Problem Statement

Demand Forecast is one of the key tasks in Supply Chain and Retail Domain in general. It is key in effective operation and optimization of retail supply chain. Effectively solving this problem requires knowledge about a wide range of tricks in Data Sciences and good understanding of ensemble techniques. You are required to predict sales for each Store-Day level for one month. All the features will be provided and actual sales that happened during that month will also be provided for model evaluation.

Week 1 Task

In [4]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set_style("whitegrid")
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error,mean_squared_error
from sklearn.preprocessing import LabelEncoder
import tensorflow as tf
import math
import warnings
warnings.filterwarnings('ignore')
from statsmodels.tsa.stattools import adfuller,acf,pacf
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima_model import ARIMA
from keras.models import Sequential
from keras.layers import Dense, LSTM
from keras import layers
from tensorflow.keras.optimizers import Adam,RMSprop,SGD,Adagrad
from keras.models import load_model
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import Ridge
from sklearn.ensemble import AdaBoostRegressor
```

In [12]:

```
# Loading the dataset
## Exploratory Data Analysis (EDA) and Linear Regression:
```

In [9]:

```
train_data = pd.read_csv(r'train_data.csv')
train_data.Date = pd.to_datetime(train_data.Date)
train_data.head()
```

Out[9]:

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	2	2015-06-30	5735	568	1	1	0	0
1	2	2	2015-06-30	9863	877	1	1	0	0
2	3	2	2015-06-30	13261	1072	1	1	0	1
3	4	2	2015-06-30	13106	1488	1	1	0	0
4	5	2	2015-06-30	6635	645	1	1	0	0

In [13]:

```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 982644 entries, 0 to 982643
Data columns (total 9 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Store          982644 non-null  int64
 1   DayOfWeek      982644 non-null  int64
 2   Date           982644 non-null  datetime64[ns]
 3   Sales          982644 non-null  int64
 4   Customers      982644 non-null  int64
 5   Open           982644 non-null  int64
 6   Promo          982644 non-null  int64
 7   StateHoliday   982644 non-null  object
 8   SchoolHoliday  982644 non-null  int64
dtypes: datetime64[ns](1), int64(7), object(1)
memory usage: 67.5+ MB
```

In [15]:

```
#1. Transform the variables by using data manipulation techniques like, One-Hot Encodin
# checking unique values
```

In [16]:

```
train_data.StateHoliday.unique()
```

Out[16]:

```
array(['0', 'a', 'b', 'c', 0], dtype=object)
```

In [18]:

```
train_data.loc[train_data.StateHoliday==0,'StateHoliday'] = '0'
```

Use One-Hot Encoding to convert this column

In [19]:

```
labelencoder= LabelEncoder()
train_data.StateHoliday = labelencoder.fit_transform(train_data['StateHoliday'])
```

In [20]:

```
train_data.StateHoliday.unique()
```

Out[20]:

```
array([0, 1, 2, 3])
```

2. Perform an EDA (Exploratory Data Analysis) to see the impact of variables over Sales.

In [21]:
```
train_data.isna().sum() # checking null values
```

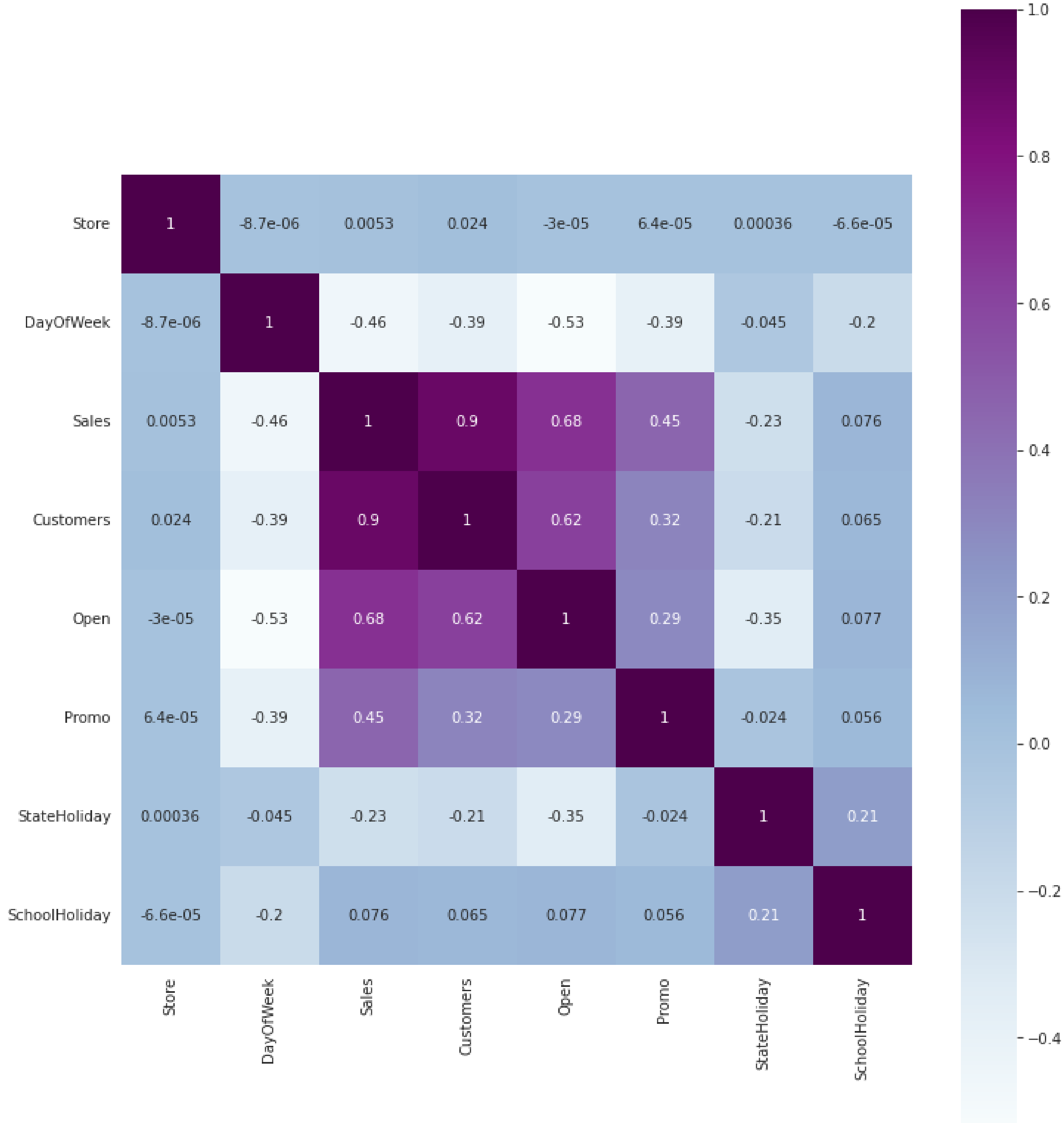
Out[21]:
```
Store            0
DayOfWeek        0
Date             0
Sales            0
Customers        0
Open             0
Promo            0
StateHoliday     0
SchoolHoliday    0
dtype: int64
```

There is no null value

In [9]:
```
train_data =train_data[~train_data.isin([np.nan, np.inf, -np.inf]).any(1)]
```

In [10]:
```
# check the correlation between variables
plt.figure(figsize=(12,14))
sns.heatmap(train_data.corr(),annot=True, square=True,cmap="BuPu")
```

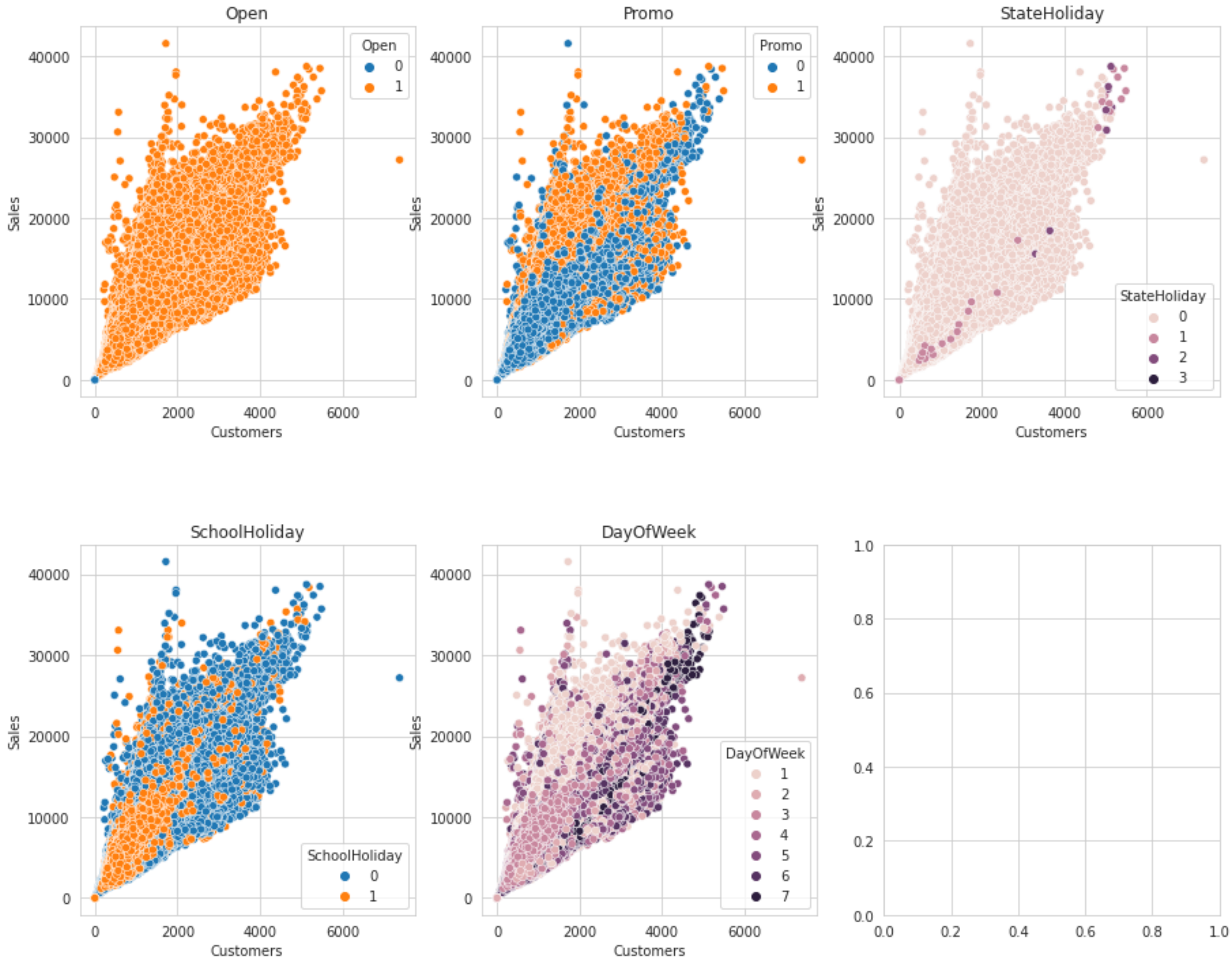
Out[10]:
```
<AxesSubplot:>
```



The above heatmap shows that there is a very good correlation between 'sales' and 'customer'. '*Sales' is also related to 'Open' and 'promo*'. 'Sales' is negatively correlated with 'StateHoliday' and 'DayOfWeek

In [11]:

```
fig, axs = plt.subplots(2,3, figsize=(15,12))
fig.subplots_adjust(hspace=0.4)
axs=axs.ravel()
A_list = ['Open', 'Promo','StateHoliday', 'SchoolHoliday','DayOfWeek']
i=0
for col in A_list:
    sns.scatterplot(train_data.Customers,train_data.Sales,hue=train_data[col],ax=axs[i]
    axs[i].set_title(col)
    i+=1
fig.show()
```



From the above EDA analysis we can conclude as follow:

- ZERO Sales when shop is closed.
- HIGH Sales when promo codes and discount is available.
- Very LOW or Very HIGH Sales on State Holidays.
- HIGH Sales when schools are open.

1. Apply Linear Regression to predict the forecast and evaluate different accuracy metrices like RMSE (Root Mean Squared Error) and MAE(Mean Absolute Error) and determine which metric makes more sense. Can there be a better accuracy metric?

In [12]:

```
##Total number of stores
n_shops = train_data.Store.nunique()
n_shops
```

Out[12]: 1115

In [13]:

```
original_data = train_data.copy()
train_data.drop('Date', axis=1, inplace=True)
```

In [14]:

```
train_data.head()
```

Out[14]:

	Store	DayOfWeek	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	2	5735	568	1	1	0	0
1	2	2	9863	877	1	1	0	0
2	3	2	13261	1072	1	1	0	1
3	4	2	13106	1488	1	1	0	0
4	5	2	6635	645	1	1	0	0

In [15]:

```
y=np.array(train_data['Sales'])
x=np.array(train_data.drop('Sales',axis=1))
```

In [16]:

```
lr = LinearRegression(normalize=True)
```

In [17]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state = 100, test_size=0.3)
```

In [18]:

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(687850, 7)
(294794, 7)
(687850,)
(294794,)
```

In [19]:

```
 lr.fit(x_train,y_train)
```

Out[19]:

```
LinearRegression(normalize=True)
```

In [20]:

```
y_pred = lr.predict(x_test)
```

In [21]:

```
def error_cal(y_true,y_pred):
    RMSE = math.sqrt(mean_squared_error(y_true,y_pred))
    MAE = mean_absolute_error(y_true,y_pred)
    return RMSE,MAE
```

In [22]:

```
all_store_lr = error_cal(y_test,y_pred)
all_store_lr
```

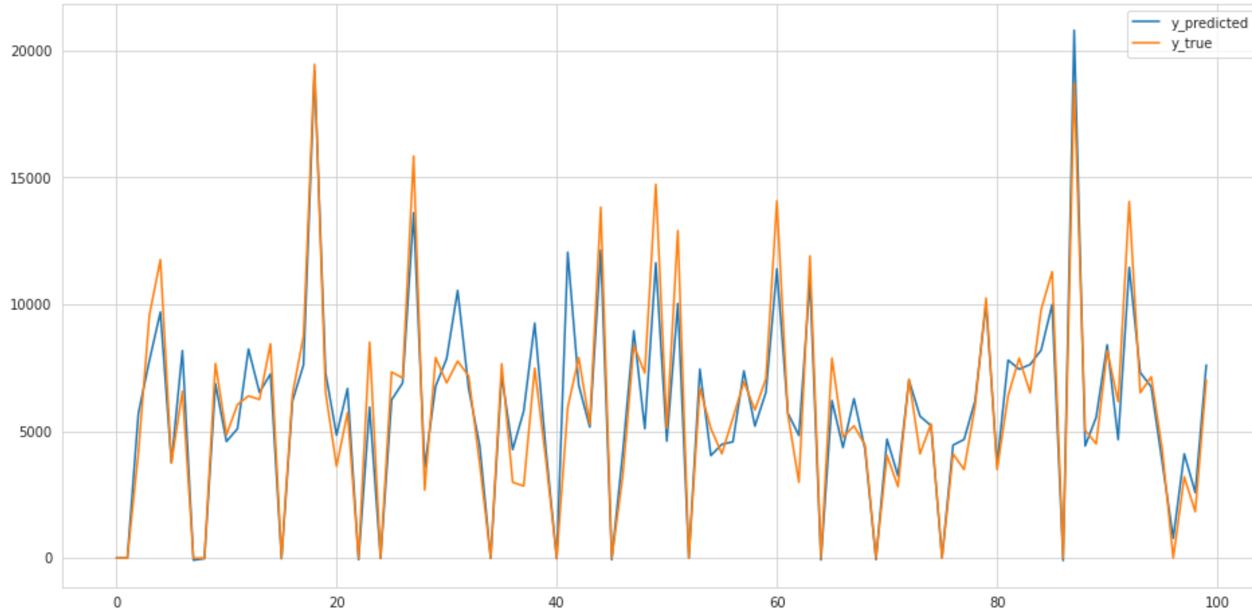
Out[22]:

```
(1471.475458119501, 978.5123617111675)
```

In [23]:

```
plt.figure(figsize=(16,8))
plt.plot(y_pred[:100],label = 'y_predicted')
plt.plot(y_test[:100], label = 'y_true')
```

```
plt.legend()
plt.show()
```



b) Train separate model for each store.

In [24]:

```
def model_single_store(x,y):
    lr = LinearRegression(normalize=True)
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=42
    lr.fit(x_train,y_train)
    y_pred = lr.predict(x_test)
    return y_test,y_pred
```

In [25]:

```
stores = [1,2,3,4,5,6,7,8,9,10,11]
```

In [26]:

```
RMSE_array_lr = []
MAE_array_lr=[]
for store in range(1,12):
    data = train_data[train_data.Store==store]
    data.drop('Store',axis=1,inplace=True)
    y=np.array(data['Sales'])
    x=np.array(data.drop('Sales',axis=1))
    y_true,y_pred = model_single_store(x,y)
    RMSE_1,MAE_1 = error_cal(y_true,y_pred)
    RMSE_array_lr.append(RMSE_1)
    MAE_array_lr.append(MAE_1)
```

In [27]:

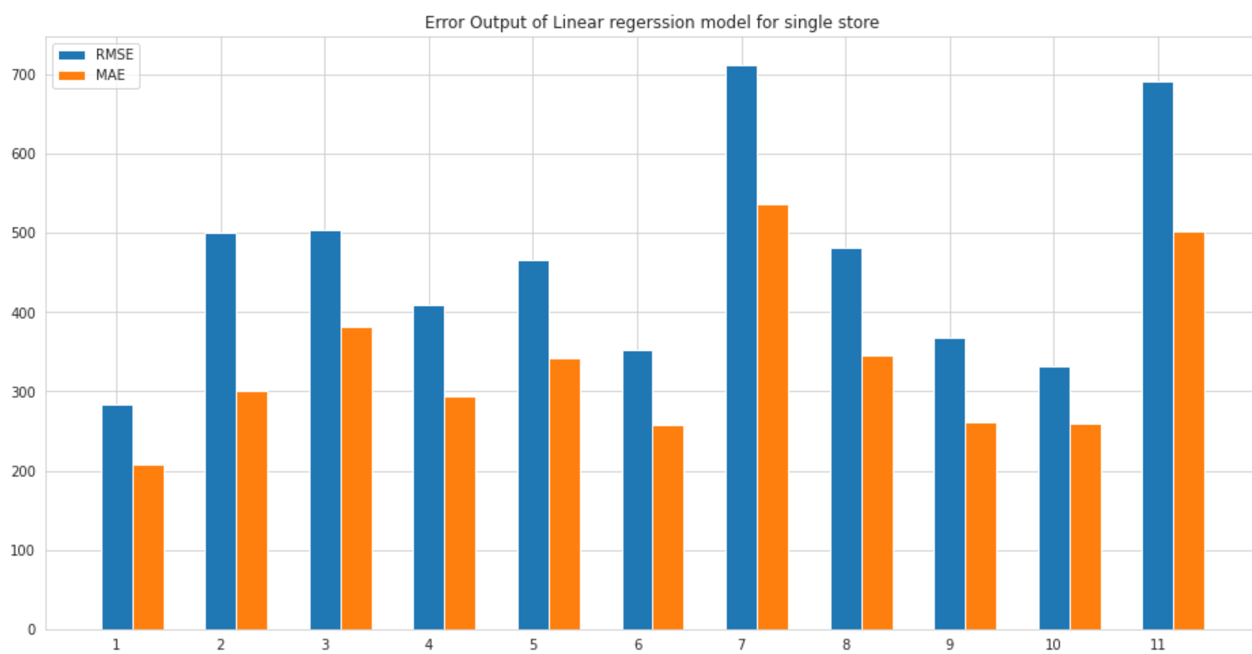
```
error_output_lr = pd.DataFrame()
error_output_lr['Stores'] = stores
error_output_lr['RMSE'] = RMSE_array_lr
error_output_lr['MAE'] = MAE_array_lr
error_output_lr
```

Out[27]:

	Stores	RMSE	MAE
0	1	284.267843	208.057711
1	2	499.547398	300.639640
2	3	503.516363	380.807745
3	4	408.954683	293.918216
4	5	466.091596	342.290667
5	6	352.195690	257.880767
6	7	712.565579	536.974131
7	8	481.808989	345.948447
8	9	368.470541	261.323809
9	10	331.881675	259.922444
10	11	691.636510	501.935023

In [28]:

```
plt.figure(figsize=(16,8))
N = 12
x = np.arange(1,N)
plt.bar(x,height=error_output_lr.RMSE,label = 'RMSE',width = 0.3)
plt.bar(x+0.3,height=error_output_lr.MAE,label = 'MAE',width = 0.3)
plt.xticks(stores)
plt.legend()
plt.title('Error Output of Linear regerssion model for single store')
plt.show()
```



c) Which performs better and Why?

Above model shows that accuracy increases when we train our model for individual store, because Sales might also depend on geographical or locality of the store, So when we predict for individual store then this factor could be treated as constant and can be neglected.

d) Try Ensemble of b) and c). What are the findings?

In [29]:

```
def adaboost_single_store(x,y):
    lr = AdaBoostRegressor(n_estimators=100, learning_rate=0.1)
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=42
    lr.fit(x_train,y_train)
    y_pred = lr.predict(x_test)
    return y_test,y_pred
```

In [30]:

```
RMSE_array_ada = []
MAE_array_ada =[]
for store in range(1,12):
    data = train_data[train_data.Store==store]
    data.drop('Store',axis=1,inplace=True)
    y=np.array(data['Sales'])
    x=np.array(data.drop('Sales',axis=1))
    y_true,y_pred = adaboost_single_store(x,y)
    RMSE_1,MAE_1 = error_cal(y_true,y_pred)
    RMSE_array_ada.append(RMSE_1)
    MAE_array_ada.append(MAE_1)
```

In [31]:

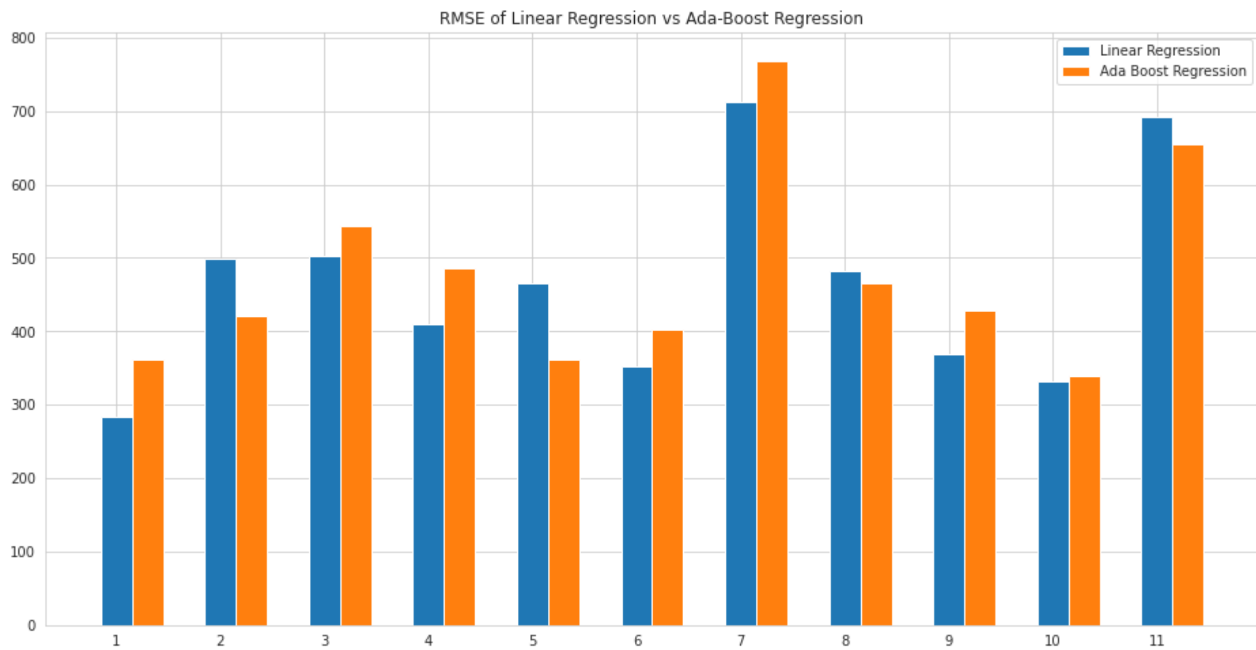
```
error_output_ada = pd.DataFrame()
error_output_ada['Stores'] = stores
error_output_ada['RMSE'] = RMSE_array_ada
error_output_ada['MAE'] = MAE_array_ada
error_output_ada
```

Out[31]:

	Stores	RMSE	MAE
0	1	361.714625	256.172928
1	2	421.019260	288.493914
2	3	543.486602	386.114747
3	4	485.118678	363.140434
4	5	361.671499	266.555999
5	6	402.878166	291.685944
6	7	768.739464	575.853433
7	8	464.738251	333.910930
8	9	428.176893	292.193204
9	10	338.385584	248.541183
10	11	655.379880	461.439511

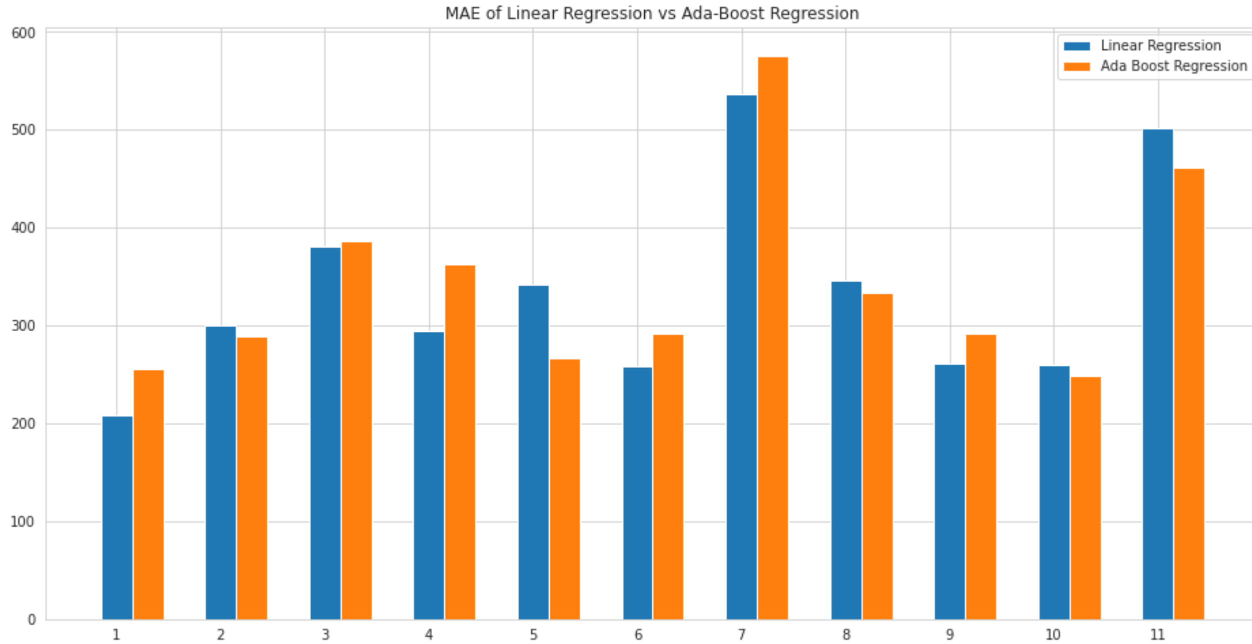
In [32]:

