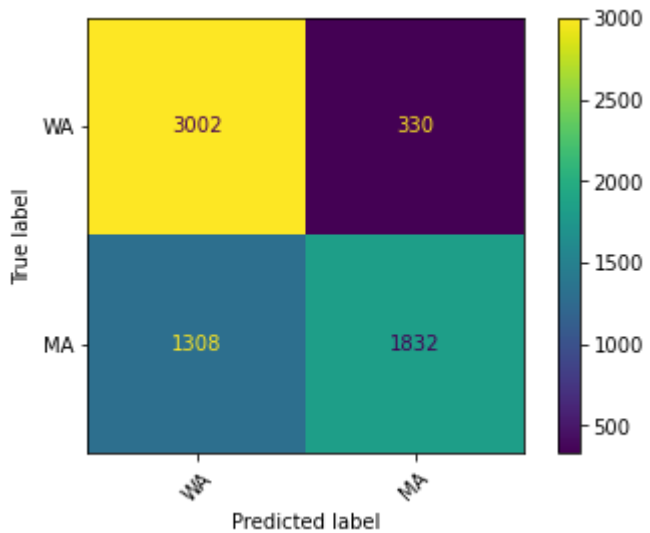
```
print("Test accuracy SVC classifier: ", accuracy_score(y_test, predsSVC))
```

```
Test accuracy SVC classifier:  0.7469097651421508
```

In [536]:

```
print("\n---------Scores for Logistic Classifier Classifier-------")
print('accuracy:', accuracy_score(y_test, predsSVC))
print('recall:', recall_score(y_test, predsSVC, average='macro'))
print('precision:', precision_score(y_test, predsSVC, average='macro'))
print('f1_score:', f1_score(y_test, predsSVC, average='macro'))

print("\n---------Confusion Matrix for Logistic Classifier-------")
plot_confusion_matrix(clfSVC, test_tfidf, y_test, display_labels=['WA', 'MA'], xtick
plt.show()

metrics.plot_roc_curve(clfSVC, test_tfidf, y_test)
plt.show()
```

```
---------Scores for Logistic Classifier Classifier-------
accuracy: 0.7469097651421508
recall: 0.7421999372997606
precision: 0.7719416369220731
f1_score: 0.7383590910127489

---------Confusion Matrix for Logistic Classifier-------

/Users/gauravsingh/miniforge3/envs/tf_metal/lib/python3.8/site-package
s/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confus
ion_matrix is deprecated; Function `plot_confusion_matrix` is deprecat
ed in 1.0 and will be removed in 1.2. Use one of the class methods: Co
nfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
```




```
/Users/gauravsingh/miniforge3/envs/tf_metal/lib/python3.8/site-package
s/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_roc_cu
```

```
rve is deprecated; Function :func:`plot_roc_curve` is deprecated in 1.
0 and will be removed in 1.2. Use one of the class methods: :meth:`skl
earn.metric.RocCurveDisplay.from_predictions` or :meth:`sklearn.metri
c.RocCurveDisplay.from_estimator`.
  warnings.warn(msg, category=FutureWarning)
```