```
plt.figure(figsize=(16,8))
N=12
x=np.arange(1,N)
plt.bar(x,height=error_output_lr.RMSE,label = 'Linear Regression',width=0.3)
plt.bar(x+0.3,height=error_output_ada.RMSE,label = 'Ada Boost Regression',width=0.3)
plt.xticks(stores)
plt.legend()
plt.title('RMSE of Linear Regression vs Ada-Boost Regression')
plt.show()
```



In [33]:

```
plt.figure(figsize=(16,8))
N=12
x=np.arange(1,N)
plt.bar(x,height=error_output_lr.MAE,label = 'Linear Regression',width=0.3)
plt.bar(x+0.3,height=error_output_ada.MAE,label = 'Ada Boost Regression',width=0.3)
plt.xticks(stores)
plt.legend()
plt.title('MAE of Linear Regression vs Ada-Boost Regression')
plt.show()
```



By comparing RMSE and MAE of ada-boost regression and Linear regression, we can say that there is not much difference between these models

e) Use Regularized Regression. It should perform better in an unseen test set.

In [34]:

```
def ridge_single_store(x,y):
    ridge = Ridge(alpha=0.00001,normalize=True)
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=18
    ridge.fit(x_train,y_train)
    y_pred = ridge.predict(x_test)
    return y_test,y_pred
```

In [35]:

```
RMSE_array_rdg = []
MAE_array_rdg =[]
for store in range(1,12):
    data = train_data[train_data.Store==store]
    data.drop('Store',axis=1,inplace=True)
    y=np.array(data['Sales'])
    x=np.array(data.drop('Sales',axis=1))
    y_true,y_pred = ridge_single_store(x,y)
    RMSE_1,MAE_1 = error_cal(y_true,y_pred)
    RMSE_array_rdg.append(RMSE_1)
    MAE_array_rdg.append(MAE_1)
```

In [36]:

```
error_output_rdg = pd.DataFrame()
error_output_rdg['Stores'] = stores
error_output_rdg['RMSE'] = RMSE_array_rdg
error_output_rdg['MAE'] = MAE_array_rdg
error_output_rdg
```

Out[36]:

Stores	RMSE	MAE

	Stores	RMSE	MAE
0	1	253.651635	185.988246
1	2	658.876634	350.142560
2	3	505.016668	359.602319
3	4	440.340415	316.605281
4	5	482.497517	329.292617
5	6	318.540876	227.956072
6	7	717.864403	509.881475
7	8	472.085464	339.311060
8	9	403.765187	269.977126
9	10	335.135667	255.293371
10	11	688.828345	480.443768

f) Open-ended modeling to get possible predictions

In [ ]:

```
## Random forest regression model
```

In [37]:

```
def random_forest(x,y):
    rdm = RandomForestRegressor(n_estimators=60)
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=36
    rdm.fit(x_train,y_train)
    y_pred = rdm.predict(x_test)
    return y_test,y_pred
```

In [38]:

```
RMSE_array_rdm = []
MAE_array_rdm =[]
for store in range(1,12):
    data = train_data[train_data.Store==store]
    data.drop('Store',axis=1,inplace=True)
    y=np.array(data['Sales'])
    x=np.array(data.drop('Sales',axis=1))
    y_true,y_pred = random_forest(x,y)
    RMSE_1,MAE_1 = error_cal(y_true,y_pred)
    RMSE_array_rdm.append(RMSE_1)
    MAE_array_rdm.append(MAE_1)
```

In [39]:

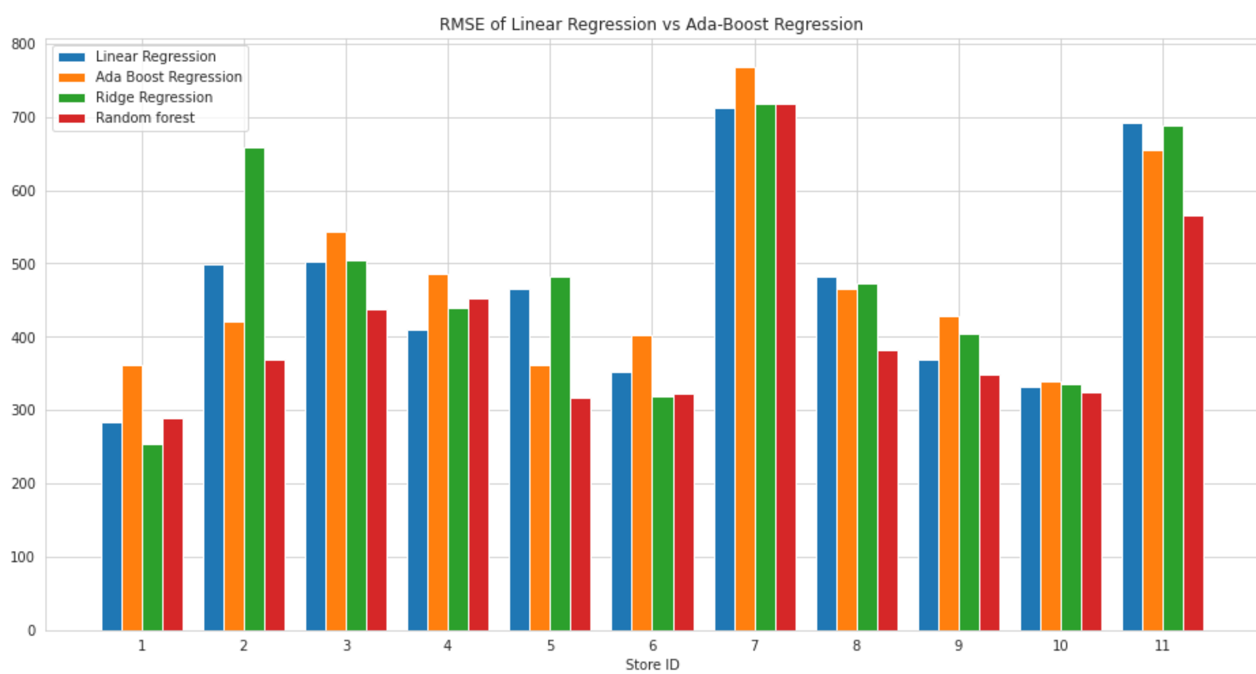
```
error_output_rdm = pd.DataFrame()
error_output_rdm['Stores'] = stores
error_output_rdm['RMSE'] = RMSE_array_rdm
error_output_rdm['MAE'] = MAE_array_rdm
error_output_rdm
```

Out[39]:

	Stores	RMSE	MAE
0	1	289.800038	209.037289
1	2	369.020455	230.685341
2	3	438.200612	307.383425
3	4	452.724575	315.005326
4	5	316.741952	223.909329
5	6	321.684684	227.355951
6	7	717.955617	473.584466
7	8	382.437141	260.311017
8	9	347.958906	242.574665
9	10	324.241841	237.028107
10	11	566.301076	387.558521

In [40]:

```
plt.figure(figsize=(16,8))
N=12
x=np.arange(1,N)
plt.bar(x,height=error_output_lr.RMSE,label = 'Linear Regression',width=0.2)
plt.bar(x+0.2,height=error_output_ada.RMSE,label = 'Ada Boost Regression',width=0.2)
plt.bar(x+0.4,height=error_output_rdg.RMSE,label = 'Ridge Regression',width=0.2)
plt.bar(x+0.6,height=error_output_rdm.RMSE,label = 'Random forest',width=0.2)
plt.xticks((2*x+0.6)/2,stores)
plt.xlabel('Store ID')
plt.legend()
plt.title('RMSE of Linear Regression vs Ada-Boost Regression')
plt.show()
```



In [41]:

```
# print('Average RMSE Linear regression Error: {}'.format(error_output_lr.RMSE.mean()))
```

```
print('Average RMSE Ada-boost regression Error: {}'.format(error_output_ada.RMSE.mean()
print('Average RMSE Ridge regression Error: {}'.format(error_output_rdg.RMSE.mean()))
print('Average RMSE Random Forest regression Error: {}'.format(error_output_rdm.RMSE.me
```

```
Average RMSE Ada-boost regression Error: 475.5735366099288
Average RMSE Ridge regression Error: 479.6911645282407
Average RMSE Random Forest regression Error: 411.551536106924
```

g) Other Regression Techniques:

1. When store is closed, sales = 0. Can this insight be used for Data Cleaning? Perform this and retrain the model. Any benefits

of this step? When store is closed then there will be no sale. Hence remove that rows.

In [42]:

```
open_store_data = train_data[train_data.Open == 1]
open_store_data.drop('Open',axis=1,inplace=True)
open_store_data.head()
```

Out[42]:

	Store	DayOfWeek	Sales	Customers	Promo	StateHoliday	SchoolHoliday
0	1	2	5735	568	1	0	0
1	2	2	9863	877	1	0	0
2	3	2	13261	1072	1	0	1
3	4	2	13106	1488	1	0	0
4	5	2	6635	645	1	0	0

In [43]:

```
RMSE_array_lrc = []
MAE_array_lrc=[]
for store in range(1,12):
    data = open_store_data[open_store_data.Store==store]
    data.drop('Store',axis=1,inplace=True)
    y=np.array(data['Sales'])
    x=np.array(data.drop('Sales',axis=1))
    y_true,y_pred = model_single_store(x,y)
    RMSE_1,MAE_1 = error_cal(y_true,y_pred)
    RMSE_array_lrc.append(RMSE_1)
    MAE_array_lrc.append(MAE_1)
```

In [44]:

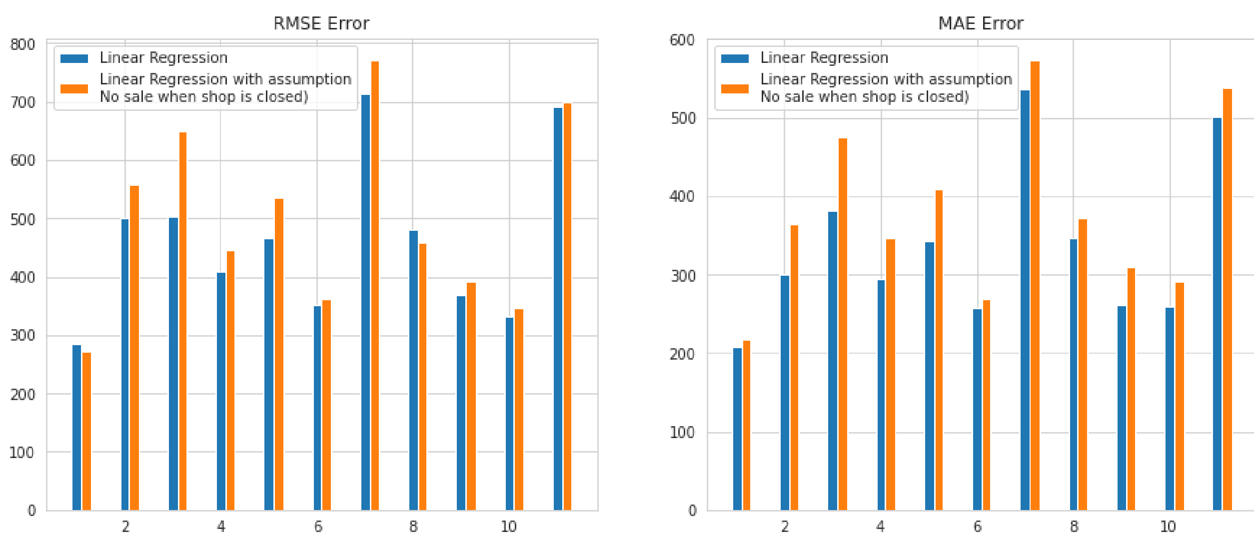
```
error_output_lrc = pd.DataFrame()
error_output_lrc['Stores'] = stores
error_output_lrc['RMSE'] = RMSE_array_lrc
error_output_lrc['MAE'] = MAE_array_lrc
error_output_lrc
```

Out[44]:

	Stores	RMSE	MAE
0	1	271.285014	216.794418
1	2	557.155161	364.294090
2	3	649.968842	476.379140
3	4	444.729628	346.986200
4	5	534.652008	409.715061
5	6	361.446231	269.610691
6	7	770.399228	572.891939
7	8	458.349199	371.704818
8	9	391.073084	309.433183
9	10	347.720527	290.885499
10	11	698.226695	537.427042

In [45]:

```
fig, axs = plt.subplots(1,2, figsize=(15,6))
fig.subplots_adjust(hspace=0.4)
axs=axs.ravel()
N=12
x=np.arange(1,N)
i=0
for col in ['RMSE','MAE']:
    axs[i].bar(x,height=error_output_lr[col],label = 'Linear Regression',width=0.2)
    axs[i].bar(x+0.2,height=error_output_lrc[col],label = 'Linear Regression with assum
    axs[i].legend()
    axs[i].set_title(col+' Error')
    i+=1
```



The above graph shoes that both types of error get increased when we removed the rows when store are closed. I think the main reason for this increased error rate is that, our previous model was predicting accurately when store was closed, so while taking mean of that portion the error output got reduced, but in updated model as we removed that rows, so while taking mean it get increased error output. So we do not get any benefit of removing those rows.

2. Use Non-Linear Regressors like Random Forest or other Tree-based Regressors.

a) Train a single model for all stores, where storeId can be a feature

In [46]:
```
open_store_data.head()
```

Out[46]:

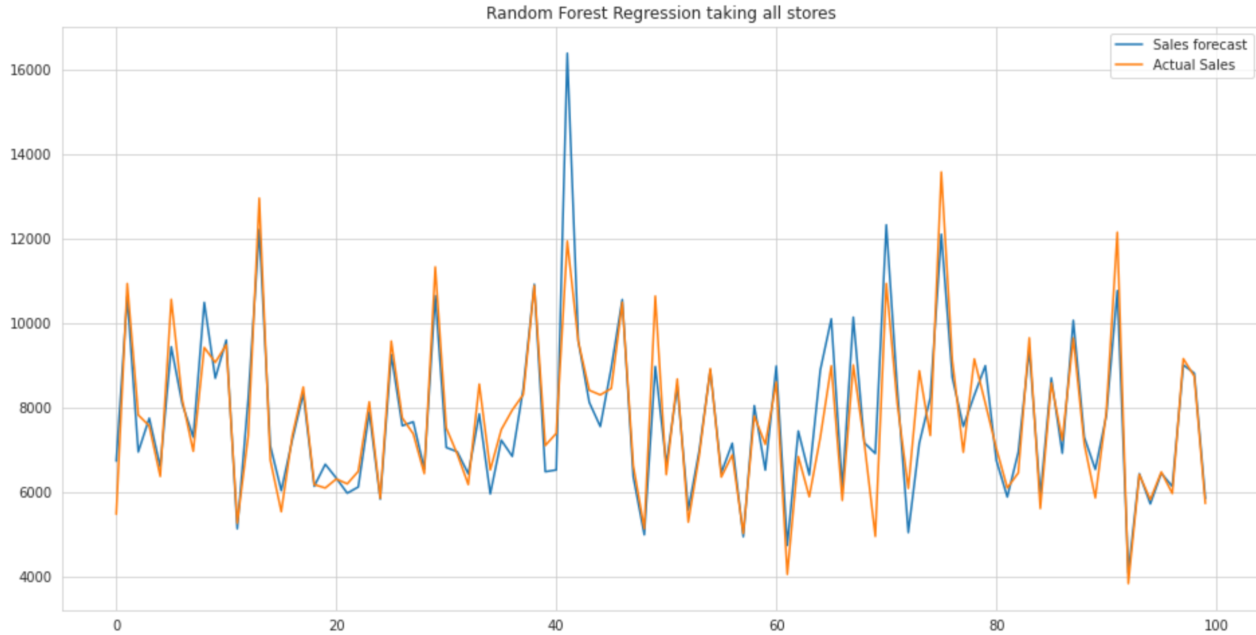
	Store	DayOfWeek	Sales	Customers	Promo	StateHoliday	SchoolHoliday
0	1	2	5735	568	1	0	0
1	2	2	9863	877	1	0	0
2	3	2	13261	1072	1	0	1
3	4	2	13106	1488	1	0	0
4	5	2	6635	645	1	0	0

In [47]:
```
y=np.array(data['Sales'])
x=np.array(data.drop('Sales',axis=1))
y_true,y_pred = random_forest(x,y)
RMSE_rdm,MAE_rdm = error_cal(y_true,y_pred)
```

In [48]:
```
print('Root mean squared error: ',RMSE_rdm)
print('Mean absolute error: ',MAE_rdm)
```

```
Root mean squared error:  735.7698390478077
Mean absolute error:  500.33408312447796
```

In [49]:
```
plt.figure(figsize=(16,8))
plt.plot(y_pred[:100],label = 'Sales forecast')
plt.plot(y_true[:100],label = 'Actual Sales')
plt.legend()
plt.title('Random Forest Regression taking all stores')
plt.show()
```



b) Train separate models for each store

In [50]:
```
open_store_data.reset_index(drop=True,inplace=True)
```

In [51]:
```
x = open_store_data.drop(['Sales','Store'],axis=1)
x = StandardScaler().fit_transform(x)
```

In [52]:
```
pca = PCA(n_components=3)
principalComponents = pca.fit_transform(x)
```

In [53]:
```
principalDf = pd.DataFrame(data = principalComponents, columns = ['PC_1', 'PC_2','PC_3'
```

In [54]:
```
finaldf = pd.concat([open_store_data[['Store','Sales']],principalDf],axis=1)
```

In [55]:
```
finaldf.head()
```

Out[55]:

	Store	Sales	PC_1	PC_2	PC_3
0	1	5735	0.891968	-0.602281	0.120771
1	2	9863	1.221942	-0.516040	0.489858
2	3	13261	2.075741	0.506591	-1.261974
3	4	13106	1.874415	-0.345512	1.219671
4	5	6635	0.974194	-0.580790	0.212744

In [56]:
```
finaldf.reset_index(drop=True,inplace=True)
```

In [57]:
```
## k-fold
```

In [58]:
```
def get_stats(model,x_train,y_train,x_test,y_test):
    model.fit(x_train,y_train)
    y_pred = model.predict(x_test)
    RMSE,MAE = error_cal(y_test,y_pred)
    return [RMSE,MAE]
```

In [59]:
```
from sklearn.model_selection import StratifiedKFold
from sklearn.tree import DecisionTreeRegressor
dt = DecisionTreeRegressor()
```

In [60]:
```
y = np.array(finaldf['Sales'])
x = np.array(finaldf.drop('Sales',axis=1))
```

In [64]:
```
score_rdm = []
score_dt = []
kf = StratifiedKFold(n_splits=5)
for train_index,test_index in kf.split(x,y):
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.3,random_state=5
    score_dt.append(get_stats(DecisionTreeRegressor(),x_train,y_train,x_test,y_test))
    score_rdm.append(get_stats(RandomForestRegressor(n_estimators=10),x_train,y_train,x
```

In [65]:
```
k_fold_df = pd.DataFrame()
k_fold_df['decision_tree']=pd.DataFrame(score_dt,columns=['RMSE','MAE']).mean(axis=0)
k_fold_df['Random_forest']=pd.DataFrame(score_rdm,columns=['RMSE','MAE']).mean(axis=0)
```

In [66]:
```
k_fold_df
```

Out[66]:

	decision_tree	Random_forest
RMSE	1110.796469	937.531476
MAE	667.480079	604.750720

for individual stores

In [67]:
```
## random forest model
RMSE_array_rdm_pca = []
MAE_array_rdm_pca = []
for store in range(1,12):
    data = finaldf[finaldf.Store==store]
    data.drop('Store',axis=1,inplace=True)
    y=np.array(data['Sales'])
    x=np.array(data.drop('Sales',axis=1))
    y_true,y_pred = random_forest(x,y)
    RMSE_1,MAE_1 = error_cal(y_true,y_pred)
```

```
    RMSE_array_rdm_pca.append(RMSE_1)
    MAE_array_rdm_pca.append(MAE_1)
```

In [68]:

```
error_output_rdm_pca = pd.DataFrame()
error_output_rdm_pca['Stores'] = stores
error_output_rdm_pca['RMSE'] = RMSE_array_rdm_pca
error_output_rdm_pca['MAE'] = MAE_array_rdm_pca
error_output_rdm_pca
```

Out[68]:

	Stores	RMSE	MAE
0	1	360.186233	271.153220
1	2	479.101762	301.414069
2	3	508.135152	373.457223
3	4	671.479959	501.701625
4	5	529.257394	338.803512
5	6	482.294879	346.129747
6	7	814.640762	600.077670
7	8	433.178325	311.003786
8	9	500.060012	374.094962
9	10	391.598935	310.417053
10	11	751.909576	535.835465

In [69]:

```
def decision_tree(x,y):
    dt = DecisionTreeRegressor()
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=36
    dt.fit(x_train,y_train)
    y_pred = dt.predict(x_test)
    return y_test,y_pred
```

In [70]:

```
## decision tree model
RMSE_array_dt_pca = []
MAE_array_dt_pca = []
for store in range(1,12):
    data = finaldf[finaldf.Store==store]
    data.drop('Store',axis=1,inplace=True)
    y=np.array(data['Sales'])
    x=np.array(data.drop('Sales',axis=1))
    y_true,y_pred = decision_tree(x,y)
    RMSE_1,MAE_1 = error_cal(y_true,y_pred)
    RMSE_array_dt_pca.append(RMSE_1)
    MAE_array_dt_pca.append(MAE_1)
```

In [71]:

```
error_output_dt_pca = pd.DataFrame()
error_output_dt_pca['Stores'] = stores
error_output_dt_pca['RMSE'] = RMSE_array_dt_pca
```

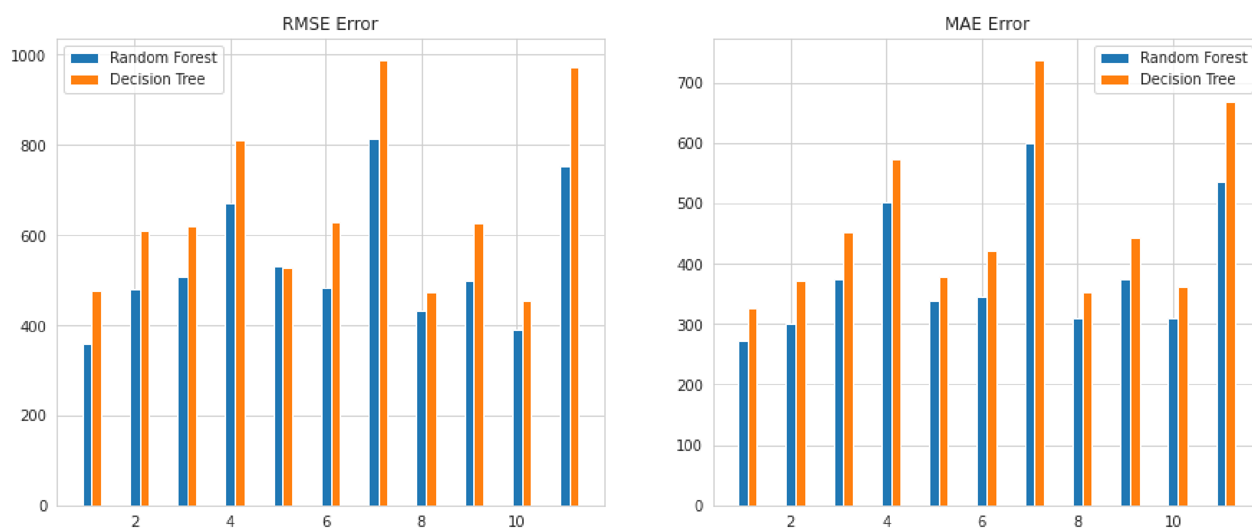
```
error_output_dt_pca['MAE'] = MAE_array_dt_pca
error_output_dt_pca
```

Out[71]:

	Stores	RMSE	MAE
0	1	477.549418	326.634921
1	2	609.044301	372.554386
2	3	619.815183	452.389184
3	4	809.361921	573.094737
4	5	527.454703	379.873227
5	6	628.113097	422.454145
6	7	988.034550	736.457895
7	8	473.650918	351.815789
8	9	626.786138	442.084220
9	10	455.739281	362.482456
10	11	971.680657	668.550000

In [72]:

```
fig, axs = plt.subplots(1,2, figsize=(15,6))
fig.subplots_adjust(hspace=0.4)
axs=axs.ravel()
N=12
x=np.arange(1,N)
i=0
for col in ['RMSE','MAE']:
    axs[i].bar(x,height=error_output_rdm_pca[col],label = 'Random Forest',width=0.2)
    axs[i].bar(x+0.2,height=error_output_dt_pca[col],label = 'Decision Tree',width=0.2)
    axs[i].legend()
    axs[i].set_title(col+' Error')
    i+=1
```



In [73]:

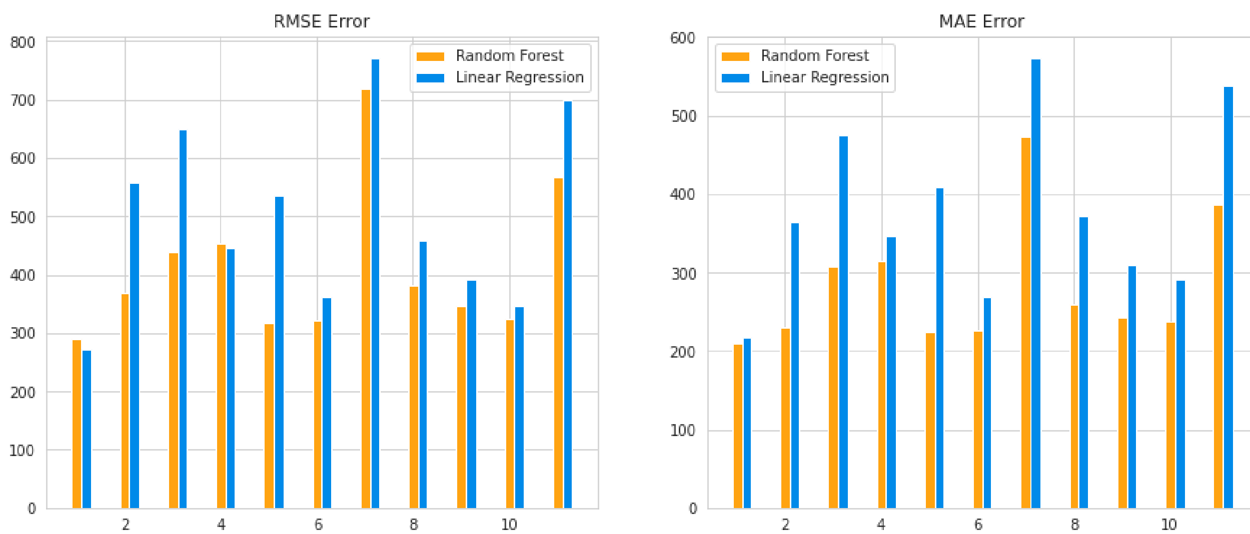
```
print('Average RMSE Decision Tree Error: {}'.format(error_output_dt_pca.RMSE.mean()))
print('Average RMSE Random Forest Error: {}'.format(error_output_rdm_pca.RMSE.mean()))
```

```
Average RMSE Decision Tree Error: 653.3845605434634
Average RMSE Random Forest Error: 538.3493627622491
```

>>Compare the performance of Linear Model and Non-Linear Model from the previous observations. Which performs better and why?

In [78]:

```
fig, axs = plt.subplots(1,2, figsize=(15,6))
fig.subplots_adjust(hspace=0.4)
axs=axs.ravel()
N=12
x=np.arange(1,N)
i=0
for col in ['RMSE','MAE']:
    axs[i].bar(x,height=error_output_rdm[col],label = 'Random Forest',width=0.2,color =
    axs[i].bar(x+0.2,height=error_output_lrc[col],label = 'Linear Regression',width=0.2
    axs[i].legend()
    axs[i].set_title(col+' Error')
    i+=1
```



In [79]:

```
print('Average RMSE Random Forest Error: {}'.format(error_output_rdm.RMSE.mean()))
print('Average RMSE Linear Regression Error: {}'.format(error_output_lrc.RMSE.mean()))
```

```
Average RMSE Random Forest Error: 411.551536106924
Average RMSE Linear Regression Error: 498.63687432231706
```

Lets Train the time series model on the data with only time as the feature

Time series Analysis

In [23]:

```
# a) Identify yearly trends and seasonal months
```

In [81]:

```
def test_stationarity(timeseries):
    rolmean = timeseries.rolling(window=52,center = False).mean()
    rolstd = timeseries.rolling(window = 52,center = False).std()
    plt.figure(figsize=(16,8))
    orig = plt.plot(timeseries,color = '#3399ff',label = 'Original')
    mean = plt.plot(rolmean,color = 'red',label = 'Rolling Mean')
    std = plt.plot(rolstd,color = 'green',label = 'Rolling Std')
    plt.title('Rolling mean and Standard deviation')
    plt.legend(loc='best')
    plt.show(block=False)
    print('Result of Dickey-Fuller Test: ')
    dftest = adfuller(timeseries,autolag='AIC')
    dfoutput = pd.Series(dftest[0:4],index=['Test Statistic','p-value','Number of lag u
    for key,value in dftest[4].items():
        dfoutput['Critical Value (%s)'%key] = value
        print(dfoutput)
```

In [82]:

```
original_data.head()
```

Out[82]:

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	2	2015-06-30	5735	568	1	1	0	0
1	2	2	2015-06-30	9863	877	1	1	0	0
2	3	2	2015-06-30	13261	1072	1	1	0	1
3	4	2	2015-06-30	13106	1488	1	1	0	0
4	5	2	2015-06-30	6635	645	1	1	0	0

In [83]:

```
original_data.sort_values('Date',inplace=True)
original_data = original_data[original_data.Open==1]
original_data.reset_index(drop = True,inplace=True)
```

In [85]:

```
datax = original_data[original_data.Store==1][['Date','Sales']]
datax.set_index('Date',inplace=True)
datax
```

Out[85]:

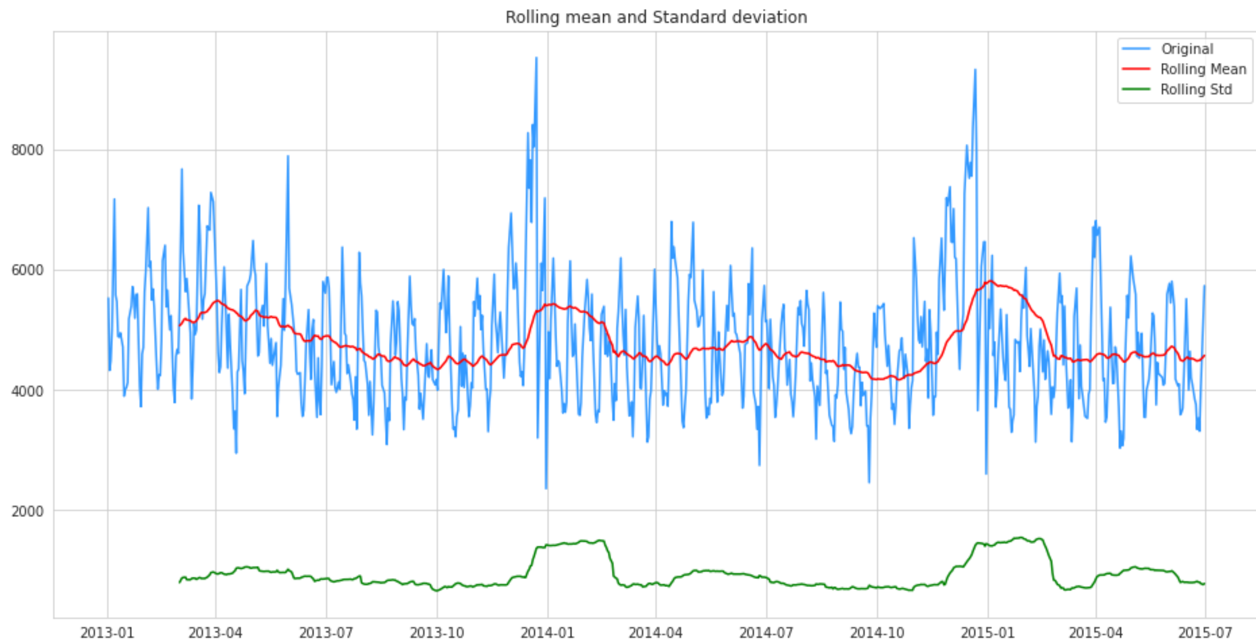
	Sales
Date	
2013-01-02	5530
2013-01-03	4327
2013-01-04	4486
2013-01-05	4997
2013-01-07	7176
...	...

	Sales
Date	
2015-06-25	3533
2015-06-26	3317
2015-06-27	4019
2015-06-29	5197
2015-06-30	5735

754 rows × 1 columns

In [86]:

```
test_stationarity(datax)
```




```
Result of Dickey-Fuller Test:
Test Statistic                 -5.336702
p-value                         0.000005
Number of lag used             13.000000
Number of observation used    740.000000
Critical Value (1%)            -3.439218
dtype: float64
Test Statistic                 -5.336702
p-value                         0.000005
Number of lag used             13.000000
Number of observation used    740.000000
Critical Value (1%)            -3.439218
Critical Value (5%)            -2.865454
dtype: float64
Test Statistic                 -5.336702
p-value                         0.000005
Number of lag used             13.000000
Number of observation used    740.000000
Critical Value (1%)            -3.439218
Critical Value (5%)            -2.865454
```

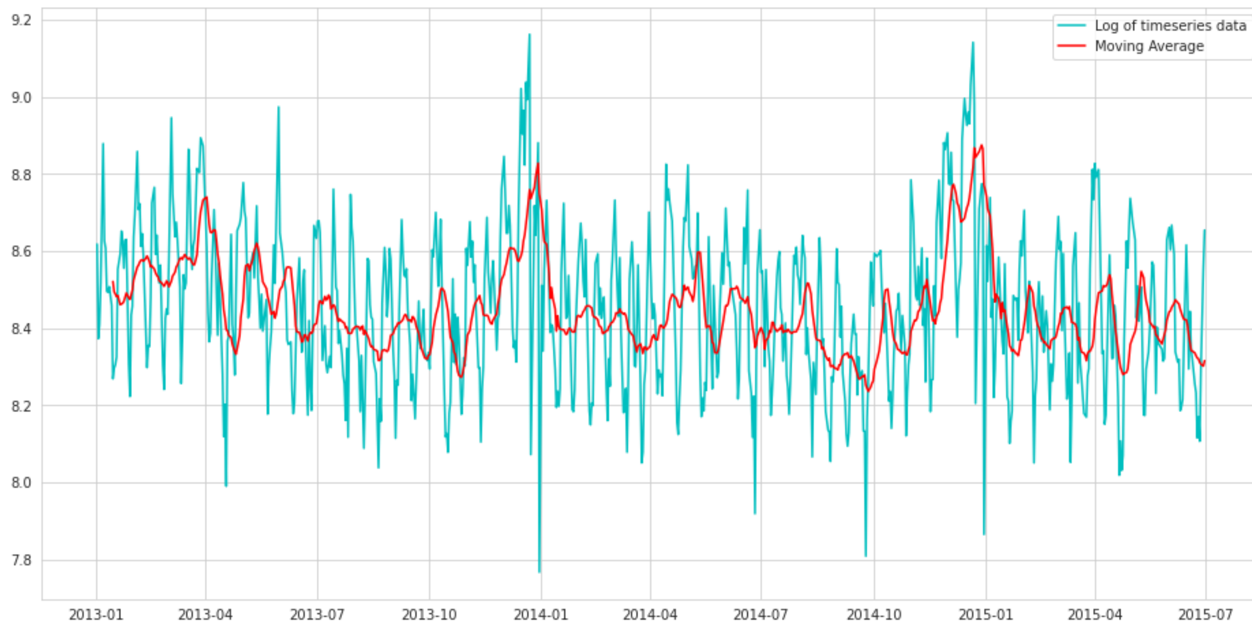
```
Critical Value (10%)          -2.568854
dtype: float64
```

In [87]:

```
# p-value is very close to zero so we will reject the null hypothesis, that data does n
# However, data shows some seasonal effects.
```

In [92]:

```
ts_log = np.log(datax)
movingavg = ts_log.rolling(window = 12).mean()
plt.figure(figsize=(16,8))
plt.plot(ts_log,color='c',label = 'Log of timeseries data')
plt.plot(movingavg,color='r',label = 'Moving Average')
plt.legend()
plt.show()
```



In [93]:

```
#From the above graph we can see seasonal effect in the dataset.
#In dec to jan month sale is high in comaprison to other month
```

In [94]:

```
ts_log_mv_diff = ts_log - movingavg
ts_log_mv_diff.dropna(inplace=True)
```

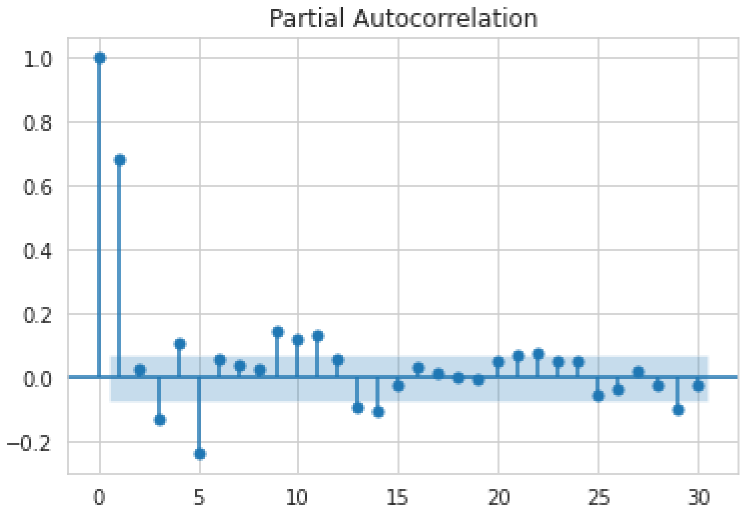
In [95]:

```
# Since p-value is less than 0.05, so we can say that data is stationary.
# hence differencing is not required, therefore d = 0.
```

In [100…

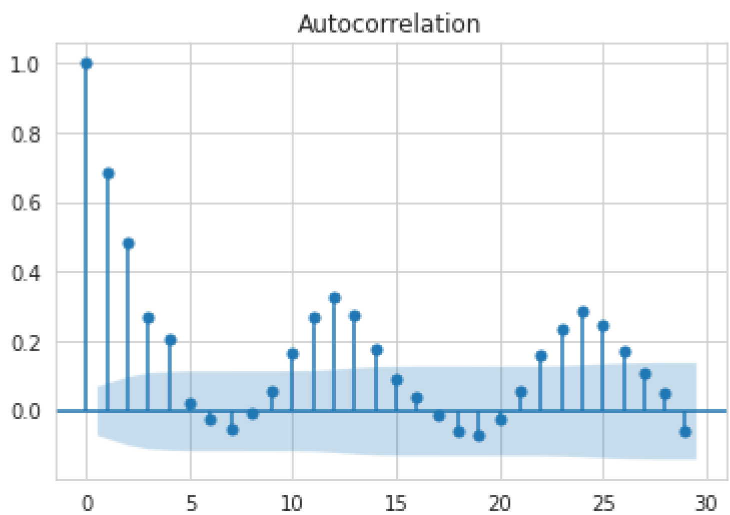
```
plt.figure(figsize=(16,8))
plot_pacf(datax.dropna(), lags=30)
plt.show()
```

```
<Figure size 1152x576 with 0 Axes>
```



In [101…

```
plot_acf(datax.dropna())
plt.show()
```

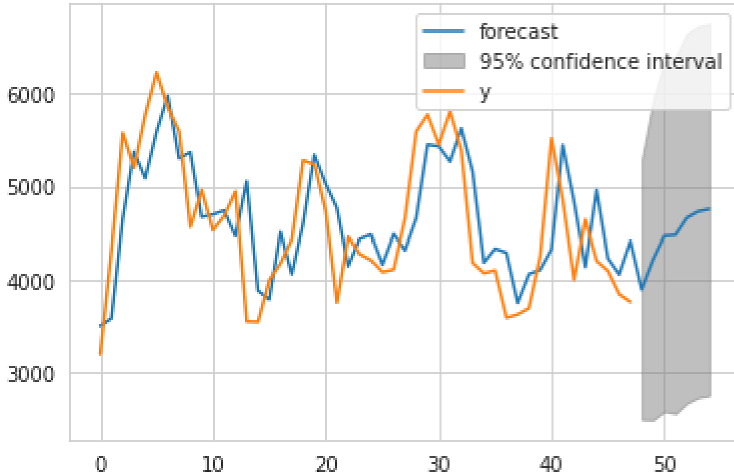


In [102…

```
model = ARIMA(np.array(datax[:-6]), order=(1, 0, 4))
results = model.fit()
```

In [103…

```
results.plot_predict(700,754)
plt.show()
```



In [104…

```
results.summary()
```

Out[104…

ARMA Model Results

Dep. Variable:	y	**No. Observations:**	748
Model:	ARMA(1, 4)	**Log Likelihood**	-5977.693
Method:	css-mle	**S.D. of innovations**	714.770
Date:	Wed, 18 May 2022	**AIC**	11969.386
Time:	03:44:38	**BIC**	12001.707
Sample:	0	**HQIC**	11981.841

	coef	std err	z	P>|z|	[0.025	0.975]
const	4771.2335	81.286	58.697	0.000	4611.916	4930.551
ar.L1.y	0.3738	0.127	2.949	0.003	0.125	0.622
ma.L1.y	0.3420	0.127	2.689	0.007	0.093	0.591
ma.L2.y	0.2795	0.092	3.030	0.002	0.099	0.460
ma.L3.y	0.0544	0.062	0.878	0.380	-0.067	0.176
ma.L4.y	0.2763	0.035	8.002	0.000	0.209	0.344

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	2.6754	+0.0000j	2.6754	0.0000
MA.1	-0.8914	-0.9326j	1.2901	-0.3714
MA.2	-0.8914	+0.9326j	1.2901	0.3714
MA.3	0.7930	-1.2434j	1.4748	-0.1596
MA.4	0.7930	+1.2434j	1.4748	0.1596

In [105…

```
RMSE_ARIMA = math.sqrt(mean_squared_error(np.array(datax[700:]) , results.predict(700,7
```

```
RMSE_ARIMA
```

Out[105…

587.1586723760986

In [106…

```
MAE_ARIMA = mean_absolute_error(np.array(datax[700:]) , results.predict(700,753))
MAE_ARIMA
```

Out[106…

482.53793430410224

Project Task: Week 2

Implementing Neural Networks:

1 Train a LSTM on the same set of features and compare the result with traditional time-series model

In [110…

```
std = datax.std()
mean = datax.mean()
timeseries = np.array((datax-mean)/std)
```

In [112…

```
training_size = int(len(timeseries)*0.65)
test_size = len(timeseries)-training_size
train_size,test_size = timeseries[:training_size,:],timeseries[training_size:len(timese
```

In [113…

```
def create_dataset(dataset,time_step = 1):
    dataX,dataY = [],[]
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step),0]
        dataX.append(a)
        dataY.append(dataset[i+time_step,0])
    return np.array(dataX),np.array(dataY)
```

In [114…

```
time_step =100
x_train,y_train = create_dataset(train_size,time_step)
x_test,y_test = create_dataset(test_size,time_step)
```

In [115…

```
x_train = x_train.reshape(x_train.shape[0],x_train.shape[1],1)
x_test = x_test.reshape(x_test.shape[0],x_test.shape[1],1)
```

In [116…

```
model = Sequential()
model.add(LSTM(50,return_sequences = True,input_shape = (100,1)))
model.add(LSTM(50,return_sequences = True))
model.add(LSTM(50))
```

```
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer = 'adam')
```

In [117…

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400     
                                                                 
 lstm_1 (LSTM)               (None, 100, 50)           20200     
                                                                 
 lstm_2 (LSTM)               (None, 50)                20200     
                                                                 
 dense (Dense)               (None, 1)                 51        
                                                                 
=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

In [118…

```
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
checkpoint = ModelCheckpoint('Sales.h5',monitor='loss',mode=min,save_best_only=True,ver
early_stopping = EarlyStopping(monitor='loss',patience=9,min_delta=0,restore_best_weigh
Reduce_ler_rate = ReduceLROnPlateau(monitor='loss',factor=0.2,patience=3,verbose=1,min_
callback = [checkpoint,early_stopping,Reduce_ler_rate]
```

```
WARNING:tensorflow:ModelCheckpoint mode <built-in function min> is unknown, fallback to
auto mode.
```

In [119…

```
history = model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=200,batch_si
```

```
Epoch 1/200
7/7 [==============================] - ETA: 0s - loss: 0.8874
Epoch 1: loss improved from inf to 0.88738, saving model to Sales.h5
7/7 [==============================] - 9s 494ms/step - loss: 0.8874 - val_loss: 1.2218 -
lr: 0.0010
Epoch 2/200
7/7 [==============================] - ETA: 0s - loss: 0.8732
Epoch 2: loss improved from 0.88738 to 0.87319, saving model to Sales.h5
7/7 [==============================] - 2s 286ms/step - loss: 0.8732 - val_loss: 1.1685 -
lr: 0.0010
Epoch 3/200
7/7 [==============================] - ETA: 0s - loss: 0.8697
Epoch 3: loss improved from 0.87319 to 0.86969, saving model to Sales.h5
7/7 [==============================] - 2s 283ms/step - loss: 0.8697 - val_loss: 1.1560 -
lr: 0.0010
Epoch 4/200
7/7 [==============================] - ETA: 0s - loss: 0.8641
Epoch 4: loss improved from 0.86969 to 0.86412, saving model to Sales.h5
7/7 [==============================] - 2s 279ms/step - loss: 0.8641 - val_loss: 1.1778 -
lr: 0.0010
Epoch 5/200
7/7 [==============================] - ETA: 0s - loss: 0.8620
Epoch 5: loss improved from 0.86412 to 0.86196, saving model to Sales.h5
```

```
7/7 [==============================] - 2s 280ms/step - loss: 0.8620 - val_loss: 1.1553 -
lr: 0.0010
Epoch 6/200
7/7 [==============================] - ETA: 0s - loss: 0.8550
Epoch 6: loss improved from 0.86196 to 0.85505, saving model to Sales.h5
7/7 [==============================] - 2s 281ms/step - loss: 0.8550 - val_loss: 1.0922 -
lr: 0.0010
Epoch 7/200
7/7 [==============================] - ETA: 0s - loss: 0.8415
Epoch 7: loss improved from 0.85505 to 0.84145, saving model to Sales.h5
7/7 [==============================] - 2s 280ms/step - loss: 0.8415 - val_loss: 1.0418 -
lr: 0.0010
Epoch 8/200
7/7 [==============================] - ETA: 0s - loss: 0.8241
Epoch 8: loss improved from 0.84145 to 0.82409, saving model to Sales.h5
7/7 [==============================] - 2s 282ms/step - loss: 0.8241 - val_loss: 0.9878 -
lr: 0.0010
Epoch 9/200
7/7 [==============================] - ETA: 0s - loss: 0.7970
Epoch 9: loss improved from 0.82409 to 0.79695, saving model to Sales.h5
7/7 [==============================] - 2s 282ms/step - loss: 0.7970 - val_loss: 1.0509 -
lr: 0.0010
Epoch 10/200
7/7 [==============================] - ETA: 0s - loss: 0.7977
Epoch 10: loss did not improve from 0.79695
7/7 [==============================] - 2s 269ms/step - loss: 0.7977 - val_loss: 0.8135 -
lr: 0.0010
Epoch 11/200
7/7 [==============================] - ETA: 0s - loss: 0.7597
Epoch 11: loss improved from 0.79695 to 0.75967, saving model to Sales.h5
7/7 [==============================] - 2s 284ms/step - loss: 0.7597 - val_loss: 0.9649 -
lr: 0.0010
Epoch 12/200
7/7 [==============================] - ETA: 0s - loss: 0.7260
Epoch 12: loss improved from 0.75967 to 0.72601, saving model to Sales.h5
7/7 [==============================] - 2s 285ms/step - loss: 0.7260 - val_loss: 0.9229 -
lr: 0.0010
Epoch 13/200
7/7 [==============================] - ETA: 0s - loss: 0.7151
Epoch 13: loss improved from 0.72601 to 0.71513, saving model to Sales.h5
7/7 [==============================] - 2s 288ms/step - loss: 0.7151 - val_loss: 0.8787 -
lr: 0.0010
Epoch 14/200
7/7 [==============================] - ETA: 0s - loss: 0.6647
Epoch 14: loss improved from 0.71513 to 0.66474, saving model to Sales.h5
7/7 [==============================] - 2s 313ms/step - loss: 0.6647 - val_loss: 0.7874 -
lr: 0.0010
Epoch 15/200
7/7 [==============================] - ETA: 0s - loss: 0.6264
Epoch 15: loss improved from 0.66474 to 0.62640, saving model to Sales.h5
7/7 [==============================] - 2s 284ms/step - loss: 0.6264 - val_loss: 0.6971 -
lr: 0.0010
Epoch 16/200
7/7 [==============================] - ETA: 0s - loss: 0.6038
Epoch 16: loss improved from 0.62640 to 0.60380, saving model to Sales.h5
7/7 [==============================] - 2s 313ms/step - loss: 0.6038 - val_loss: 0.7432 -
lr: 0.0010
Epoch 17/200
7/7 [==============================] - ETA: 0s - loss: 0.5862
Epoch 17: loss improved from 0.60380 to 0.58618, saving model to Sales.h5
```

```
7/7 [==============================] - 2s 290ms/step - loss: 0.5862 - val_loss: 0.7466 -
lr: 0.0010
Epoch 18/200
7/7 [==============================] - ETA: 0s - loss: 0.5818
Epoch 18: loss improved from 0.58618 to 0.58176, saving model to Sales.h5
7/7 [==============================] - 2s 283ms/step - loss: 0.5818 - val_loss: 0.7255 -
lr: 0.0010
Epoch 19/200
7/7 [==============================] - ETA: 0s - loss: 0.5520
Epoch 19: loss improved from 0.58176 to 0.55196, saving model to Sales.h5
7/7 [==============================] - 2s 311ms/step - loss: 0.5520 - val_loss: 0.6388 -
lr: 0.0010
Epoch 20/200
7/7 [==============================] - ETA: 0s - loss: 0.5280
Epoch 20: loss improved from 0.55196 to 0.52799, saving model to Sales.h5
7/7 [==============================] - 2s 287ms/step - loss: 0.5280 - val_loss: 0.6199 -
lr: 0.0010
Epoch 21/200
7/7 [==============================] - ETA: 0s - loss: 0.5388
Epoch 21: loss did not improve from 0.52799
7/7 [==============================] - 2s 308ms/step - loss: 0.5388 - val_loss: 0.5845 -
lr: 0.0010
Epoch 22/200
7/7 [==============================] - ETA: 0s - loss: 0.5099
Epoch 22: loss improved from 0.52799 to 0.50991, saving model to Sales.h5
7/7 [==============================] - 2s 288ms/step - loss: 0.5099 - val_loss: 0.6728 -
lr: 0.0010
Epoch 23/200
7/7 [==============================] - ETA: 0s - loss: 0.5043
Epoch 23: loss improved from 0.50991 to 0.50434, saving model to Sales.h5
7/7 [==============================] - 2s 285ms/step - loss: 0.5043 - val_loss: 0.6674 -
lr: 0.0010
Epoch 24/200
7/7 [==============================] - ETA: 0s - loss: 0.5196
Epoch 24: loss did not improve from 0.50434
7/7 [==============================] - 2s 263ms/step - loss: 0.5196 - val_loss: 0.6014 -
lr: 0.0010
Epoch 25/200
7/7 [==============================] - ETA: 0s - loss: 0.4990
Epoch 25: loss improved from 0.50434 to 0.49897, saving model to Sales.h5
7/7 [==============================] - 2s 275ms/step - loss: 0.4990 - val_loss: 0.5694 -
lr: 0.0010
Epoch 26/200
7/7 [==============================] - ETA: 0s - loss: 0.5098
Epoch 26: loss did not improve from 0.49897
7/7 [==============================] - 2s 263ms/step - loss: 0.5098 - val_loss: 0.6946 -
lr: 0.0010
Epoch 27/200
7/7 [==============================] - ETA: 0s - loss: 0.5188
Epoch 27: loss did not improve from 0.49897
7/7 [==============================] - 2s 264ms/step - loss: 0.5188 - val_loss: 0.6639 -
lr: 0.0010
Epoch 28/200
7/7 [==============================] - ETA: 0s - loss: 0.4926
Epoch 28: loss improved from 0.49897 to 0.49259, saving model to Sales.h5
7/7 [==============================] - 2s 277ms/step - loss: 0.4926 - val_loss: 0.6545 -
lr: 0.0010
Epoch 29/200
7/7 [==============================] - ETA: 0s - loss: 0.4868
Epoch 29: loss improved from 0.49259 to 0.48681, saving model to Sales.h5
```

```
7/7 [==============================] - 2s 275ms/step - loss: 0.4868 - val_loss: 0.6391 - 
lr: 0.0010
Epoch 30/200
7/7 [==============================] - ETA: 0s - loss: 0.4713
Epoch 30: loss improved from 0.48681 to 0.47131, saving model to Sales.h5
7/7 [==============================] - 2s 322ms/step - loss: 0.4713 - val_loss: 0.5819 - 
lr: 0.0010
Epoch 31/200
7/7 [==============================] - ETA: 0s - loss: 0.4611
Epoch 31: loss improved from 0.47131 to 0.46105, saving model to Sales.h5
7/7 [==============================] - 2s 301ms/step - loss: 0.4611 - val_loss: 0.5339 - 
lr: 0.0010
Epoch 32/200
7/7 [==============================] - ETA: 0s - loss: 0.4444
Epoch 32: loss improved from 0.46105 to 0.44435, saving model to Sales.h5
7/7 [==============================] - 2s 285ms/step - loss: 0.4444 - val_loss: 0.5535 - 
lr: 0.0010
Epoch 33/200
7/7 [==============================] - ETA: 0s - loss: 0.4729
Epoch 33: loss did not improve from 0.44435
7/7 [==============================] - 2s 264ms/step - loss: 0.4729 - val_loss: 0.6456 - 
lr: 0.0010
Epoch 34/200
7/7 [==============================] - ETA: 0s - loss: 0.4615
Epoch 34: loss did not improve from 0.44435
7/7 [==============================] - 2s 262ms/step - loss: 0.4615 - val_loss: 0.5502 - 
lr: 0.0010
Epoch 35/200
7/7 [==============================] - ETA: 0s - loss: 0.4256
Epoch 35: loss improved from 0.44435 to 0.42563, saving model to Sales.h5
7/7 [==============================] - 2s 274ms/step - loss: 0.4256 - val_loss: 0.5139 - 
lr: 0.0010
Epoch 36/200
7/7 [==============================] - ETA: 0s - loss: 0.4153
Epoch 36: loss improved from 0.42563 to 0.41530, saving model to Sales.h5
7/7 [==============================] - 2s 279ms/step - loss: 0.4153 - val_loss: 0.5133 - 
lr: 0.0010
Epoch 37/200
7/7 [==============================] - ETA: 0s - loss: 0.4063
Epoch 37: loss improved from 0.41530 to 0.40627, saving model to Sales.h5
7/7 [==============================] - 2s 276ms/step - loss: 0.4063 - val_loss: 0.5217 - 
lr: 0.0010
Epoch 38/200
7/7 [==============================] - ETA: 0s - loss: 0.3902
Epoch 38: loss improved from 0.40627 to 0.39016, saving model to Sales.h5
7/7 [==============================] - 2s 282ms/step - loss: 0.3902 - val_loss: 0.5340 - 
lr: 0.0010
Epoch 39/200
7/7 [==============================] - ETA: 0s - loss: 0.3915
Epoch 39: loss did not improve from 0.39016
7/7 [==============================] - 2s 263ms/step - loss: 0.3915 - val_loss: 0.5153 - 
lr: 0.0010
Epoch 40/200
7/7 [==============================] - ETA: 0s - loss: 0.3803
Epoch 40: loss improved from 0.39016 to 0.38032, saving model to Sales.h5
7/7 [==============================] - 2s 289ms/step - loss: 0.3803 - val_loss: 0.5145 - 
lr: 0.0010
Epoch 41/200
7/7 [==============================] - ETA: 0s - loss: 0.3784
Epoch 41: loss improved from 0.38032 to 0.37837, saving model to Sales.h5
```

```
7/7 [==============================] - 2s 294ms/step - loss: 0.3784 - val_loss: 0.5358 -
lr: 0.0010
Epoch 42/200
7/7 [==============================] - ETA: 0s - loss: 0.3798
Epoch 42: loss did not improve from 0.37837
7/7 [==============================] - 2s 279ms/step - loss: 0.3798 - val_loss: 0.5422 -
lr: 0.0010
Epoch 43/200
7/7 [==============================] - ETA: 0s - loss: 0.3863
Epoch 43: loss did not improve from 0.37837
7/7 [==============================] - 2s 279ms/step - loss: 0.3863 - val_loss: 0.5629 -
lr: 0.0010
Epoch 44/200
7/7 [==============================] - ETA: 0s - loss: 0.3764
Epoch 44: loss improved from 0.37837 to 0.37644, saving model to Sales.h5
7/7 [==============================] - 2s 295ms/step - loss: 0.3764 - val_loss: 0.5254 -
lr: 0.0010
Epoch 45/200
7/7 [==============================] - ETA: 0s - loss: 0.3748
Epoch 45: loss improved from 0.37644 to 0.37478, saving model to Sales.h5
7/7 [==============================] - 2s 285ms/step - loss: 0.3748 - val_loss: 0.5063 -
lr: 0.0010
Epoch 46/200
7/7 [==============================] - ETA: 0s - loss: 0.3689
Epoch 46: loss improved from 0.37478 to 0.36887, saving model to Sales.h5
7/7 [==============================] - 2s 289ms/step - loss: 0.3689 - val_loss: 0.5186 -
lr: 0.0010
Epoch 47/200
7/7 [==============================] - ETA: 0s - loss: 0.3733
Epoch 47: loss did not improve from 0.36887
7/7 [==============================] - 2s 290ms/step - loss: 0.3733 - val_loss: 0.5465 -
lr: 0.0010
Epoch 48/200
7/7 [==============================] - ETA: 0s - loss: 0.3775
Epoch 48: loss did not improve from 0.36887
7/7 [==============================] - 2s 286ms/step - loss: 0.3775 - val_loss: 0.5125 -
lr: 0.0010
Epoch 49/200
7/7 [==============================] - ETA: 0s - loss: 0.3743
Epoch 49: loss did not improve from 0.36887

Epoch 49: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
7/7 [==============================] - 2s 286ms/step - loss: 0.3743 - val_loss: 0.5079 -
lr: 0.0010
Epoch 50/200
7/7 [==============================] - ETA: 0s - loss: 0.3566
Epoch 50: loss improved from 0.36887 to 0.35663, saving model to Sales.h5
7/7 [==============================] - 2s 361ms/step - loss: 0.3566 - val_loss: 0.5073 -
lr: 2.0000e-04
Epoch 51/200
7/7 [==============================] - ETA: 0s - loss: 0.3521
Epoch 51: loss improved from 0.35663 to 0.35215, saving model to Sales.h5
7/7 [==============================] - 2s 349ms/step - loss: 0.3521 - val_loss: 0.5084 -
lr: 2.0000e-04
Epoch 52/200
7/7 [==============================] - ETA: 0s - loss: 0.3524
Epoch 52: loss did not improve from 0.35215
7/7 [==============================] - 2s 307ms/step - loss: 0.3524 - val_loss: 0.5090 -
lr: 2.0000e-04
Epoch 53/200
```

```
7/7 [==============================] - ETA: 0s - loss: 0.3488
Epoch 53: loss improved from 0.35215 to 0.34878, saving model to Sales.h5
7/7 [==============================] - 2s 279ms/step - loss: 0.3488 - val_loss: 0.5058 -
lr: 2.0000e-04
Epoch 54/200
7/7 [==============================] - ETA: 0s - loss: 0.3476
Epoch 54: loss improved from 0.34878 to 0.34760, saving model to Sales.h5
7/7 [==============================] - 2s 272ms/step - loss: 0.3476 - val_loss: 0.5042 -
lr: 2.0000e-04
Epoch 55/200
7/7 [==============================] - ETA: 0s - loss: 0.3463
Epoch 55: loss improved from 0.34760 to 0.34632, saving model to Sales.h5
7/7 [==============================] - 2s 279ms/step - loss: 0.3463 - val_loss: 0.5043 -
lr: 2.0000e-04
Epoch 56/200
7/7 [==============================] - ETA: 0s - loss: 0.3455
Epoch 56: loss improved from 0.34632 to 0.34546, saving model to Sales.h5
7/7 [==============================] - 2s 272ms/step - loss: 0.3455 - val_loss: 0.4987 -
lr: 2.0000e-04
Epoch 57/200
7/7 [==============================] - ETA: 0s - loss: 0.3444
Epoch 57: loss improved from 0.34546 to 0.34444, saving model to Sales.h5
7/7 [==============================] - 2s 303ms/step - loss: 0.3444 - val_loss: 0.4919 -
lr: 2.0000e-04
Epoch 58/200
7/7 [==============================] - ETA: 0s - loss: 0.3439
Epoch 58: loss improved from 0.34444 to 0.34385, saving model to Sales.h5
7/7 [==============================] - 2s 300ms/step - loss: 0.3439 - val_loss: 0.4816 -
lr: 2.0000e-04
Epoch 59/200
7/7 [==============================] - ETA: 0s - loss: 0.3443
Epoch 59: loss did not improve from 0.34385
7/7 [==============================] - 2s 262ms/step - loss: 0.3443 - val_loss: 0.4807 -
lr: 2.0000e-04
Epoch 60/200
7/7 [==============================] - ETA: 0s - loss: 0.3431
Epoch 60: loss improved from 0.34385 to 0.34314, saving model to Sales.h5
7/7 [==============================] - 2s 273ms/step - loss: 0.3431 - val_loss: 0.4866 -
lr: 2.0000e-04
Epoch 61/200
7/7 [==============================] - ETA: 0s - loss: 0.3424
Epoch 61: loss improved from 0.34314 to 0.34239, saving model to Sales.h5
7/7 [==============================] - 2s 293ms/step - loss: 0.3424 - val_loss: 0.4998 -
lr: 2.0000e-04
Epoch 62/200
7/7 [==============================] - ETA: 0s - loss: 0.3415
Epoch 62: loss improved from 0.34239 to 0.34146, saving model to Sales.h5
7/7 [==============================] - 2s 281ms/step - loss: 0.3415 - val_loss: 0.5103 -
lr: 2.0000e-04
Epoch 63/200
7/7 [==============================] - ETA: 0s - loss: 0.3463
Epoch 63: loss did not improve from 0.34146
7/7 [==============================] - 2s 287ms/step - loss: 0.3463 - val_loss: 0.5239 -
lr: 2.0000e-04
Epoch 64/200
7/7 [==============================] - ETA: 0s - loss: 0.3441
Epoch 64: loss did not improve from 0.34146
7/7 [==============================] - 2s 277ms/step - loss: 0.3441 - val_loss: 0.5205 -
lr: 2.0000e-04
Epoch 65/200
```

```
7/7 [==============================] - ETA: 0s - loss: 0.3417
Epoch 65: loss did not improve from 0.34146

Epoch 65: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-05.
7/7 [==============================] - 2s 274ms/step - loss: 0.3417 - val_loss: 0.5110 -
lr: 2.0000e-04
Epoch 66/200
7/7 [==============================] - ETA: 0s - loss: 0.3396
Epoch 66: loss improved from 0.34146 to 0.33964, saving model to Sales.h5
7/7 [==============================] - 2s 288ms/step - loss: 0.3396 - val_loss: 0.5102 -
lr: 4.0000e-05
Epoch 67/200
7/7 [==============================] - ETA: 0s - loss: 0.3390
Epoch 67: loss improved from 0.33964 to 0.33896, saving model to Sales.h5
7/7 [==============================] - 2s 274ms/step - loss: 0.3390 - val_loss: 0.5098 -
lr: 4.0000e-05
Epoch 68/200
7/7 [==============================] - ETA: 0s - loss: 0.3386
Epoch 68: loss improved from 0.33896 to 0.33855, saving model to Sales.h5
7/7 [==============================] - 2s 284ms/step - loss: 0.3386 - val_loss: 0.5071 -
lr: 4.0000e-05
Epoch 69/200
7/7 [==============================] - ETA: 0s - loss: 0.3380
Epoch 69: loss improved from 0.33855 to 0.33802, saving model to Sales.h5
7/7 [==============================] - 2s 279ms/step - loss: 0.3380 - val_loss: 0.5034 -
lr: 4.0000e-05
Epoch 70/200
7/7 [==============================] - ETA: 0s - loss: 0.3370
Epoch 70: loss improved from 0.33802 to 0.33702, saving model to Sales.h5
7/7 [==============================] - 2s 285ms/step - loss: 0.3370 - val_loss: 0.5022 -
lr: 4.0000e-05
Epoch 71/200
7/7 [==============================] - ETA: 0s - loss: 0.3366
Epoch 71: loss improved from 0.33702 to 0.33661, saving model to Sales.h5
7/7 [==============================] - 2s 277ms/step - loss: 0.3366 - val_loss: 0.5013 -
lr: 4.0000e-05
Epoch 72/200
7/7 [==============================] - ETA: 0s - loss: 0.3364
Epoch 72: loss improved from 0.33661 to 0.33641, saving model to Sales.h5
7/7 [==============================] - 2s 273ms/step - loss: 0.3364 - val_loss: 0.5004 -
lr: 4.0000e-05
Epoch 73/200
7/7 [==============================] - ETA: 0s - loss: 0.3364
Epoch 73: loss did not improve from 0.33641

Epoch 73: ReduceLROnPlateau reducing learning rate to 8.000000525498762e-06.
7/7 [==============================] - 2s 270ms/step - loss: 0.3364 - val_loss: 0.4982 -
lr: 4.0000e-05
Epoch 74/200
7/7 [==============================] - ETA: 0s - loss: 0.3358
Epoch 74: loss improved from 0.33641 to 0.33584, saving model to Sales.h5
7/7 [==============================] - 2s 277ms/step - loss: 0.3358 - val_loss: 0.4979 -
lr: 8.0000e-06
Epoch 75/200
7/7 [==============================] - ETA: 0s - loss: 0.3358
Epoch 75: loss improved from 0.33584 to 0.33583, saving model to Sales.h5
7/7 [==============================] - 2s 280ms/step - loss: 0.3358 - val_loss: 0.4974 -
lr: 8.0000e-06
Epoch 76/200
7/7 [==============================] - ETA: 0s - loss: 0.3358
```

```
Epoch 76: loss improved from 0.33583 to 0.33576, saving model to Sales.h5
7/7 [==============================] - 2s 278ms/step - loss: 0.3358 - val_loss: 0.4973 -
lr: 8.0000e-06
Epoch 77/200
7/7 [==============================] - ETA: 0s - loss: 0.3359
Epoch 77: loss did not improve from 0.33576

Epoch 77: ReduceLROnPlateau reducing learning rate to 1.6000001778593287e-06.
7/7 [==============================] - 2s 262ms/step - loss: 0.3359 - val_loss: 0.4973 -
lr: 8.0000e-06
Epoch 78/200
7/7 [==============================] - ETA: 0s - loss: 0.3357
Epoch 78: loss improved from 0.33576 to 0.33568, saving model to Sales.h5
7/7 [==============================] - 2s 300ms/step - loss: 0.3357 - val_loss: 0.4974 -
lr: 1.6000e-06
Epoch 79/200
7/7 [==============================] - ETA: 0s - loss: 0.3357
Epoch 79: loss did not improve from 0.33568
7/7 [==============================] - 2s 313ms/step - loss: 0.3357 - val_loss: 0.4974 -
lr: 1.6000e-06
Epoch 80/200
7/7 [==============================] - ETA: 0s - loss: 0.3357
Epoch 80: loss improved from 0.33568 to 0.33566, saving model to Sales.h5

Epoch 80: ReduceLROnPlateau reducing learning rate to 3.200000264769187e-07.
7/7 [==============================] - 2s 288ms/step - loss: 0.3357 - val_loss: 0.4974 -
lr: 1.6000e-06
Epoch 81/200
7/7 [==============================] - ETA: 0s - loss: 0.3357
Epoch 81: loss improved from 0.33566 to 0.33565, saving model to Sales.h5
7/7 [==============================] - 2s 290ms/step - loss: 0.3357 - val_loss: 0.4974 -
lr: 3.2000e-07
Epoch 82/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 82: loss improved from 0.33565 to 0.33565, saving model to Sales.h5
7/7 [==============================] - 2s 283ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 3.2000e-07
Epoch 83/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 83: loss improved from 0.33565 to 0.33565, saving model to Sales.h5

Epoch 83: ReduceLROnPlateau reducing learning rate to 6.400000529538374e-08.
7/7 [==============================] - 2s 335ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 3.2000e-07
Epoch 84/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 84: loss improved from 0.33565 to 0.33564, saving model to Sales.h5
7/7 [==============================] - 2s 324ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 6.4000e-08
Epoch 85/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 85: loss improved from 0.33564 to 0.33564, saving model to Sales.h5
7/7 [==============================] - 2s 296ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 6.4000e-08
Epoch 86/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 86: loss improved from 0.33564 to 0.33564, saving model to Sales.h5

Epoch 86: ReduceLROnPlateau reducing learning rate to 1.2800001059076749e-08.
7/7 [==============================] - 2s 303ms/step - loss: 0.3356 - val_loss: 0.4974 -
```

```
lr: 6.4000e-08
Epoch 87/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 87: loss improved from 0.33564 to 0.33564, saving model to Sales.h5
7/7 [==============================] - 2s 293ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 1.2800e-08
Epoch 88/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 88: loss improved from 0.33564 to 0.33564, saving model to Sales.h5
7/7 [==============================] - 2s 291ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 1.2800e-08
Epoch 89/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 89: loss improved from 0.33564 to 0.33564, saving model to Sales.h5

Epoch 89: ReduceLROnPlateau reducing learning rate to 2.5600002118153498e-09.
7/7 [==============================] - 2s 299ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 1.2800e-08
Epoch 90/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 90: loss improved from 0.33564 to 0.33564, saving model to Sales.h5
7/7 [==============================] - 2s 303ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 2.5600e-09
Epoch 91/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 91: loss did not improve from 0.33564
7/7 [==============================] - 2s 292ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 2.5600e-09
Epoch 92/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 92: loss did not improve from 0.33564

Epoch 92: ReduceLROnPlateau reducing learning rate to 5.1200004236307e-10.
7/7 [==============================] - 2s 285ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 2.5600e-09
Epoch 93/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 93: loss did not improve from 0.33564
7/7 [==============================] - 2s 263ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 5.1200e-10
Epoch 94/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 94: loss did not improve from 0.33564
7/7 [==============================] - 2s 262ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 5.1200e-10
Epoch 95/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 95: loss did not improve from 0.33564

Epoch 95: ReduceLROnPlateau reducing learning rate to 1.0240001069306004e-10.
7/7 [==============================] - 2s 276ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 5.1200e-10
Epoch 96/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 96: loss did not improve from 0.33564
7/7 [==============================] - 2s 282ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 1.0240e-10
Epoch 97/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 97: loss did not improve from 0.33564
```

```
7/7 [==============================] - 2s 275ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 1.0240e-10
Epoch 98/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 98: loss improved from 0.33564 to 0.33564, saving model to Sales.h5

Epoch 98: ReduceLROnPlateau reducing learning rate to 2.0480002416167767e-11.
7/7 [==============================] - 2s 277ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 1.0240e-10
Epoch 99/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 99: loss did not improve from 0.33564
7/7 [==============================] - 2s 265ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 2.0480e-11
Epoch 100/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 100: loss did not improve from 0.33564
7/7 [==============================] - 2s 262ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 2.0480e-11
Epoch 101/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 101: loss did not improve from 0.33564

Epoch 101: ReduceLROnPlateau reducing learning rate to 4.096000622011431e-12.
7/7 [==============================] - 2s 273ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 2.0480e-11
Epoch 102/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 102: loss did not improve from 0.33564
7/7 [==============================] - 2s 268ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 4.0960e-12
Epoch 103/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 103: loss did not improve from 0.33564
7/7 [==============================] - 2s 277ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 4.0960e-12
Epoch 104/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 104: loss did not improve from 0.33564

Epoch 104: ReduceLROnPlateau reducing learning rate to 8.192000897078167e-13.
7/7 [==============================] - 2s 291ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 4.0960e-12
Epoch 105/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 105: loss did not improve from 0.33564
7/7 [==============================] - 2s 269ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 8.1920e-13
Epoch 106/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 106: loss did not improve from 0.33564
7/7 [==============================] - 2s 265ms/step - loss: 0.3356 - val_loss: 0.4974 -
lr: 8.1920e-13
Epoch 107/200
7/7 [==============================] - ETA: 0s - loss: 0.3356
Epoch 107: loss did not improve from 0.33564
Restoring model weights from the end of the best epoch: 98.

Epoch 107: ReduceLROnPlateau reducing learning rate to 1.6384001360475466e-13.
7/7 [==============================] - 2s 284ms/step - loss: 0.3356 - val_loss: 0.4974 -
```

```
lr: 8.1920e-13
Epoch 107: early stopping
```

In [124…

```
train_predict = model.predict(x_train)
test_predict = model.predict(x_test)
```

In [125…

```
#train_predict = std.inverse_transform(train_predict)
train_predict = train_predict.reshape(len(train_predict))
#test_predict = std.inverse_transform(test_predict)
test_predict = test_predict.reshape(len(test_predict))
```

In [126…

```
# inversion of normalisation
train_predict = train_predict*std.values + mean.values
test_predict = test_predict*std.values + mean.values
y_train = y_train*std.values + mean.values
y_test = y_test*std.values + mean.values
```

In [127…

```
RMSE_LSTM = math.sqrt(mean_squared_error(y_train,train_predict))
RMSE_LSTM
```

Out[127…

```
591.3372080725682
```

In [128…

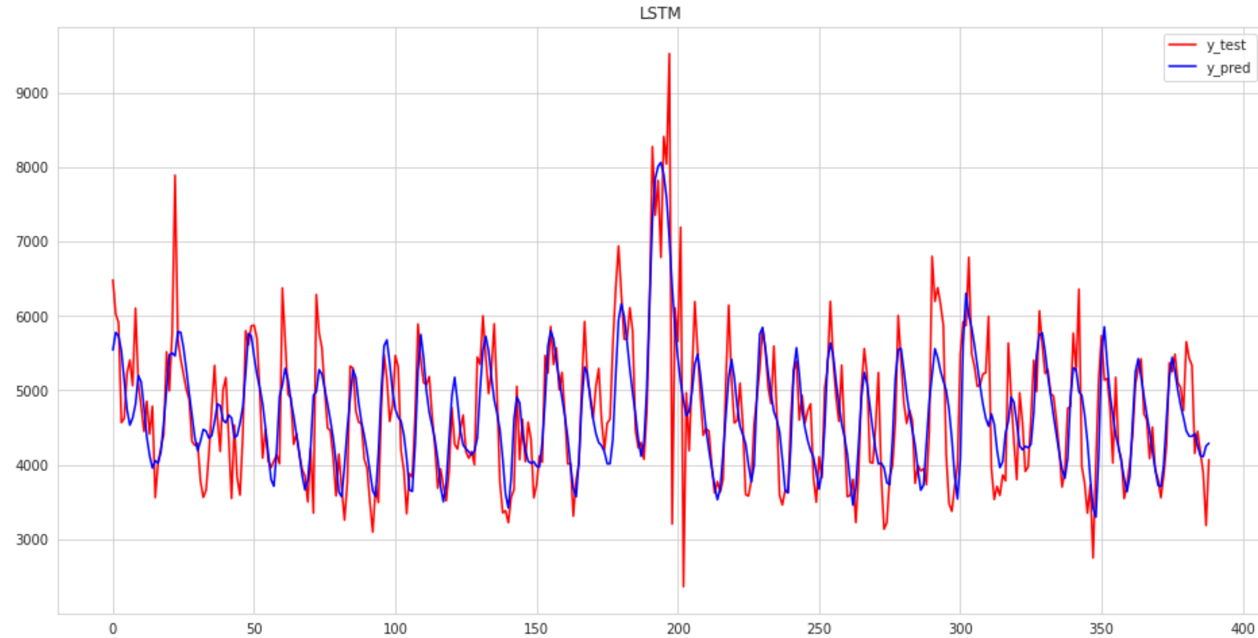
```
RMSE_LSTM = math.sqrt(mean_squared_error(y_test,test_predict))
RMSE_LSTM
```

Out[128…

```
719.8649163443836
```

In [135…

```
plt.figure(figsize = (16,8))
plt.plot(y_train, label = 'y_test',color ='r')
plt.plot(train_predict,label = 'y_pred',color='b')
plt.title('LSTM')
plt.legend()
plt.show()
```



In [136…

```
# Conclusion : The time series model performs better than the LSTM model
```

2. Comment on the behavior of all the models you have built so far

In [138…

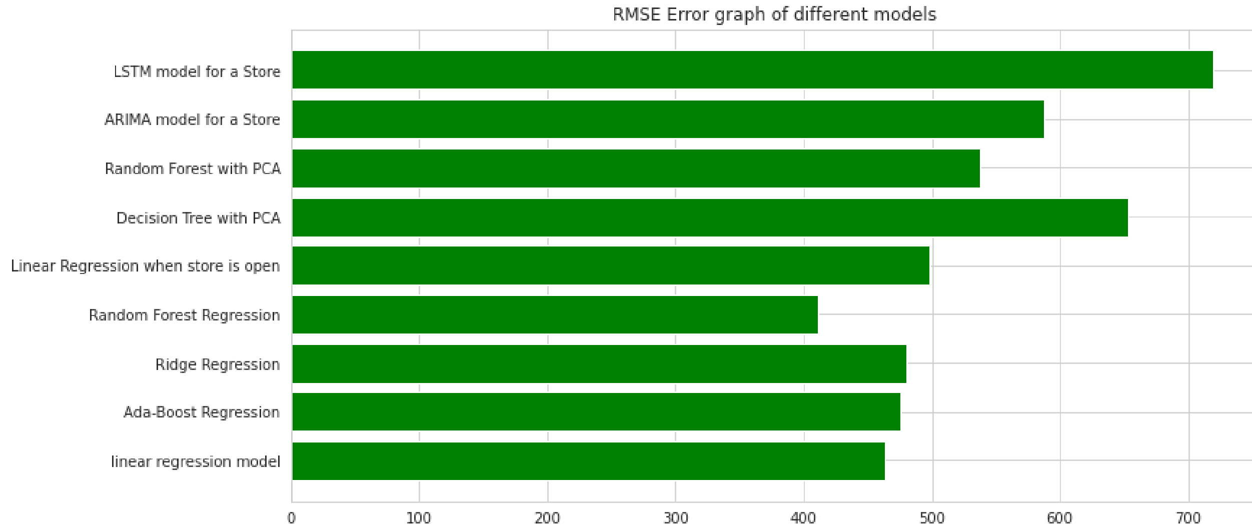
```
models_error = [[error_output_lr.RMSE.mean(), 'linear regression model']] # linear regr
models_error.append([error_output_ada.RMSE.mean(),'Ada-Boost Regression']) # Ada-Boost
models_error.append([error_output_rdg.RMSE.mean(),'Ridge Regression']) # Ridge Regressi
models_error.append([error_output_rdm.RMSE.mean(),'Random Forest Regression']) # Random
models_error.append([error_output_lrc.RMSE.mean(),'Linear Regression when store is open
models_error.append([error_output_dt_pca.RMSE.mean(),'Decision Tree with PCA'])
models_error.append([error_output_rdm_pca.RMSE.mean(),'Random Forest with PCA'])
models_error.append([RMSE_ARIMA,'ARIMA model for a Store'])
models_error.append([RMSE_LSTM, 'LSTM model for a Store'])
```

In [139…

```
models_error = pd.DataFrame(models_error)
```

In [141…

```
plt.figure(figsize=(12,6))
plt.barh(models_error[1],models_error[0], color = "g")
plt.title('RMSE Error graph of different models')
plt.show()
```



In [142…

```
# Conclusion : This gives us the idea that the Random Forest is the best amonst all the
```

In [143…

```
# Finding how many clusters are possible
```

3. Cluster stores using sales and customer visits as features. Find out how many clusters or groups are possible. Also visualize the results.

In [144…

```
cluster_data = train_data[train_data.Open==1]
cluster_data = cluster_data[['Store','Sales','Customers']]
cluster_data.head()
```

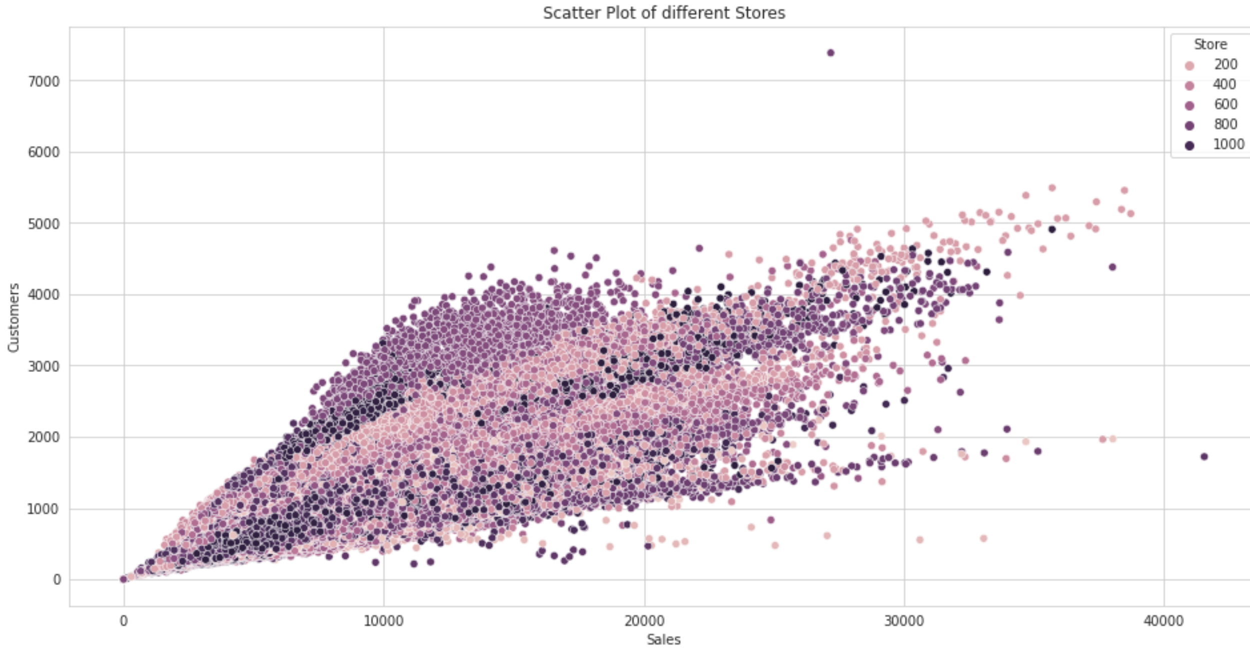
Out[144…

	Store	Sales	Customers
0	1	5735	568
1	2	9863	877
2	3	13261	1072
3	4	13106	1488
4	5	6635	645

In [ ]:

In [145…

```
plt.figure(figsize=(16,8))
sns.scatterplot(data=cluster_data,x='Sales',y='Customers', hue = 'Store')
plt.title('Scatter Plot of different Stores')
plt.show()
```



In [146…

```
kmeans = KMeans(n_clusters=5, random_state=24).fit(np.array(cluster_data[['Sales','Cust
cluster_data['forecast'] = kmeans.predict(np.array(cluster_data[['Sales','Customers']])
cluster_data.head()
```

Out[146…

	Store	Sales	Customers	forecast
0	1	5735	568	2
1	2	9863	877	4
2	3	13261	1072	1
3	4	13106	1488	1
4	5	6635	645	2

In [147…

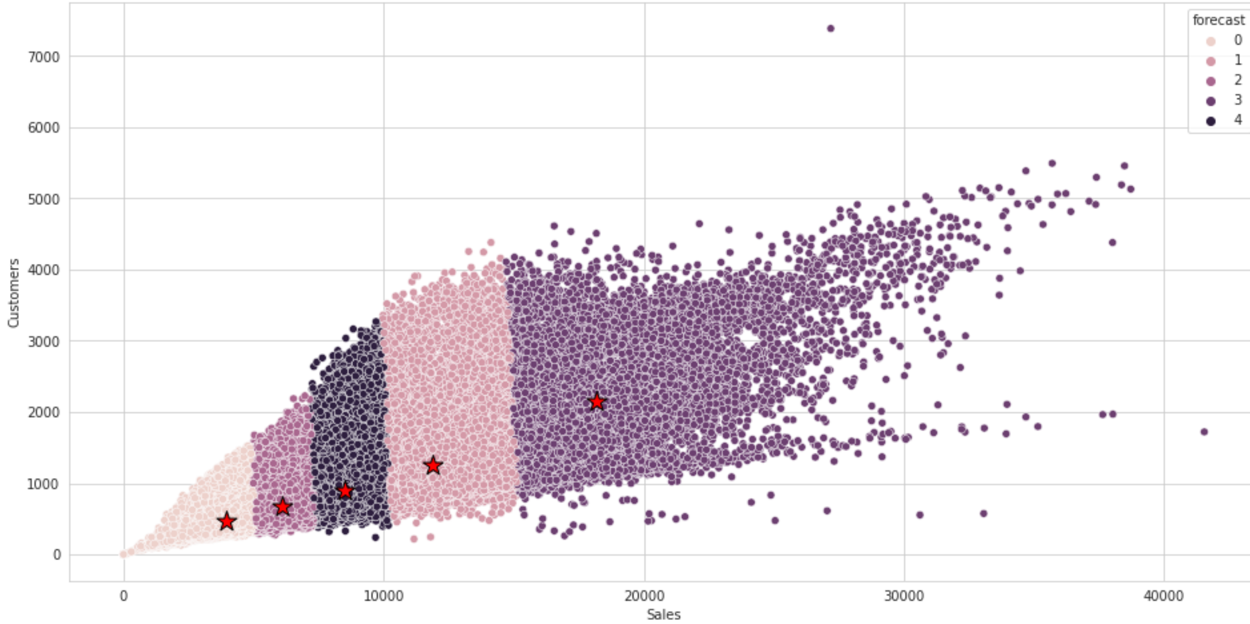
```
kmeans.labels_
kmeans.cluster_centers_
```

Out[147…

```
array([[ 3927.10606094,   465.94338323],
       [11874.9189934 ,  1259.72065898],
       [ 6120.57968859,   674.83040976],
       [18198.68233558,  2146.44714376],
       [ 8526.27176071,   897.11022624]])
```

In [148…

```
plt.figure(figsize=(16,8))
sns.scatterplot(data=cluster_data,x='Sales',y='Customers', hue = 'forecast')
plt.scatter(
    kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
    s=250, marker='*',
    c='red', edgecolor='black',
    label='centroids'
)
plt.show()
```



In [26]:
```
# Lets separate the prediction models for each clusters.
```

4. Is it possible to have separate prediction models for each cluster? Compare results with the previous models.

In [150…
```
cluster_data = train_data[train_data.Open == 1]
cluster_data.drop('Open',axis=1,inplace=True)
kmeans = KMeans(n_clusters=5, random_state=24).fit(np.array(cluster_data[['Sales','Cust
cluster_data['forecast'] = kmeans.predict(np.array(cluster_data[['Sales','Customers']])
cluster_data.head()
```

Out[150…

	Store	DayOfWeek	Sales	Customers	Promo	StateHoliday	SchoolHoliday	forecast
0	1	2	5735	568	1	0	0	2
1	2	2	9863	877	1	0	0	4
2	3	2	13261	1072	1	0	1	1
3	4	2	13106	1488	1	0	0	1
4	5	2	6635	645	1	0	0	2

In [151…
```
cluster_data.drop('Store',axis=1,inplace=True)
RMSE_cluster_rdm = []
MAE_cluster_rdm=[]
for clust in range(5):
    data = cluster_data[cluster_data.forecast==clust]
    data.drop('forecast',axis=1,inplace=True)
    y=np.array(data['Sales'])
    x=np.array(data.drop('Sales',axis=1))
    y_test,y_pred = random_forest(np.array(x),np.array(y))
```

```
    RMSE_1,MAE_1 = error_cal(y_test,y_pred)
    RMSE_cluster_rdm.append(RMSE_1)
    MAE_cluster_rdm.append(MAE_1)
```

In [152…

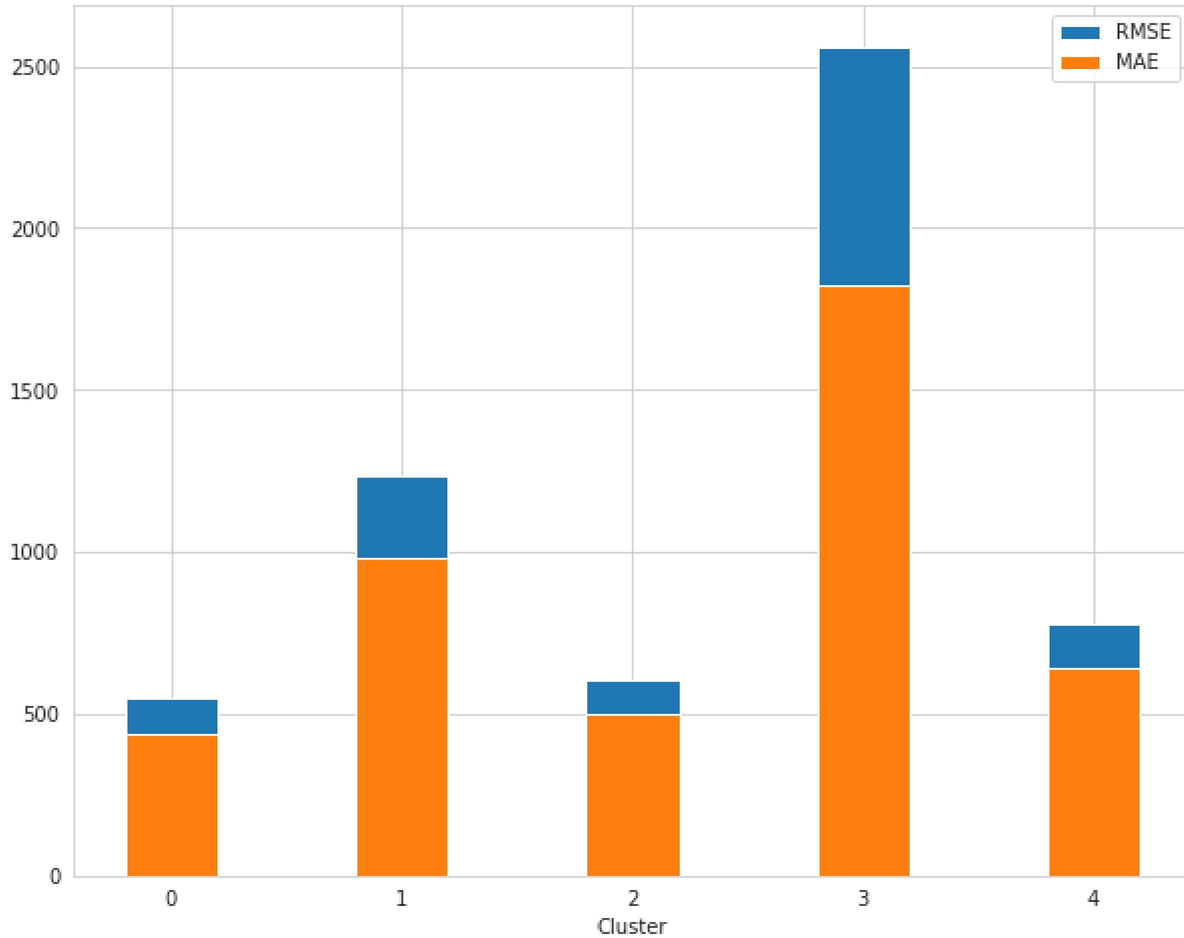
```
cluster = [0,1,2,3,4]
error_output_cluster_rdm = pd.DataFrame()
error_output_cluster_rdm['cluster'] = cluster
error_output_cluster_rdm['RMSE'] = RMSE_cluster_rdm
error_output_cluster_rdm['MAE'] = MAE_cluster_rdm
error_output_cluster_rdm
```

Out[152…

	cluster	RMSE	MAE
0	0	548.076478	436.845845
1	1	1234.593415	982.949886
2	2	601.942714	498.720675
3	3	2562.038866	1826.680012
4	4	777.805035	640.092054

In [153…

```
plt.figure(figsize=(10,8))
plt.bar(x=error_output_cluster_rdm.cluster,height=error_output_cluster_rdm.RMSE,label =
plt.bar(x=error_output_cluster_rdm.cluster,height=error_output_cluster_rdm.MAE,label =
plt.xlabel('Cluster')
plt.legend()
plt.show()
```



In [154…

```
#since data is not suitable for clustring, we can not separate data into different clus
#so while predicting sales based on clusters, it shows unpredictible result (RMSE, and 
```

Applying ANN:

1. Use ANN (Artificial Neural Network) to predict Store Sales. a) Fine-tune number of layers,
   b) Number of Neurons in each layers.
   c) Experiment in batch-size.
   d) Experiment with number of epochs. Carefully observe the loss and accuracy? What are the observations?
   e) Play with different Learning Rate variants of Gradient Descent like Adam, SGD, RMS-prop.
   f) Which activation performs best for this use case and why?
   g) Check how it performed in the dataset, calculate RMSE.

In [156…

```
train_data.head()
```

Out[156…

	Store	DayOfWeek	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	2	5735	568	1	1	0	0
1	2	2	9863	877	1	1	0	0

	Store	DayOfWeek	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
2	3	2	13261	1072	1	1	0	1
3	4	2	13106	1488	1	1	0	0
4	5	2	6635	645	1	1	0	0

In [ ]:

```

```

In [157…

```
train_data = train_data[train_data.Store<=100]
train_data = train_data[train_data.Open == 1]
train_data.reset_index(drop=True, inplace=True)
y = train_data['Sales']
x = train_data.drop(['Sales','Open'],axis=1)
std = StandardScaler()
x = std.fit_transform(x)
```

In [158…

```
x_train,x_test,y_train,y_test = train_test_split(np.array(x),np.array(y),random_state=4
```

In [159…

```
model_1 = Sequential()
model_1.add(layers.Dense(32, activation='elu', input_shape = (x_train.shape[1],)))
model_1.add(layers.Dense(64, activation='elu'))
model_1.add(layers.Dense(64, activation='elu'))
model_1.add(layers.BatchNormalization())
## block 2
model_1.add(layers.Dense(128, activation='elu'))
model_1.add(layers.Dense(128, activation='elu'))
model_1.add(layers.BatchNormalization())
## block 3
model_1.add(layers.Dense(256, activation='elu'))
model_1.add(layers.Dense(256, activation='elu'))
model_1.add(layers.BatchNormalization())
## block 4
model_1.add(layers.Dense(128, activation='elu'))
model_1.add(layers.Dense(128, activation='elu'))
model_1.add(layers.BatchNormalization())
## block 5
model_1.add(layers.Dense(64, activation='elu'))
model_1.add(layers.Dense(64, activation='elu'))
model_1.add(layers.Dense(32, activation='elu'))
model_1.add(layers.Dense(1))
```

In [160…

```
model_1.compile(loss='mse',optimizer = Adam(learning_rate=0.001), metrics=['mae'])
```

In [161…

```
checkpoint = ModelCheckpoint('Sales_ann.h5',
monitor='loss',
mode=min,
save_best_only=True,
verbose=1)
early_stopping = EarlyStopping(monitor='loss',
patience=9,
```

```
min_delta=0,
restore_best_weights=True,
verbose=1)
Reduce_ler_rate = ReduceLROnPlateau(monitor='loss',
factor=0.2,
patience=3,
verbose=1,
min_delta=0.001)
callback = [checkpoint,early_stopping,Reduce_ler_rate]
```

```
WARNING:tensorflow:ModelCheckpoint mode <built-in function min> is unknown, fallback to
auto mode.
```

In [162…

```
history = model_1.fit(x_train,y_train,epochs=100,batch_size=20,verbose=1,callbacks=call
```

```
Epoch 1/100
2552/2552 [==============================] - ETA: 0s - loss: 4811496.5000 - mae: 1533.46
56
Epoch 1: loss improved from inf to 4811496.50000, saving model to Sales_ann.h5
2552/2552 [==============================] - 28s 10ms/step - loss: 4811496.5000 - mae: 1
533.4656 - lr: 0.0010
Epoch 2/100
2550/2552 [============================>.] - ETA: 0s - loss: 2287976.7500 - mae: 1132.71
69
Epoch 2: loss improved from 4811496.50000 to 2294520.50000, saving model to Sales_ann.h5
2552/2552 [==============================] - 23s 9ms/step - loss: 2294520.5000 - mae: 11
33.1794 - lr: 0.0010
Epoch 3/100
2551/2552 [============================>.] - ETA: 0s - loss: 2039987.6250 - mae: 1068.38
06
Epoch 3: loss improved from 2294520.50000 to 2040060.75000, saving model to Sales_ann.h5
2552/2552 [==============================] - 23s 9ms/step - loss: 2040060.7500 - mae: 10
68.4197 - lr: 0.0010
Epoch 4/100
2547/2552 [============================>.] - ETA: 0s - loss: 1847703.0000 - mae: 1023.46
08
Epoch 4: loss improved from 2040060.75000 to 1848382.50000, saving model to Sales_ann.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 1848382.5000 - mae: 10
23.4998 - lr: 0.0010
Epoch 5/100
2550/2552 [============================>.] - ETA: 0s - loss: 1749290.2500 - mae: 996.959
8
Epoch 5: loss improved from 1848382.50000 to 1749494.62500, saving model to Sales_ann.h5
2552/2552 [==============================] - 18s 7ms/step - loss: 1749494.6250 - mae: 99
7.0168 - lr: 0.0010
Epoch 6/100
2546/2552 [============================>.] - ETA: 0s - loss: 1712536.7500 - mae: 986.303
0
Epoch 6: loss improved from 1749494.62500 to 1712476.87500, saving model to Sales_ann.h5
2552/2552 [==============================] - 15s 6ms/step - loss: 1712476.8750 - mae: 98
6.1966 - lr: 0.0010
Epoch 7/100
2546/2552 [============================>.] - ETA: 0s - loss: 1672813.6250 - mae: 972.279
4
Epoch 7: loss improved from 1712476.87500 to 1674769.87500, saving model to Sales_ann.h5
2552/2552 [==============================] - 18s 7ms/step - loss: 1674769.8750 - mae: 97
2.5356 - lr: 0.0010
Epoch 8/100
2545/2552 [============================>.] - ETA: 0s - loss: 1640951.2500 - mae: 963.293
```

```
8
Epoch 8: loss improved from 1674769.87500 to 1641398.37500, saving model to Sales_ann.h5
2552/2552 [==============================] - 14s 6ms/step - loss: 1641398.3750 - mae: 96
3.4650 - lr: 0.0010
Epoch 9/100
2549/2552 [============================>.] - ETA: 0s - loss: 1621227.0000 - mae: 955.424
6
Epoch 9: loss improved from 1641398.37500 to 1621098.50000, saving model to Sales_ann.h5
2552/2552 [==============================] - 17s 7ms/step - loss: 1621098.5000 - mae: 95
5.4664 - lr: 0.0010
Epoch 10/100
2548/2552 [============================>.] - ETA: 0s - loss: 1601600.7500 - mae: 947.308
3
Epoch 10: loss improved from 1621098.50000 to 1602030.25000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 14s 6ms/step - loss: 1602030.2500 - mae: 94
7.5041 - lr: 0.0010
Epoch 11/100
2545/2552 [============================>.] - ETA: 0s - loss: 1578833.8750 - mae: 940.607
8
Epoch 11: loss improved from 1602030.25000 to 1579421.00000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 17s 7ms/step - loss: 1579421.0000 - mae: 94
0.8808 - lr: 0.0010
Epoch 12/100
2549/2552 [============================>.] - ETA: 0s - loss: 1554608.1250 - mae: 933.575
1
Epoch 12: loss improved from 1579421.00000 to 1554666.87500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 17s 7ms/step - loss: 1554666.8750 - mae: 93
3.5635 - lr: 0.0010
Epoch 13/100
2549/2552 [============================>.] - ETA: 0s - loss: 1535901.6250 - mae: 925.511
0
Epoch 13: loss improved from 1554666.87500 to 1536294.87500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 15s 6ms/step - loss: 1536294.8750 - mae: 92
5.6436 - lr: 0.0010
Epoch 14/100
2544/2552 [============================>.] - ETA: 0s - loss: 1526264.3750 - mae: 923.410
9
Epoch 14: loss improved from 1536294.87500 to 1525366.25000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 16s 6ms/step - loss: 1525366.2500 - mae: 92
3.1331 - lr: 0.0010
Epoch 15/100
2551/2552 [============================>.] - ETA: 0s - loss: 1490723.5000 - mae: 912.266
4
Epoch 15: loss improved from 1525366.25000 to 1490681.87500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 16s 6ms/step - loss: 1490681.8750 - mae: 91
2.2600 - lr: 0.0010
Epoch 16/100
2550/2552 [============================>.] - ETA: 0s - loss: 1486249.6250 - mae: 910.328
9
Epoch 16: loss improved from 1490681.87500 to 1486318.37500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 20s 8ms/step - loss: 1486318.3750 - mae: 91
0.3878 - lr: 0.0010
Epoch 17/100
```

```
2551/2552 [============================>.] - ETA: 0s - loss: 1475540.3750 - mae: 907.315
1
Epoch 17: loss improved from 1486318.37500 to 1475963.50000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 17s 7ms/step - loss: 1475963.5000 - mae: 90
7.4292 - lr: 0.0010
Epoch 18/100
2551/2552 [============================>.] - ETA: 0s - loss: 1447937.5000 - mae: 896.462
2
Epoch 18: loss improved from 1475963.50000 to 1448035.00000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 17s 6ms/step - loss: 1448035.0000 - mae: 89
6.5024 - lr: 0.0010
Epoch 19/100
2545/2552 [============================>.] - ETA: 0s - loss: 1445278.3750 - mae: 895.281
1
Epoch 19: loss improved from 1448035.00000 to 1444953.62500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 21s 8ms/step - loss: 1444953.6250 - mae: 89
5.2623 - lr: 0.0010
Epoch 20/100
2551/2552 [============================>.] - ETA: 0s - loss: 1417170.3750 - mae: 888.346
6
Epoch 20: loss improved from 1444953.62500 to 1417305.62500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 18s 7ms/step - loss: 1417305.6250 - mae: 88
8.3780 - lr: 0.0010
Epoch 21/100
2548/2552 [============================>.] - ETA: 0s - loss: 1407165.7500 - mae: 883.247
4
Epoch 21: loss improved from 1417305.62500 to 1406796.25000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 20s 8ms/step - loss: 1406796.2500 - mae: 88
3.1383 - lr: 0.0010
Epoch 22/100
2552/2552 [==============================] - ETA: 0s - loss: 1393998.3750 - mae: 880.606
0
Epoch 22: loss improved from 1406796.25000 to 1393998.37500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 19s 8ms/step - loss: 1393998.3750 - mae: 88
0.6060 - lr: 0.0010
Epoch 23/100
2549/2552 [============================>.] - ETA: 0s - loss: 1377180.7500 - mae: 874.529
1
Epoch 23: loss improved from 1393998.37500 to 1378346.12500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 18s 7ms/step - loss: 1378346.1250 - mae: 87
4.7028 - lr: 0.0010
Epoch 24/100
2550/2552 [============================>.] - ETA: 0s - loss: 1361311.8750 - mae: 868.541
4
Epoch 24: loss improved from 1378346.12500 to 1361911.62500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 21s 8ms/step - loss: 1361911.6250 - mae: 86
8.6603 - lr: 0.0010
Epoch 25/100
2551/2552 [============================>.] - ETA: 0s - loss: 1341393.0000 - mae: 864.186
0
Epoch 25: loss improved from 1361911.62500 to 1341399.00000, saving model to Sales_ann.h
5
```

```
2552/2552 [==============================] - 19s 8ms/step - loss: 1341399.0000 - mae: 86
4.1943 - lr: 0.0010
Epoch 26/100
2548/2552 [============================>.] - ETA: 0s - loss: 1324210.3750 - mae: 858.676
9
Epoch 26: loss improved from 1341399.00000 to 1323508.87500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 21s 8ms/step - loss: 1323508.8750 - mae: 85
8.4783 - lr: 0.0010
Epoch 27/100
2550/2552 [============================>.] - ETA: 0s - loss: 1312099.1250 - mae: 855.054
5
Epoch 27: loss improved from 1323508.87500 to 1312221.62500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 19s 8ms/step - loss: 1312221.6250 - mae: 85
5.0746 - lr: 0.0010
Epoch 28/100
2546/2552 [============================>.] - ETA: 0s - loss: 1306640.6250 - mae: 851.746
5
Epoch 28: loss improved from 1312221.62500 to 1306531.37500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 20s 8ms/step - loss: 1306531.3750 - mae: 85
1.7335 - lr: 0.0010
Epoch 29/100
2549/2552 [============================>.] - ETA: 0s - loss: 1290389.3750 - mae: 845.016
1
Epoch 29: loss improved from 1306531.37500 to 1290178.12500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 19s 8ms/step - loss: 1290178.1250 - mae: 84
4.9556 - lr: 0.0010
Epoch 30/100
2549/2552 [============================>.] - ETA: 0s - loss: 1282642.0000 - mae: 844.552
0
Epoch 30: loss improved from 1290178.12500 to 1282788.75000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 18s 7ms/step - loss: 1282788.7500 - mae: 84
4.6745 - lr: 0.0010
Epoch 31/100
2552/2552 [==============================] - ETA: 0s - loss: 1260311.6250 - mae: 834.975
0
Epoch 31: loss improved from 1282788.75000 to 1260311.62500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 20s 8ms/step - loss: 1260311.6250 - mae: 83
4.9750 - lr: 0.0010
Epoch 32/100
2546/2552 [============================>.] - ETA: 0s - loss: 1256223.1250 - mae: 832.631
0
Epoch 32: loss improved from 1260311.62500 to 1255763.75000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 20s 8ms/step - loss: 1255763.7500 - mae: 83
2.5312 - lr: 0.0010
Epoch 33/100
2548/2552 [============================>.] - ETA: 0s - loss: 1236603.5000 - mae: 826.627
1
Epoch 33: loss improved from 1255763.75000 to 1236572.75000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 23s 9ms/step - loss: 1236572.7500 - mae: 82
6.6063 - lr: 0.0010
Epoch 34/100
2548/2552 [============================>.] - ETA: 0s - loss: 1226658.8750 - mae: 823.344
```

```
3
Epoch 34: loss improved from 1236572.75000 to 1226149.62500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 25s 10ms/step - loss: 1226149.6250 - mae: 8
23.2411 - lr: 0.0010
Epoch 35/100
2545/2552 [============================>.] - ETA: 0s - loss: 1211172.3750 - mae: 818.406
7
Epoch 35: loss improved from 1226149.62500 to 1211868.62500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 18s 7ms/step - loss: 1211868.6250 - mae: 81
8.5053 - lr: 0.0010
Epoch 36/100
2546/2552 [============================>.] - ETA: 0s - loss: 1197265.3750 - mae: 810.925
0
Epoch 36: loss improved from 1211868.62500 to 1197065.87500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 18s 7ms/step - loss: 1197065.8750 - mae: 81
0.9139 - lr: 0.0010
Epoch 37/100
2545/2552 [============================>.] - ETA: 0s - loss: 1194429.1250 - mae: 810.606
6
Epoch 37: loss improved from 1197065.87500 to 1194462.12500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 19s 7ms/step - loss: 1194462.1250 - mae: 81
0.4910 - lr: 0.0010
Epoch 38/100
2550/2552 [============================>.] - ETA: 0s - loss: 1175647.5000 - mae: 801.601
0
Epoch 38: loss improved from 1194462.12500 to 1175667.12500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 16s 6ms/step - loss: 1175667.1250 - mae: 80
1.6624 - lr: 0.0010
Epoch 39/100
2551/2552 [============================>.] - ETA: 0s - loss: 1170516.8750 - mae: 800.299
3
Epoch 39: loss improved from 1175667.12500 to 1170809.25000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 19s 7ms/step - loss: 1170809.2500 - mae: 80
0.3691 - lr: 0.0010
Epoch 40/100
2546/2552 [============================>.] - ETA: 0s - loss: 1156973.0000 - mae: 792.133
3
Epoch 40: loss improved from 1170809.25000 to 1156876.75000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 21s 8ms/step - loss: 1156876.7500 - mae: 79
2.1372 - lr: 0.0010
Epoch 41/100
2552/2552 [==============================] - ETA: 0s - loss: 1146776.6250 - mae: 789.976
2
Epoch 41: loss improved from 1156876.75000 to 1146776.62500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 20s 8ms/step - loss: 1146776.6250 - mae: 78
9.9762 - lr: 0.0010
Epoch 42/100
2549/2552 [============================>.] - ETA: 0s - loss: 1139269.6250 - mae: 784.363
6
Epoch 42: loss improved from 1146776.62500 to 1138768.50000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 23s 9ms/step - loss: 1138768.5000 - mae: 78
```

```
4.1927 - lr: 0.0010
Epoch 43/100
2550/2552 [============================>.] - ETA: 0s - loss: 1129516.8750 - mae: 783.097
8
Epoch 43: loss improved from 1138768.50000 to 1129542.87500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 24s 9ms/step - loss: 1129542.8750 - mae: 78
3.0735 - lr: 0.0010
Epoch 44/100
2552/2552 [==============================] - ETA: 0s - loss: 1126799.2500 - mae: 781.124
9
Epoch 44: loss improved from 1129542.87500 to 1126799.25000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 29s 11ms/step - loss: 1126799.2500 - mae: 7
81.1249 - lr: 0.0010
Epoch 45/100
2550/2552 [============================>.] - ETA: 0s - loss: 1115947.3750 - mae: 776.682
1
Epoch 45: loss improved from 1126799.25000 to 1116290.12500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 28s 11ms/step - loss: 1116290.1250 - mae: 7
76.8010 - lr: 0.0010
Epoch 46/100
2549/2552 [============================>.] - ETA: 0s - loss: 1101156.3750 - mae: 769.632
0
Epoch 46: loss improved from 1116290.12500 to 1100991.37500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 23s 9ms/step - loss: 1100991.3750 - mae: 76
9.5712 - lr: 0.0010
Epoch 47/100
2546/2552 [============================>.] - ETA: 0s - loss: 1089274.2500 - mae: 766.257
8
Epoch 47: loss improved from 1100991.37500 to 1091543.12500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 18s 7ms/step - loss: 1091543.1250 - mae: 76
6.4042 - lr: 0.0010
Epoch 48/100
2548/2552 [============================>.] - ETA: 0s - loss: 1098803.8750 - mae: 768.034
7
Epoch 48: loss did not improve from 1091543.12500
2552/2552 [==============================] - 18s 7ms/step - loss: 1098594.1250 - mae: 76
8.0084 - lr: 0.0010
Epoch 49/100
2545/2552 [============================>.] - ETA: 0s - loss: 1093649.0000 - mae: 764.620
3
Epoch 49: loss did not improve from 1091543.12500
2552/2552 [==============================] - 15s 6ms/step - loss: 1093252.2500 - mae: 76
4.5656 - lr: 0.0010
Epoch 50/100
2545/2552 [============================>.] - ETA: 0s - loss: 1083735.0000 - mae: 760.657
3
Epoch 50: loss improved from 1091543.12500 to 1084221.25000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 14s 6ms/step - loss: 1084221.2500 - mae: 76
0.6738 - lr: 0.0010
Epoch 51/100
2546/2552 [============================>.] - ETA: 0s - loss: 1078371.1250 - mae: 757.936
0
Epoch 51: loss improved from 1084221.25000 to 1078238.62500, saving model to Sales_ann.h
5
```

```
2552/2552 [==============================] - 15s 6ms/step - loss: 1078238.6250 - mae: 75
7.9117 - lr: 0.0010
Epoch 52/100
2550/2552 [============================>.] - ETA: 0s - loss: 1068968.1250 - mae: 755.056
0
Epoch 52: loss improved from 1078238.62500 to 1068678.50000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 15s 6ms/step - loss: 1068678.5000 - mae: 75
4.9467 - lr: 0.0010
Epoch 53/100
2552/2552 [==============================] - ETA: 0s - loss: 1062827.2500 - mae: 751.395
3
Epoch 53: loss improved from 1068678.50000 to 1062827.25000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 14s 6ms/step - loss: 1062827.2500 - mae: 75
1.3953 - lr: 0.0010
Epoch 54/100
2549/2552 [============================>.] - ETA: 0s - loss: 1060569.2500 - mae: 752.433
2
Epoch 54: loss improved from 1062827.25000 to 1060949.12500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 14s 6ms/step - loss: 1060949.1250 - mae: 75
2.5190 - lr: 0.0010
Epoch 55/100
2551/2552 [============================>.] - ETA: 0s - loss: 1045806.1875 - mae: 743.710
6
Epoch 55: loss improved from 1060949.12500 to 1045755.31250, saving model to Sales_ann.h
5
2552/2552 [==============================] - 14s 6ms/step - loss: 1045755.3125 - mae: 74
3.6920 - lr: 0.0010
Epoch 56/100
2545/2552 [============================>.] - ETA: 0s - loss: 1044831.3125 - mae: 743.527
5
Epoch 56: loss improved from 1045755.31250 to 1045446.81250, saving model to Sales_ann.h
5
2552/2552 [==============================] - 14s 6ms/step - loss: 1045446.8125 - mae: 74
3.6687 - lr: 0.0010
Epoch 57/100
2549/2552 [============================>.] - ETA: 0s - loss: 1035735.4375 - mae: 741.978
0
Epoch 57: loss improved from 1045446.81250 to 1035556.87500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 14s 5ms/step - loss: 1035556.8750 - mae: 74
1.9596 - lr: 0.0010
Epoch 58/100
2547/2552 [============================>.] - ETA: 0s - loss: 1025297.8125 - mae: 736.340
5
Epoch 58: loss improved from 1035556.87500 to 1026385.18750, saving model to Sales_ann.h
5
2552/2552 [==============================] - 15s 6ms/step - loss: 1026385.1875 - mae: 73
6.7125 - lr: 0.0010
Epoch 59/100
2551/2552 [============================>.] - ETA: 0s - loss: 1029848.9375 - mae: 737.299
0
Epoch 59: loss did not improve from 1026385.18750
2552/2552 [==============================] - 15s 6ms/step - loss: 1029799.6875 - mae: 73
7.2744 - lr: 0.0010
Epoch 60/100
2544/2552 [============================>.] - ETA: 0s - loss: 1018178.7500 - mae: 732.347
1
```

```
Epoch 60: loss improved from 1026385.18750 to 1018250.12500, saving model to Sales_ann.h
5
2552/2552 [==============================] - 15s 6ms/step - loss: 1018250.1250 - mae: 73
2.4562 - lr: 0.0010
Epoch 61/100
2550/2552 [============================>.] - ETA: 0s - loss: 1023263.3750 - mae: 733.430
3
Epoch 61: loss did not improve from 1018250.12500
2552/2552 [==============================] - 15s 6ms/step - loss: 1023128.3750 - mae: 73
3.4089 - lr: 0.0010
Epoch 62/100
2544/2552 [============================>.] - ETA: 0s - loss: 1012765.6250 - mae: 731.762
5
Epoch 62: loss improved from 1018250.12500 to 1013146.00000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 15s 6ms/step - loss: 1013146.0000 - mae: 73
1.8661 - lr: 0.0010
Epoch 63/100
2551/2552 [============================>.] - ETA: 0s - loss: 1003854.7500 - mae: 727.351
6
Epoch 63: loss improved from 1013146.00000 to 1003790.50000, saving model to Sales_ann.h
5
2552/2552 [==============================] - 20s 8ms/step - loss: 1003790.5000 - mae: 72
7.3329 - lr: 0.0010
Epoch 64/100
2547/2552 [============================>.] - ETA: 0s - loss: 993473.2500 - mae: 724.0655
Epoch 64: loss improved from 1003790.50000 to 993414.31250, saving model to Sales_ann.h5
2552/2552 [==============================] - 20s 8ms/step - loss: 993414.3125 - mae: 72
4.0286 - lr: 0.0010
Epoch 65/100
2546/2552 [============================>.] - ETA: 0s - loss: 987247.4375 - mae: 721.2065
Epoch 65: loss improved from 993414.31250 to 987162.43750, saving model to Sales_ann.h5
2552/2552 [==============================] - 17s 7ms/step - loss: 987162.4375 - mae: 72
1.2048 - lr: 0.0010
Epoch 66/100
2549/2552 [============================>.] - ETA: 0s - loss: 986244.6250 - mae: 719.9149
Epoch 66: loss improved from 987162.43750 to 986433.06250, saving model to Sales_ann.h5
2552/2552 [==============================] - 17s 7ms/step - loss: 986433.0625 - mae: 71
9.9966 - lr: 0.0010
Epoch 67/100
2547/2552 [============================>.] - ETA: 0s - loss: 983324.8750 - mae: 717.4817
Epoch 67: loss improved from 986433.06250 to 983116.75000, saving model to Sales_ann.h5
2552/2552 [==============================] - 14s 6ms/step - loss: 983116.7500 - mae: 71
7.5283 - lr: 0.0010
Epoch 68/100
2546/2552 [============================>.] - ETA: 0s - loss: 972118.9375 - mae: 713.0747
Epoch 68: loss improved from 983116.75000 to 972133.68750, saving model to Sales_ann.h5
2552/2552 [==============================] - 15s 6ms/step - loss: 972133.6875 - mae: 71
3.0720 - lr: 0.0010
Epoch 69/100
2551/2552 [============================>.] - ETA: 0s - loss: 960039.7500 - mae: 711.3426
Epoch 69: loss improved from 972133.68750 to 960254.93750, saving model to Sales_ann.h5
2552/2552 [==============================] - 15s 6ms/step - loss: 960254.9375 - mae: 71
1.4229 - lr: 0.0010
Epoch 70/100
2550/2552 [============================>.] - ETA: 0s - loss: 960716.0000 - mae: 709.1064
Epoch 70: loss did not improve from 960254.93750
2552/2552 [==============================] - 15s 6ms/step - loss: 961308.3125 - mae: 70
9.2504 - lr: 0.0010
Epoch 71/100
```

```
2548/2552 [============================>.] - ETA: 0s - loss: 957594.3125 - mae: 710.8342
Epoch 71: loss improved from 960254.93750 to 957091.00000, saving model to Sales_ann.h5
2552/2552 [==============================] - 17s 7ms/step - loss: 957091.0000 - mae: 71
0.6564 - lr: 0.0010
Epoch 72/100
2545/2552 [============================>.] - ETA: 0s - loss: 948920.7500 - mae: 703.4914
Epoch 72: loss improved from 957091.00000 to 949227.87500, saving model to Sales_ann.h5
2552/2552 [==============================] - 18s 7ms/step - loss: 949227.8750 - mae: 70
3.5972 - lr: 0.0010
Epoch 73/100
2544/2552 [============================>.] - ETA: 0s - loss: 940186.0000 - mae: 701.1802
Epoch 73: loss improved from 949227.87500 to 939247.18750, saving model to Sales_ann.h5
2552/2552 [==============================] - 17s 7ms/step - loss: 939247.1875 - mae: 70
0.9122 - lr: 0.0010
Epoch 74/100
2546/2552 [============================>.] - ETA: 0s - loss: 930231.3125 - mae: 697.9703
Epoch 74: loss improved from 939247.18750 to 930318.81250, saving model to Sales_ann.h5
2552/2552 [==============================] - 15s 6ms/step - loss: 930318.8125 - mae: 69
8.1081 - lr: 0.0010
Epoch 75/100
2546/2552 [============================>.] - ETA: 0s - loss: 925826.3750 - mae: 695.1223
Epoch 75: loss improved from 930318.81250 to 925813.18750, saving model to Sales_ann.h5
2552/2552 [==============================] - 15s 6ms/step - loss: 925813.1875 - mae: 69
5.1445 - lr: 0.0010
Epoch 76/100
2548/2552 [============================>.] - ETA: 0s - loss: 914574.2500 - mae: 690.9155
Epoch 76: loss improved from 925813.18750 to 915041.87500, saving model to Sales_ann.h5
2552/2552 [==============================] - 15s 6ms/step - loss: 915041.8750 - mae: 69
1.0904 - lr: 0.0010
Epoch 77/100
2549/2552 [============================>.] - ETA: 0s - loss: 910417.0000 - mae: 688.5111
Epoch 77: loss improved from 915041.87500 to 910256.12500, saving model to Sales_ann.h5
2552/2552 [==============================] - 16s 6ms/step - loss: 910256.1250 - mae: 68
8.4750 - lr: 0.0010
Epoch 78/100
2549/2552 [============================>.] - ETA: 0s - loss: 903365.1875 - mae: 686.3673
Epoch 78: loss improved from 910256.12500 to 903389.18750, saving model to Sales_ann.h5
2552/2552 [==============================] - 20s 8ms/step - loss: 903389.1875 - mae: 68
6.3627 - lr: 0.0010
Epoch 79/100
2550/2552 [============================>.] - ETA: 0s - loss: 892637.9375 - mae: 682.0482
Epoch 79: loss improved from 903389.18750 to 892919.00000, saving model to Sales_ann.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 892919.0000 - mae: 68
2.1385 - lr: 0.0010
Epoch 80/100
2545/2552 [============================>.] - ETA: 0s - loss: 883387.4375 - mae: 677.3474
Epoch 80: loss improved from 892919.00000 to 883706.81250, saving model to Sales_ann.h5
2552/2552 [==============================] - 20s 8ms/step - loss: 883706.8125 - mae: 67
7.4731 - lr: 0.0010
Epoch 81/100
2546/2552 [============================>.] - ETA: 0s - loss: 873161.3125 - mae: 674.2670
Epoch 81: loss improved from 883706.81250 to 872908.93750, saving model to Sales_ann.h5
2552/2552 [==============================] - 18s 7ms/step - loss: 872908.9375 - mae: 67
4.2249 - lr: 0.0010
Epoch 82/100
2546/2552 [============================>.] - ETA: 0s - loss: 879101.7500 - mae: 674.7071
Epoch 82: loss did not improve from 872908.93750
2552/2552 [==============================] - 18s 7ms/step - loss: 879026.0625 - mae: 67
4.7444 - lr: 0.0010
Epoch 83/100
```

```
2549/2552 [============================>.] - ETA: 0s - loss: 865155.5625 - mae: 670.2803
Epoch 83: loss improved from 872908.93750 to 864798.62500, saving model to Sales_ann.h5
2552/2552 [==============================] - 22s 8ms/step - loss: 864798.6250 - mae: 67
0.1506 - lr: 0.0010
Epoch 84/100
2551/2552 [============================>.] - ETA: 0s - loss: 851648.3750 - mae: 665.4742
Epoch 84: loss improved from 864798.62500 to 851730.25000, saving model to Sales_ann.h5
2552/2552 [==============================] - 17s 7ms/step - loss: 851730.2500 - mae: 66
5.5047 - lr: 0.0010
Epoch 85/100
2545/2552 [============================>.] - ETA: 0s - loss: 844601.3750 - mae: 665.0186
Epoch 85: loss improved from 851730.25000 to 844183.93750, saving model to Sales_ann.h5
2552/2552 [==============================] - 17s 7ms/step - loss: 844183.9375 - mae: 66
4.8978 - lr: 0.0010
Epoch 86/100
2548/2552 [============================>.] - ETA: 0s - loss: 833593.5625 - mae: 659.4109
Epoch 86: loss improved from 844183.93750 to 833389.68750, saving model to Sales_ann.h5
2552/2552 [==============================] - 18s 7ms/step - loss: 833389.6875 - mae: 65
9.4217 - lr: 0.0010
Epoch 87/100
2550/2552 [============================>.] - ETA: 0s - loss: 828047.9375 - mae: 655.9324
Epoch 87: loss improved from 833389.68750 to 827912.31250, saving model to Sales_ann.h5
2552/2552 [==============================] - 16s 6ms/step - loss: 827912.3125 - mae: 65
5.8937 - lr: 0.0010
Epoch 88/100
2544/2552 [============================>.] - ETA: 0s - loss: 809359.3125 - mae: 650.7542
Epoch 88: loss improved from 827912.31250 to 809662.62500, saving model to Sales_ann.h5
2552/2552 [==============================] - 18s 7ms/step - loss: 809662.6250 - mae: 65
0.9487 - lr: 0.0010
Epoch 89/100
2551/2552 [============================>.] - ETA: 0s - loss: 807751.2500 - mae: 647.5331
Epoch 89: loss improved from 809662.62500 to 807842.12500, saving model to Sales_ann.h5
2552/2552 [==============================] - 17s 7ms/step - loss: 807842.1250 - mae: 64
7.5612 - lr: 0.0010
Epoch 90/100
2546/2552 [============================>.] - ETA: 0s - loss: 812491.3125 - mae: 649.1507
Epoch 90: loss did not improve from 807842.12500
2552/2552 [==============================] - 17s 7ms/step - loss: 811963.3750 - mae: 64
8.9825 - lr: 0.0010
Epoch 91/100
2547/2552 [============================>.] - ETA: 0s - loss: 790529.3750 - mae: 642.3789
Epoch 91: loss improved from 807842.12500 to 790625.56250, saving model to Sales_ann.h5
2552/2552 [==============================] - 16s 6ms/step - loss: 790625.5625 - mae: 64
2.5236 - lr: 0.0010
Epoch 92/100
2551/2552 [============================>.] - ETA: 0s - loss: 790574.0000 - mae: 641.5633
Epoch 92: loss did not improve from 790625.56250
2552/2552 [==============================] - 17s 7ms/step - loss: 790813.9375 - mae: 64
1.6472 - lr: 0.0010
Epoch 93/100
2549/2552 [============================>.] - ETA: 0s - loss: 783896.3750 - mae: 638.7541
Epoch 93: loss improved from 790625.56250 to 783670.87500, saving model to Sales_ann.h5
2552/2552 [==============================] - 19s 7ms/step - loss: 783670.8750 - mae: 63
8.7047 - lr: 0.0010
Epoch 94/100
2551/2552 [============================>.] - ETA: 0s - loss: 778631.9375 - mae: 636.1077
Epoch 94: loss improved from 783670.87500 to 778694.18750, saving model to Sales_ann.h5
2552/2552 [==============================] - 16s 6ms/step - loss: 778694.1875 - mae: 63
6.1363 - lr: 0.0010
Epoch 95/100
```

```
2551/2552 [============================>.] - ETA: 0s - loss: 769553.8125 - mae: 632.3608
Epoch 95: loss improved from 778694.18750 to 769910.75000, saving model to Sales_ann.h5
2552/2552 [==============================] - 15s 6ms/step - loss: 769910.7500 - mae: 63
2.4765 - lr: 0.0010
Epoch 96/100
2552/2552 [==============================] - ETA: 0s - loss: 768505.1250 - mae: 633.6362
Epoch 96: loss improved from 769910.75000 to 768505.12500, saving model to Sales_ann.h5
2552/2552 [==============================] - 17s 7ms/step - loss: 768505.1250 - mae: 63
3.6362 - lr: 0.0010
Epoch 97/100
2549/2552 [============================>.] - ETA: 0s - loss: 753393.8750 - mae: 626.3856
Epoch 97: loss improved from 768505.12500 to 753251.37500, saving model to Sales_ann.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 753251.3750 - mae: 62
6.3411 - lr: 0.0010
Epoch 98/100
2550/2552 [============================>.] - ETA: 0s - loss: 758932.4375 - mae: 627.2510
Epoch 98: loss did not improve from 753251.37500
2552/2552 [==============================] - 19s 7ms/step - loss: 759562.9375 - mae: 62
7.3859 - lr: 0.0010
Epoch 99/100
2548/2552 [============================>.] - ETA: 0s - loss: 752639.9375 - mae: 624.8098
Epoch 99: loss improved from 753251.37500 to 752461.56250, saving model to Sales_ann.h5
2552/2552 [==============================] - 19s 7ms/step - loss: 752461.5625 - mae: 62
4.7696 - lr: 0.0010
Epoch 100/100
2546/2552 [============================>.] - ETA: 0s - loss: 757746.3750 - mae: 628.7435
Epoch 100: loss did not improve from 752461.56250
2552/2552 [==============================] - 17s 7ms/step - loss: 757139.8125 - mae: 62
8.5262 - lr: 0.0010
```

In [163…

```
model_1 = load_model('Sales_ann.h5')
y_pred = model_1.predict(x_test)
```

In [164…

```
RMSE = math.sqrt(mean_squared_error(y_test,y_pred))
RMSE
```

Out[164…

```
1575.3432393623336
```

In [172…

```
#importing testing data
test_data = pd.read_csv(r'test_data_hidden.csv')
test_data.head()
```

Out[172…

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	5	2015-07-31	5263	555	1	1	0	1
1	2	5	2015-07-31	6064	625	1	1	0	1
2	3	5	2015-07-31	8314	821	1	1	0	1
3	4	5	2015-07-31	13995	1498	1	1	0	1
4	5	5	2015-07-31	4822	559	1	1	0	1

In [173…

```
test_data.drop('Date',axis = 1, inplace=True)
test_data.loc[test_data.StateHoliday==0,'StateHoliday'] = '0'
```

```
labelencoder= LabelEncoder()
test_data.StateHoliday = labelencoder.fit_transform(test_data['StateHoliday'])
test_data = test_data[test_data.Store<=100]
test_data = test_data[test_data.Open == 1]
test_data.reset_index(drop=True, inplace=True)
y = test_data['Sales']
x = test_data.drop(['Sales','Open'],axis=1)
std = StandardScaler()
x = std.fit_transform(x)
```

In [174…

```
y_pred = model_1.predict(x)
math.sqrt(mean_squared_error(y,y_pred))
```

Out[174…

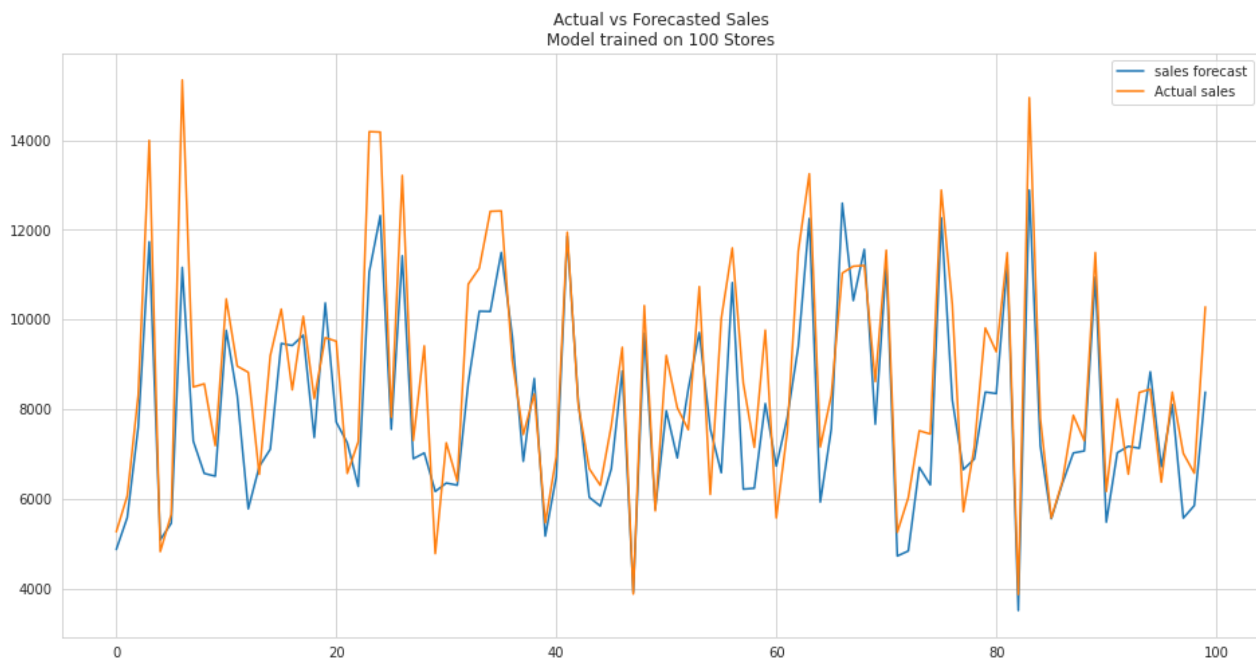
```
883.9005694611457
```

In [175…

```
plt.figure(figsize=(16,8))
plt.plot(y_pred[:100],label = 'sales forecast')
plt.plot(y[:100],label = 'Actual sales')
plt.legend()
plt.title('Actual vs Forecasted Sales\nModel trained on 100 Stores')
```

Out[175…

```
Text(0.5, 1.0, 'Actual vs Forecasted Sales\nModel trained on 100 Stores')
```



In [176…

```
# Comparing Model performance with the traditional ML based Prediction Models
```

2. Use Dropout for ANN and find the optimum number of clusters (clusters formed considering the features: sales and customer visits). Compare model performance with traditional ML based prediction models

In [177…

```
train_data = train_data[train_data.Store<=100]
train_data = train_data[train_data.Open == 1]
train_data.reset_index(drop=True, inplace=True)
y = train_data['Sales']
x = train_data.drop(['Sales','Open'],axis=1)
std = StandardScaler()
x = std.fit_transform(x)
x_train,x_test,y_train,y_test = train_test_split(np.array(x),np.array(y),random_state=4
```

In [178…

```
model_2 = Sequential()
model_2.add(layers.Dense(32, activation='elu', input_shape = (x_train.shape[1],)))
model_2.add(layers.Dense(64, activation='elu'))
model_2.add(layers.Dense(64, activation='elu'))
model_2.add(layers.BatchNormalization())
## block 2
model_2.add(layers.Dense(128, activation='elu'))
model_2.add(layers.Dense(128, activation='elu'))
model_2.add(layers.BatchNormalization())
## block 3
model_2.add(layers.Dense(256, activation='elu'))
model_2.add(layers.Dense(256, activation='elu'))
model_2.add(layers.BatchNormalization())
model_2.add(layers.Dropout(0.8))
## block 4
model_2.add(layers.Dense(128, activation='elu'))
model_2.add(layers.Dense(128, activation='elu'))
model_2.add(layers.BatchNormalization())
model_2.add(layers.Dropout(0.8))
## block 5
model_2.add(layers.Dense(64, activation='elu'))
model_2.add(layers.Dense(64, activation='elu'))
model_2.add(layers.Dense(32, activation='elu'))
model_2.add(layers.Dropout(0.8))
model_2.add(layers.Dense(1))
```

In [179…

```
model_2.compile(loss='mse',
optimizer = Adam(learning_rate=0.001),
metrics=['mae'])
```

In [180…

```
checkpoint = ModelCheckpoint('Sales_ann_with_dropout.h5',
monitor='loss',
mode=min,
save_best_only=True,
verbose=1)
early_stopping = EarlyStopping(monitor='loss',
patience=9,
min_delta=0,
                               restore_best_weights=True,
verbose=1)
Reduce_ler_rate = ReduceLROnPlateau(monitor='loss',
factor=0.2,
patience=3,
verbose=1,
```

```
min_delta=0.001)
callback = [checkpoint,early_stopping,Reduce_ler_rate]
```

```
WARNING:tensorflow:ModelCheckpoint mode <built-in function min> is unknown, fallback to
auto mode.
```

In [181…

```
history = model_2.fit(x_train,y_train,epochs=50,batch_size=20,verbose=1,callbacks=callb
```

```
Epoch 1/50
2551/2552 [============================>.] - ETA: 0s - loss: 16636906.0000 - mae: 3259.1
450
Epoch 1: loss improved from inf to 16638317.00000, saving model to Sales_ann_with_dropou
t.h5
2552/2552 [==============================] - 20s 7ms/step - loss: 16638317.0000 - mae: 3
259.3525 - lr: 0.0010
Epoch 2/50
2547/2552 [============================>.] - ETA: 0s - loss: 13201642.0000 - mae: 2845.4
307
Epoch 2: loss improved from 16638317.00000 to 13193533.00000, saving model to Sales_ann_
with_dropout.h5
2552/2552 [==============================] - 20s 8ms/step - loss: 13193533.0000 - mae: 2
844.3430 - lr: 0.0010
Epoch 3/50
2551/2552 [============================>.] - ETA: 0s - loss: 12593418.0000 - mae: 2745.6
492
Epoch 3: loss improved from 13193533.00000 to 12595341.00000, saving model to Sales_ann_
with_dropout.h5
2552/2552 [==============================] - 23s 9ms/step - loss: 12595341.0000 - mae: 2
745.8708 - lr: 0.0010
Epoch 4/50
2546/2552 [============================>.] - ETA: 0s - loss: 12233301.0000 - mae: 2692.9
856
Epoch 4: loss improved from 12595341.00000 to 12237170.00000, saving model to Sales_ann_
with_dropout.h5
2552/2552 [==============================] - 22s 8ms/step - loss: 12237170.0000 - mae: 2
693.2834 - lr: 0.0010
Epoch 5/50
2546/2552 [============================>.] - ETA: 0s - loss: 11948336.0000 - mae: 2662.4
446
Epoch 5: loss improved from 12237170.00000 to 11950460.00000, saving model to Sales_ann_
with_dropout.h5
2552/2552 [==============================] - 20s 8ms/step - loss: 11950460.0000 - mae: 2
662.6177 - lr: 0.0010
Epoch 6/50
2549/2552 [============================>.] - ETA: 0s - loss: 11810823.0000 - mae: 2639.2
815
Epoch 6: loss improved from 11950460.00000 to 11815900.00000, saving model to Sales_ann_
with_dropout.h5
2552/2552 [==============================] - 22s 8ms/step - loss: 11815900.0000 - mae: 2
639.8403 - lr: 0.0010
Epoch 7/50
2550/2552 [============================>.] - ETA: 0s - loss: 11639852.0000 - mae: 2616.4
976
Epoch 7: loss improved from 11815900.00000 to 11643045.00000, saving model to Sales_ann_
with_dropout.h5
2552/2552 [==============================] - 22s 9ms/step - loss: 11643045.0000 - mae: 2
616.8389 - lr: 0.0010
Epoch 8/50
2552/2552 [==============================] - ETA: 0s - loss: 11592136.0000 - mae: 2610.4
```

```
646
Epoch 8: loss improved from 11643045.00000 to 11592136.00000, saving model to Sales_ann_
with_dropout.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 11592136.0000 - mae: 2
610.4646 - lr: 0.0010
Epoch 9/50
2548/2552 [============================>.] - ETA: 0s - loss: 11430286.0000 - mae: 2589.2
578
Epoch 9: loss improved from 11592136.00000 to 11427590.00000, saving model to Sales_ann_
with_dropout.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 11427590.0000 - mae: 2
588.9824 - lr: 0.0010
Epoch 10/50
2551/2552 [============================>.] - ETA: 0s - loss: 11378049.0000 - mae: 2580.6
963
Epoch 10: loss improved from 11427590.00000 to 11377793.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 19s 7ms/step - loss: 11377793.0000 - mae: 2
580.7158 - lr: 0.0010
Epoch 11/50
2549/2552 [============================>.] - ETA: 0s - loss: 11253710.0000 - mae: 2575.6
316
Epoch 11: loss improved from 11377793.00000 to 11250971.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 22s 8ms/step - loss: 11250971.0000 - mae: 2
575.4297 - lr: 0.0010
Epoch 12/50
2552/2552 [==============================] - ETA: 0s - loss: 11414456.0000 - mae: 2583.1
377
Epoch 12: loss did not improve from 11250971.00000
2552/2552 [==============================] - 22s 9ms/step - loss: 11414456.0000 - mae: 2
583.1377 - lr: 0.0010
Epoch 13/50
2550/2552 [============================>.] - ETA: 0s - loss: 11309842.0000 - mae: 2570.1
082
Epoch 13: loss did not improve from 11250971.00000
2552/2552 [==============================] - 19s 8ms/step - loss: 11307651.0000 - mae: 2
569.9495 - lr: 0.0010
Epoch 14/50
2548/2552 [============================>.] - ETA: 0s - loss: 11112370.0000 - mae: 2557.7
627
Epoch 14: loss improved from 11250971.00000 to 11110988.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 11110988.0000 - mae: 2
557.6262 - lr: 0.0010
Epoch 15/50
2548/2552 [============================>.] - ETA: 0s - loss: 10889012.0000 - mae: 2527.1
897
Epoch 15: loss improved from 11110988.00000 to 10890640.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 10890640.0000 - mae: 2
527.4873 - lr: 0.0010
Epoch 16/50
2547/2552 [============================>.] - ETA: 0s - loss: 11004190.0000 - mae: 2539.9
153
Epoch 16: loss did not improve from 10890640.00000
2552/2552 [==============================] - 20s 8ms/step - loss: 11004246.0000 - mae: 2
539.8979 - lr: 0.0010
Epoch 17/50
2550/2552 [============================>.] - ETA: 0s - loss: 10832693.0000 - mae: 2520.6
```

```
948
Epoch 17: loss improved from 10890640.00000 to 10833263.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 23s 9ms/step - loss: 10833263.0000 - mae: 2
520.8528 - lr: 0.0010
Epoch 18/50
2550/2552 [============================>.] - ETA: 0s - loss: 10812939.0000 - mae: 2510.2
781
Epoch 18: loss improved from 10833263.00000 to 10814722.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 22s 8ms/step - loss: 10814722.0000 - mae: 2
510.5498 - lr: 0.0010
Epoch 19/50
2552/2552 [==============================] - ETA: 0s - loss: 10622481.0000 - mae: 2500.7
559
Epoch 19: loss improved from 10814722.00000 to 10622481.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 22s 9ms/step - loss: 10622481.0000 - mae: 2
500.7559 - lr: 0.0010
Epoch 20/50
2550/2552 [============================>.] - ETA: 0s - loss: 10597532.0000 - mae: 2491.3
464
Epoch 20: loss improved from 10622481.00000 to 10597510.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 20s 8ms/step - loss: 10597510.0000 - mae: 2
491.2749 - lr: 0.0010
Epoch 21/50
2551/2552 [============================>.] - ETA: 0s - loss: 10634724.0000 - mae: 2483.9
951
Epoch 21: loss did not improve from 10597510.00000
2552/2552 [==============================] - 21s 8ms/step - loss: 10634445.0000 - mae: 2
483.9614 - lr: 0.0010
Epoch 22/50
2550/2552 [============================>.] - ETA: 0s - loss: 10525107.0000 - mae: 2483.5
417
Epoch 22: loss improved from 10597510.00000 to 10523759.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 22s 9ms/step - loss: 10523759.0000 - mae: 2
483.5627 - lr: 0.0010
Epoch 23/50
2551/2552 [============================>.] - ETA: 0s - loss: 10588346.0000 - mae: 2480.4
583
Epoch 23: loss did not improve from 10523759.00000
2552/2552 [==============================] - 23s 9ms/step - loss: 10588954.0000 - mae: 2
480.5632 - lr: 0.0010
Epoch 24/50
2547/2552 [============================>.] - ETA: 0s - loss: 10393479.0000 - mae: 2467.8
367
Epoch 24: loss improved from 10523759.00000 to 10392598.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 23s 9ms/step - loss: 10392598.0000 - mae: 2
467.9038 - lr: 0.0010
Epoch 25/50
2550/2552 [============================>.] - ETA: 0s - loss: 10499137.0000 - mae: 2472.9
546
Epoch 25: loss did not improve from 10392598.00000
2552/2552 [==============================] - 22s 9ms/step - loss: 10497311.0000 - mae: 2
472.7722 - lr: 0.0010
Epoch 26/50
2546/2552 [============================>.] - ETA: 0s - loss: 10352733.0000 - mae: 2460.2
```

```
725
Epoch 26: loss improved from 10392598.00000 to 10348185.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 24s 9ms/step - loss: 10348185.0000 - mae: 2
459.8691 - lr: 0.0010
Epoch 27/50
2549/2552 [============================>.] - ETA: 0s - loss: 10293696.0000 - mae: 2450.7
327
Epoch 27: loss improved from 10348185.00000 to 10290537.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 10290537.0000 - mae: 2
450.3838 - lr: 0.0010
Epoch 28/50
2551/2552 [============================>.] - ETA: 0s - loss: 10226504.0000 - mae: 2439.2
581
Epoch 28: loss improved from 10290537.00000 to 10227725.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 22s 9ms/step - loss: 10227725.0000 - mae: 2
439.4219 - lr: 0.0010
Epoch 29/50
2549/2552 [============================>.] - ETA: 0s - loss: 10023979.0000 - mae: 2416.3
472
Epoch 29: loss improved from 10227725.00000 to 10026678.00000, saving model to Sales_ann
_with_dropout.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 10026678.0000 - mae: 2
416.6936 - lr: 0.0010
Epoch 30/50
2551/2552 [============================>.] - ETA: 0s - loss: 10078422.0000 - mae: 2428.8
008
Epoch 30: loss did not improve from 10026678.00000
2552/2552 [==============================] - 23s 9ms/step - loss: 10079991.0000 - mae: 2
428.9985 - lr: 0.0010
Epoch 31/50
2552/2552 [==============================] - ETA: 0s - loss: 10053265.0000 - mae: 2427.4
583
Epoch 31: loss did not improve from 10026678.00000
2552/2552 [==============================] - 24s 9ms/step - loss: 10053265.0000 - mae: 2
427.4583 - lr: 0.0010
Epoch 32/50
2549/2552 [============================>.] - ETA: 0s - loss: 9948639.0000 - mae: 2405.99
76
Epoch 32: loss improved from 10026678.00000 to 9948221.00000, saving model to Sales_ann_
with_dropout.h5
2552/2552 [==============================] - 22s 9ms/step - loss: 9948221.0000 - mae: 24
06.1155 - lr: 0.0010
Epoch 33/50
2548/2552 [============================>.] - ETA: 0s - loss: 9908794.0000 - mae: 2402.10
45
Epoch 33: loss improved from 9948221.00000 to 9911204.00000, saving model to Sales_ann_w
ith_dropout.h5
2552/2552 [==============================] - 22s 8ms/step - loss: 9911204.0000 - mae: 24
02.3933 - lr: 0.0010
Epoch 34/50
2548/2552 [============================>.] - ETA: 0s - loss: 9746177.0000 - mae: 2379.18
75
Epoch 34: loss improved from 9911204.00000 to 9742999.00000, saving model to Sales_ann_w
ith_dropout.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 9742999.0000 - mae: 23
78.8235 - lr: 0.0010
Epoch 35/50
```

```
2548/2552 [============================>.] - ETA: 0s - loss: 9869725.0000 - mae: 2390.86
43
Epoch 35: loss did not improve from 9742999.00000
2552/2552 [==============================] - 21s 8ms/step - loss: 9873001.0000 - mae: 23
91.1746 - lr: 0.0010
Epoch 36/50
2546/2552 [============================>.] - ETA: 0s - loss: 9749817.0000 - mae: 2377.55
96
Epoch 36: loss did not improve from 9742999.00000
2552/2552 [==============================] - 22s 9ms/step - loss: 9754523.0000 - mae: 23
77.9165 - lr: 0.0010
Epoch 37/50
2548/2552 [============================>.] - ETA: 0s - loss: 9629257.0000 - mae: 2372.20
51
Epoch 37: loss improved from 9742999.00000 to 9633756.00000, saving model to Sales_ann_w
ith_dropout.h5
2552/2552 [==============================] - 24s 9ms/step - loss: 9633756.0000 - mae: 23
72.7249 - lr: 0.0010
Epoch 38/50
2546/2552 [============================>.] - ETA: 0s - loss: 9833603.0000 - mae: 2379.45
70
Epoch 38: loss did not improve from 9633756.00000
2552/2552 [==============================] - 22s 9ms/step - loss: 9831975.0000 - mae: 23
79.2712 - lr: 0.0010
Epoch 39/50
2552/2552 [==============================] - ETA: 0s - loss: 9696165.0000 - mae: 2371.53
08
Epoch 39: loss did not improve from 9633756.00000
2552/2552 [==============================] - 22s 9ms/step - loss: 9696165.0000 - mae: 23
71.5308 - lr: 0.0010
Epoch 40/50
2550/2552 [============================>.] - ETA: 0s - loss: 9605093.0000 - mae: 2363.63
96
Epoch 40: loss improved from 9633756.00000 to 9603482.00000, saving model to Sales_ann_w
ith_dropout.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 9603482.0000 - mae: 23
63.3677 - lr: 0.0010
Epoch 41/50
2547/2552 [============================>.] - ETA: 0s - loss: 9488382.0000 - mae: 2345.25
39
Epoch 41: loss improved from 9603482.00000 to 9483205.00000, saving model to Sales_ann_w
ith_dropout.h5
2552/2552 [==============================] - 22s 9ms/step - loss: 9483205.0000 - mae: 23
44.7612 - lr: 0.0010
Epoch 42/50
2552/2552 [==============================] - ETA: 0s - loss: 9540081.0000 - mae: 2350.20
97
Epoch 42: loss did not improve from 9483205.00000
2552/2552 [==============================] - 22s 9ms/step - loss: 9540081.0000 - mae: 23
50.2097 - lr: 0.0010
Epoch 43/50
2551/2552 [============================>.] - ETA: 0s - loss: 9417323.0000 - mae: 2333.86
04
Epoch 43: loss improved from 9483205.00000 to 9417306.00000, saving model to Sales_ann_w
ith_dropout.h5
2552/2552 [==============================] - 21s 8ms/step - loss: 9417306.0000 - mae: 23
33.8855 - lr: 0.0010
Epoch 44/50
2546/2552 [============================>.] - ETA: 0s - loss: 9448074.0000 - mae: 2344.24
19
```

```
Epoch 44: loss did not improve from 9417306.00000
2552/2552 [==============================] - 19s 8ms/step - loss: 9449936.0000 - mae: 23
44.3813 - lr: 0.0010
Epoch 45/50
2551/2552 [============================>.] - ETA: 0s - loss: 9387439.0000 - mae: 2333.98
80
Epoch 45: loss improved from 9417306.00000 to 9387210.00000, saving model to Sales_ann_w
ith_dropout.h5
2552/2552 [==============================] - 19s 8ms/step - loss: 9387210.0000 - mae: 23
33.9641 - lr: 0.0010
Epoch 46/50
2546/2552 [============================>.] - ETA: 0s - loss: 9309885.0000 - mae: 2323.32
76
Epoch 46: loss improved from 9387210.00000 to 9308550.00000, saving model to Sales_ann_w
ith_dropout.h5
2552/2552 [==============================] - 19s 8ms/step - loss: 9308550.0000 - mae: 23
23.1963 - lr: 0.0010
Epoch 47/50
2547/2552 [============================>.] - ETA: 0s - loss: 9232301.0000 - mae: 2318.83
98
Epoch 47: loss improved from 9308550.00000 to 9230709.00000, saving model to Sales_ann_w
ith_dropout.h5
2552/2552 [==============================] - 20s 8ms/step - loss: 9230709.0000 - mae: 23
18.6418 - lr: 0.0010
Epoch 48/50
2548/2552 [============================>.] - ETA: 0s - loss: 9324986.0000 - mae: 2317.01
54
Epoch 48: loss did not improve from 9230709.00000
2552/2552 [==============================] - 21s 8ms/step - loss: 9324427.0000 - mae: 23
16.9446 - lr: 0.0010
Epoch 49/50
2547/2552 [============================>.] - ETA: 0s - loss: 9294215.0000 - mae: 2315.01
07
Epoch 49: loss did not improve from 9230709.00000
2552/2552 [==============================] - 20s 8ms/step - loss: 9293928.0000 - mae: 23
14.8064 - lr: 0.0010
Epoch 50/50
2550/2552 [============================>.] - ETA: 0s - loss: 9272234.0000 - mae: 2305.68
14
Epoch 50: loss did not improve from 9230709.00000

Epoch 50: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
2552/2552 [==============================] - 20s 8ms/step - loss: 9274892.0000 - mae: 23
05.8740 - lr: 0.0010
```

In [182…

```python
model_2 = load_model('Sales_ann_with_dropout.h5')
y_pred = model_2.predict(x_test)
math.sqrt(mean_squared_error(y_test,y_pred))
```

Out[182…

```
1885.6955015453786
```

In [183…

```python
# This does not seem to be a good model . The RMSE increased
```

3. Find the best setting of neural net that minimizes the loss and can predict the sales

best. Use techniques like Grid search, cross-validation and Random search.

In [184…

```
# Lets do cross validation
```

In [186…

```
def modelkf(x_train,y_train,x_test,y_test):
    model = Sequential()
    model.add(layers.Dense(32, activation='elu', input_shape = (x_train.shape[1],)))
    model.add(layers.Dense(64, activation='elu'))
    model.add(layers.Dense(64, activation='elu'))
    model.add(layers.BatchNormalization())
## block 2
    model.add(layers.Dense(128, activation='elu'))
    model.add(layers.Dense(128, activation='elu'))
    model.add(layers.BatchNormalization())
## block 3
    model.add(layers.Dense(256, activation='elu'))
    model.add(layers.Dense(256, activation='elu'))
    model.add(layers.BatchNormalization())
    model.add(layers.Dropout(0.8))
## block 4
    model.add(layers.Dense(128, activation='elu'))
    model.add(layers.Dense(128, activation='elu'))
    model.add(layers.BatchNormalization())
    model.add(layers.Dropout(0.8))
## block 5
    model.add(layers.Dense(64, activation='elu'))
    model.add(layers.Dense(64, activation='elu'))
    model.add(layers.Dense(32, activation='elu'))
    model.add(layers.Dropout(0.8))
    model.add(layers.Dense(1))
    model.compile(loss='mse',
    optimizer = Adam(learning_rate=0.001),
    metrics=['mae'])
    model.fit(x_train,y_train,epochs=50,batch_size=20)
    y_pred = model.predict(x_test)
    return (math.sqrt(mean_squared_error(y_test,y_pred)))
```

In [189…

```
score_kf_ann = []
kf = StratifiedKFold(n_splits=4)
for train_index,test_index in kf.split(x,y):
    x_train,x_test,y_train,y_test = train_test_split(np.array(x),np.array(y),test_size
    score_kf_ann.append(modelkf(x_train,y_train,x_test,y_test))
```

```
Epoch 1/50
2552/2552 [==============================] - 20s 7ms/step - loss: 17201008.0000 - mae: 3
343.0298
Epoch 2/50
2552/2552 [==============================] - 19s 7ms/step - loss: 13831685.0000 - mae: 2
935.4629
Epoch 3/50
2552/2552 [==============================] - 21s 8ms/step - loss: 13054048.0000 - mae: 2
796.3650
Epoch 4/50
2552/2552 [==============================] - 20s 8ms/step - loss: 12569152.0000 - mae: 2
```

```
733.9648
Epoch 5/50
2552/2552 [==============================] - 18s 7ms/step - loss: 12216441.0000 - mae: 2
694.4900
Epoch 6/50
2552/2552 [==============================] - 17s 7ms/step - loss: 12109310.0000 - mae: 2
676.4321
Epoch 7/50
2552/2552 [==============================] - 16s 6ms/step - loss: 12106777.0000 - mae: 2
675.8025
Epoch 8/50
2552/2552 [==============================] - 15s 6ms/step - loss: 12075968.0000 - mae: 2
663.9985
Epoch 9/50
2552/2552 [==============================] - 17s 7ms/step - loss: 11779025.0000 - mae: 2
636.1958
Epoch 10/50
2552/2552 [==============================] - 21s 8ms/step - loss: 11739596.0000 - mae: 2
623.4761
Epoch 11/50
2552/2552 [==============================] - 19s 7ms/step - loss: 11725778.0000 - mae: 2
621.0266
Epoch 12/50
2552/2552 [==============================] - 19s 8ms/step - loss: 11411105.0000 - mae: 2
587.4089
Epoch 13/50
2552/2552 [==============================] - 19s 7ms/step - loss: 11423618.0000 - mae: 2
590.0791
Epoch 14/50
2552/2552 [==============================] - 17s 7ms/step - loss: 11349615.0000 - mae: 2
581.9634
Epoch 15/50
2552/2552 [==============================] - 18s 7ms/step - loss: 11307171.0000 - mae: 2
581.0574
Epoch 16/50
2552/2552 [==============================] - 21s 8ms/step - loss: 11163926.0000 - mae: 2
557.6516
Epoch 17/50
2552/2552 [==============================] - 21s 8ms/step - loss: 11062594.0000 - mae: 2
545.3933
Epoch 18/50
2552/2552 [==============================] - 21s 8ms/step - loss: 11081266.0000 - mae: 2
543.3701
Epoch 19/50
2552/2552 [==============================] - 20s 8ms/step - loss: 11016190.0000 - mae: 2
536.9993
Epoch 20/50
2552/2552 [==============================] - 20s 8ms/step - loss: 10937019.0000 - mae: 2
526.4243
Epoch 21/50
2552/2552 [==============================] - 19s 7ms/step - loss: 10979173.0000 - mae: 2
527.5842
Epoch 22/50
2552/2552 [==============================] - 19s 7ms/step - loss: 10729971.0000 - mae: 2
500.9561
Epoch 23/50
2552/2552 [==============================] - 21s 8ms/step - loss: 10797855.0000 - mae: 2
506.0876
Epoch 24/50
2552/2552 [==============================] - 21s 8ms/step - loss: 10796919.0000 - mae: 2
```

```
497.7529
Epoch 25/50
2552/2552 [==============================] - 21s 8ms/step - loss: 10554178.0000 - mae: 2
474.3560
Epoch 26/50
2552/2552 [==============================] - 21s 8ms/step - loss: 10642533.0000 - mae: 2
474.4910
Epoch 27/50
2552/2552 [==============================] - 21s 8ms/step - loss: 10554763.0000 - mae: 2
470.2773
Epoch 28/50
2552/2552 [==============================] - 21s 8ms/step - loss: 10379608.0000 - mae: 2
448.3547
Epoch 29/50
2552/2552 [==============================] - 22s 9ms/step - loss: 10338691.0000 - mae: 2
446.9080
Epoch 30/50
2552/2552 [==============================] - 22s 9ms/step - loss: 10396266.0000 - mae: 2
453.6953
Epoch 31/50
2552/2552 [==============================] - 21s 8ms/step - loss: 10208953.0000 - mae: 2
432.2766
Epoch 32/50
2552/2552 [==============================] - 22s 9ms/step - loss: 10073136.0000 - mae: 2
412.3215
Epoch 33/50
2552/2552 [==============================] - 23s 9ms/step - loss: 10184391.0000 - mae: 2
423.9336
Epoch 34/50
2552/2552 [==============================] - 22s 9ms/step - loss: 10157595.0000 - mae: 2
418.8848
Epoch 35/50
2552/2552 [==============================] - 24s 9ms/step - loss: 9893589.0000 - mae: 23
96.8743
Epoch 36/50
2552/2552 [==============================] - 24s 10ms/step - loss: 9810459.0000 - mae: 2
376.0076
Epoch 37/50
2552/2552 [==============================] - 24s 9ms/step - loss: 10030959.0000 - mae: 2
394.7446
Epoch 38/50
2552/2552 [==============================] - 25s 10ms/step - loss: 9795052.0000 - mae: 2
381.4719
Epoch 39/50
2552/2552 [==============================] - 24s 10ms/step - loss: 9890987.0000 - mae: 2
392.2759
Epoch 40/50
2552/2552 [==============================] - 24s 9ms/step - loss: 9748795.0000 - mae: 23
72.7100
Epoch 41/50
2552/2552 [==============================] - 23s 9ms/step - loss: 9649603.0000 - mae: 23
58.1221
Epoch 42/50
2552/2552 [==============================] - 27s 11ms/step - loss: 9535824.0000 - mae: 2
347.4272
Epoch 43/50
2552/2552 [==============================] - 27s 11ms/step - loss: 9506781.0000 - mae: 2
341.7798
Epoch 44/50
2552/2552 [==============================] - 26s 10ms/step - loss: 9520836.0000 - mae: 2
```

```
333.3167
Epoch 45/50
2552/2552 [==============================] - 29s 11ms/step - loss: 9298395.0000 - mae: 2
317.5706
Epoch 46/50
2552/2552 [==============================] - 28s 11ms/step - loss: 9382559.0000 - mae: 2
316.7271
Epoch 47/50
2552/2552 [==============================] - 25s 10ms/step - loss: 9370952.0000 - mae: 2
320.5103
Epoch 48/50
2552/2552 [==============================] - 26s 10ms/step - loss: 9305314.0000 - mae: 2
306.2349
Epoch 49/50
2552/2552 [==============================] - 24s 9ms/step - loss: 9240357.0000 - mae: 23
02.5305
Epoch 50/50
2552/2552 [==============================] - 22s 9ms/step - loss: 9077255.0000 - mae: 22
80.8311
Epoch 1/50
2552/2552 [==============================] - 25s 9ms/step - loss: 17584246.0000 - mae: 3
325.1960
Epoch 2/50
2552/2552 [==============================] - 25s 10ms/step - loss: 14441851.0000 - mae:
2960.1816
Epoch 3/50
2552/2552 [==============================] - 25s 10ms/step - loss: 14107429.0000 - mae:
2923.7271
Epoch 4/50
2552/2552 [==============================] - 22s 9ms/step - loss: 13674039.0000 - mae: 2
871.3701
Epoch 5/50
2552/2552 [==============================] - 19s 8ms/step - loss: 13436677.0000 - mae: 2
834.8923
Epoch 6/50
2552/2552 [==============================] - 20s 8ms/step - loss: 13209602.0000 - mae: 2
811.8950
Epoch 7/50
2552/2552 [==============================] - 23s 9ms/step - loss: 13035734.0000 - mae: 2
788.0920
Epoch 8/50
2552/2552 [==============================] - 24s 9ms/step - loss: 12815954.0000 - mae: 2
759.6873
Epoch 9/50
2552/2552 [==============================] - 23s 9ms/step - loss: 12750811.0000 - mae: 2
746.8582
Epoch 10/50
2552/2552 [==============================] - 20s 8ms/step - loss: 12561599.0000 - mae: 2
727.6760
Epoch 11/50
2552/2552 [==============================] - 19s 8ms/step - loss: 12373930.0000 - mae: 2
714.6548
Epoch 12/50
2552/2552 [==============================] - 18s 7ms/step - loss: 12351727.0000 - mae: 2
695.0410
Epoch 13/50
2552/2552 [==============================] - 21s 8ms/step - loss: 12468809.0000 - mae: 2
701.5796
Epoch 14/50
2552/2552 [==============================] - 22s 8ms/step - loss: 12136039.0000 - mae: 2
```

```
670.4968
Epoch 15/50
2552/2552 [==============================] - 23s 9ms/step - loss: 12223526.0000 - mae: 2
678.1184
Epoch 16/50
2552/2552 [==============================] - 18s 7ms/step - loss: 11856627.0000 - mae: 2
636.7910
Epoch 17/50
2552/2552 [==============================] - 17s 7ms/step - loss: 11853088.0000 - mae: 2
637.2803
Epoch 18/50
2552/2552 [==============================] - 17s 7ms/step - loss: 11907108.0000 - mae: 2
634.0447
Epoch 19/50
2552/2552 [==============================] - 22s 9ms/step - loss: 11666527.0000 - mae: 2
611.5964
Epoch 20/50
2552/2552 [==============================] - 24s 9ms/step - loss: 11553639.0000 - mae: 2
595.6091
Epoch 21/50
2552/2552 [==============================] - 21s 8ms/step - loss: 11580456.0000 - mae: 2
601.4226
Epoch 22/50
2552/2552 [==============================] - 20s 8ms/step - loss: 11429839.0000 - mae: 2
580.7437
Epoch 23/50
2552/2552 [==============================] - 19s 7ms/step - loss: 11339011.0000 - mae: 2
563.2278
Epoch 24/50
2552/2552 [==============================] - 22s 8ms/step - loss: 11364686.0000 - mae: 2
569.4597
Epoch 25/50
2552/2552 [==============================] - 22s 9ms/step - loss: 11270536.0000 - mae: 2
560.3186
Epoch 26/50
2552/2552 [==============================] - 25s 10ms/step - loss: 11239630.0000 - mae:
2554.1104
Epoch 27/50
2552/2552 [==============================] - 22s 9ms/step - loss: 10945265.0000 - mae: 2
519.4119
Epoch 28/50
2552/2552 [==============================] - 20s 8ms/step - loss: 10965880.0000 - mae: 2
521.8633
Epoch 29/50
2552/2552 [==============================] - 23s 9ms/step - loss: 10845617.0000 - mae: 2
508.1306
Epoch 30/50
2552/2552 [==============================] - 24s 9ms/step - loss: 10900579.0000 - mae: 2
502.7805
Epoch 31/50
2552/2552 [==============================] - 25s 10ms/step - loss: 10873167.0000 - mae:
2502.0906
Epoch 32/50
2552/2552 [==============================] - 23s 9ms/step - loss: 10670382.0000 - mae: 2
478.8625
Epoch 33/50
2552/2552 [==============================] - 22s 9ms/step - loss: 10721810.0000 - mae: 2
483.0398
Epoch 34/50
2552/2552 [==============================] - 21s 8ms/step - loss: 10637930.0000 - mae: 2
```

```
478.6943
Epoch 35/50
2552/2552 [==============================] - 21s 8ms/step - loss: 10567653.0000 - mae: 2
468.8787
Epoch 36/50
2552/2552 [==============================] - 22s 9ms/step - loss: 10310066.0000 - mae: 2
440.6741
Epoch 37/50
2552/2552 [==============================] - 24s 10ms/step - loss: 10410291.0000 - mae:
2449.7058
Epoch 38/50
2552/2552 [==============================] - 21s 8ms/step - loss: 10302490.0000 - mae: 2
427.9324
Epoch 39/50
2552/2552 [==============================] - 22s 9ms/step - loss: 10178174.0000 - mae: 2
429.6443
Epoch 40/50
2552/2552 [==============================] - 19s 7ms/step - loss: 10157773.0000 - mae: 2
421.9255
Epoch 41/50
2552/2552 [==============================] - 20s 8ms/step - loss: 9972416.0000 - mae: 23
93.5261
Epoch 42/50
2552/2552 [==============================] - 23s 9ms/step - loss: 10057025.0000 - mae: 2
405.4097
Epoch 43/50
2552/2552 [==============================] - 21s 8ms/step - loss: 9903513.0000 - mae: 23
87.5205
Epoch 44/50
2552/2552 [==============================] - 21s 8ms/step - loss: 9967490.0000 - mae: 23
94.6750
Epoch 45/50
2552/2552 [==============================] - 22s 8ms/step - loss: 9799242.0000 - mae: 23
75.8181
Epoch 46/50
2552/2552 [==============================] - 22s 9ms/step - loss: 9740825.0000 - mae: 23
59.6445
Epoch 47/50
2552/2552 [==============================] - 21s 8ms/step - loss: 9655248.0000 - mae: 23
52.9683
Epoch 48/50
2552/2552 [==============================] - 19s 8ms/step - loss: 9658199.0000 - mae: 23
52.1152
Epoch 49/50
2552/2552 [==============================] - 20s 8ms/step - loss: 9699630.0000 - mae: 23
50.0366
Epoch 50/50
2552/2552 [==============================] - 22s 8ms/step - loss: 9509477.0000 - mae: 23
32.0300
Epoch 1/50
2552/2552 [==============================] - 24s 9ms/step - loss: 16651116.0000 - mae: 3
214.9995
Epoch 2/50
2552/2552 [==============================] - 23s 9ms/step - loss: 13262073.0000 - mae: 2
828.6345
Epoch 3/50
2552/2552 [==============================] - 21s 8ms/step - loss: 13123500.0000 - mae: 2
795.2195
Epoch 4/50
2552/2552 [==============================] - 21s 8ms/step - loss: 12393229.0000 - mae: 2
```

```
715.8303
Epoch 5/50
2552/2552 [==============================] - 19s 8ms/step - loss: 12291602.0000 - mae: 2
698.0278
Epoch 6/50
2552/2552 [==============================] - 19s 7ms/step - loss: 12191683.0000 - mae: 2
684.8418
Epoch 7/50
2552/2552 [==============================] - 20s 8ms/step - loss: 12082471.0000 - mae: 2
668.7341
Epoch 8/50
2552/2552 [==============================] - 20s 8ms/step - loss: 11847013.0000 - mae: 2
646.2361
Epoch 9/50
2552/2552 [==============================] - 20s 8ms/step - loss: 11678328.0000 - mae: 2
628.8562
Epoch 10/50
2552/2552 [==============================] - 18s 7ms/step - loss: 11765340.0000 - mae: 2
630.7393
Epoch 11/50
2552/2552 [==============================] - 19s 8ms/step - loss: 11580750.0000 - mae: 2
615.1216
Epoch 12/50
2552/2552 [==============================] - 23s 9ms/step - loss: 11741697.0000 - mae: 2
623.8645
Epoch 13/50
2552/2552 [==============================] - 23s 9ms/step - loss: 11496756.0000 - mae: 2
587.8625
Epoch 14/50
2552/2552 [==============================] - 20s 8ms/step - loss: 11421258.0000 - mae: 2
588.5999
Epoch 15/50
2552/2552 [==============================] - 20s 8ms/step - loss: 11269405.0000 - mae: 2
571.1438
Epoch 16/50
2552/2552 [==============================] - 18s 7ms/step - loss: 11281477.0000 - mae: 2
567.5564
Epoch 17/50
2552/2552 [==============================] - 19s 8ms/step - loss: 11123791.0000 - mae: 2
545.5149
Epoch 18/50
2552/2552 [==============================] - 19s 8ms/step - loss: 11291559.0000 - mae: 2
570.3164
Epoch 19/50
2552/2552 [==============================] - 21s 8ms/step - loss: 11109241.0000 - mae: 2
547.9827
Epoch 20/50
2552/2552 [==============================] - 19s 8ms/step - loss: 11048792.0000 - mae: 2
539.5027
Epoch 21/50
2552/2552 [==============================] - 19s 7ms/step - loss: 11015685.0000 - mae: 2
538.5010
Epoch 22/50
2552/2552 [==============================] - 18s 7ms/step - loss: 10865609.0000 - mae: 2
508.1516
Epoch 23/50
2552/2552 [==============================] - 18s 7ms/step - loss: 10765501.0000 - mae: 2
501.6479
Epoch 24/50
2552/2552 [==============================] - 18s 7ms/step - loss: 10752284.0000 - mae: 2
```

```
502.4998
Epoch 25/50
2552/2552 [==============================] - 19s 7ms/step - loss: 10552643.0000 - mae: 2
480.0256
Epoch 26/50
2552/2552 [==============================] - 22s 9ms/step - loss: 10657345.0000 - mae: 2
490.1160
Epoch 27/50
2552/2552 [==============================] - 18s 7ms/step - loss: 10522340.0000 - mae: 2
478.5930
Epoch 28/50
2552/2552 [==============================] - 20s 8ms/step - loss: 10440154.0000 - mae: 2
467.1204
Epoch 29/50
2552/2552 [==============================] - 19s 7ms/step - loss: 10501181.0000 - mae: 2
472.4507
Epoch 30/50
2552/2552 [==============================] - 17s 7ms/step - loss: 10285520.0000 - mae: 2
444.5034
Epoch 31/50
2552/2552 [==============================] - 19s 8ms/step - loss: 10236490.0000 - mae: 2
443.0759
Epoch 32/50
2552/2552 [==============================] - 19s 7ms/step - loss: 10214808.0000 - mae: 2
430.1614
Epoch 33/50
2552/2552 [==============================] - 17s 7ms/step - loss: 10219777.0000 - mae: 2
439.0403
Epoch 34/50
2552/2552 [==============================] - 18s 7ms/step - loss: 10093306.0000 - mae: 2
425.7888
Epoch 35/50
2552/2552 [==============================] - 18s 7ms/step - loss: 10024822.0000 - mae: 2
414.0151
Epoch 36/50
2552/2552 [==============================] - 20s 8ms/step - loss: 10055455.0000 - mae: 2
418.1458
Epoch 37/50
2552/2552 [==============================] - 19s 7ms/step - loss: 10012920.0000 - mae: 2
410.9358
Epoch 38/50
2552/2552 [==============================] - 21s 8ms/step - loss: 9967511.0000 - mae: 24
00.4077
Epoch 39/50
2552/2552 [==============================] - 18s 7ms/step - loss: 9903680.0000 - mae: 23
92.8547
Epoch 40/50
2552/2552 [==============================] - 19s 7ms/step - loss: 9774295.0000 - mae: 23
81.0325
Epoch 41/50
2552/2552 [==============================] - 17s 7ms/step - loss: 9750015.0000 - mae: 23
78.8726
Epoch 42/50
2552/2552 [==============================] - 18s 7ms/step - loss: 9667063.0000 - mae: 23
64.3215
Epoch 43/50
2552/2552 [==============================] - 19s 7ms/step - loss: 9611469.0000 - mae: 23
61.3779
Epoch 44/50
2552/2552 [==============================] - 17s 7ms/step - loss: 9593740.0000 - mae: 23
```

```
49.6831
Epoch 45/50
2552/2552 [==============================] - 17s 7ms/step - loss: 9685295.0000 - mae: 23
62.6877
Epoch 46/50
2552/2552 [==============================] - 18s 7ms/step - loss: 9501405.0000 - mae: 23
41.3247
Epoch 47/50
2552/2552 [==============================] - 17s 7ms/step - loss: 9500192.0000 - mae: 23
45.7485
Epoch 48/50
2552/2552 [==============================] - 17s 7ms/step - loss: 9345075.0000 - mae: 23
20.2959
Epoch 49/50
2552/2552 [==============================] - 18s 7ms/step - loss: 9346962.0000 - mae: 23
18.1101
Epoch 50/50
2552/2552 [==============================] - 18s 7ms/step - loss: 9366630.0000 - mae: 23
16.0906
Epoch 1/50
2552/2552 [==============================] - 19s 7ms/step - loss: 17724354.0000 - mae: 3
392.6123
Epoch 2/50
2552/2552 [==============================] - 18s 7ms/step - loss: 13834603.0000 - mae: 2
904.7654
Epoch 3/50
2552/2552 [==============================] - 17s 7ms/step - loss: 13571716.0000 - mae: 2
859.5750
Epoch 4/50
2552/2552 [==============================] - 16s 6ms/step - loss: 13198320.0000 - mae: 2
808.9026
Epoch 5/50
2552/2552 [==============================] - 18s 7ms/step - loss: 12963592.0000 - mae: 2
774.5454
Epoch 6/50
2552/2552 [==============================] - 17s 7ms/step - loss: 12719637.0000 - mae: 2
745.9656
Epoch 7/50
2552/2552 [==============================] - 16s 6ms/step - loss: 12595771.0000 - mae: 2
731.9524
Epoch 8/50
2552/2552 [==============================] - 17s 7ms/step - loss: 12353411.0000 - mae: 2
702.1763
Epoch 9/50
2552/2552 [==============================] - 16s 6ms/step - loss: 12465279.0000 - mae: 2
710.6641
Epoch 10/50
2552/2552 [==============================] - 17s 7ms/step - loss: 12154789.0000 - mae: 2
681.1543
Epoch 11/50
2552/2552 [==============================] - 17s 7ms/step - loss: 12068186.0000 - mae: 2
667.9905
Epoch 12/50
2552/2552 [==============================] - 14s 6ms/step - loss: 12038731.0000 - mae: 2
661.6218
Epoch 13/50
2552/2552 [==============================] - 14s 6ms/step - loss: 11766366.0000 - mae: 2
632.9817
Epoch 14/50
2552/2552 [==============================] - 18s 7ms/step - loss: 11742184.0000 - mae: 2
```

```
626.5359
Epoch 15/50
2552/2552 [==============================] - 16s 6ms/step - loss: 11863884.0000 - mae: 2
632.7588
Epoch 16/50
2552/2552 [==============================] - 15s 6ms/step - loss: 11795794.0000 - mae: 2
624.3752
Epoch 17/50
2552/2552 [==============================] - 15s 6ms/step - loss: 11637100.0000 - mae: 2
613.5486
Epoch 18/50
2552/2552 [==============================] - 15s 6ms/step - loss: 11425624.0000 - mae: 2
586.1296
Epoch 19/50
2552/2552 [==============================] - 15s 6ms/step - loss: 11483360.0000 - mae: 2
586.3301
Epoch 20/50
2552/2552 [==============================] - 16s 6ms/step - loss: 11357061.0000 - mae: 2
576.0852
Epoch 21/50
2552/2552 [==============================] - 18s 7ms/step - loss: 11287666.0000 - mae: 2
570.8123
Epoch 22/50
2552/2552 [==============================] - 20s 8ms/step - loss: 11150257.0000 - mae: 2
549.4702
Epoch 23/50
2552/2552 [==============================] - 19s 7ms/step - loss: 11129617.0000 - mae: 2
549.4070
Epoch 24/50
2552/2552 [==============================] - 16s 6ms/step - loss: 11169831.0000 - mae: 2
541.6335
Epoch 25/50
2552/2552 [==============================] - 16s 6ms/step - loss: 10998413.0000 - mae: 2
529.9597
Epoch 26/50
2552/2552 [==============================] - 16s 6ms/step - loss: 10893261.0000 - mae: 2
516.9983
Epoch 27/50
2552/2552 [==============================] - 18s 7ms/step - loss: 10879526.0000 - mae: 2
518.4656
Epoch 28/50
2552/2552 [==============================] - 16s 6ms/step - loss: 10806542.0000 - mae: 2
497.3428
Epoch 29/50
2552/2552 [==============================] - 15s 6ms/step - loss: 10736612.0000 - mae: 2
498.8352
Epoch 30/50
2552/2552 [==============================] - 16s 6ms/step - loss: 10575397.0000 - mae: 2
476.2480
Epoch 31/50
2552/2552 [==============================] - 16s 6ms/step - loss: 10574266.0000 - mae: 2
475.4375
Epoch 32/50
2552/2552 [==============================] - 17s 6ms/step - loss: 10389894.0000 - mae: 2
456.4539
Epoch 33/50
2552/2552 [==============================] - 17s 7ms/step - loss: 10583586.0000 - mae: 2
476.5273
Epoch 34/50
2552/2552 [==============================] - 17s 7ms/step - loss: 10226516.0000 - mae: 2
```

```
439.1357
Epoch 35/50
2552/2552 [==============================] - 17s 6ms/step - loss: 10229551.0000 - mae: 2
436.7268
Epoch 36/50
2552/2552 [==============================] - 16s 6ms/step - loss: 10182419.0000 - mae: 2
424.8713
Epoch 37/50
2552/2552 [==============================] - 16s 6ms/step - loss: 10269451.0000 - mae: 2
438.0811
Epoch 38/50
2552/2552 [==============================] - 15s 6ms/step - loss: 10054123.0000 - mae: 2
409.7625
Epoch 39/50
2552/2552 [==============================] - 17s 6ms/step - loss: 10063650.0000 - mae: 2
408.5225
Epoch 40/50
2552/2552 [==============================] - 15s 6ms/step - loss: 9950581.0000 - mae: 23
92.0588
Epoch 41/50
2552/2552 [==============================] - 15s 6ms/step - loss: 9903255.0000 - mae: 23
93.3252
Epoch 42/50
2552/2552 [==============================] - 18s 7ms/step - loss: 9679972.0000 - mae: 23
67.1489
Epoch 43/50
2552/2552 [==============================] - 18s 7ms/step - loss: 9775085.0000 - mae: 23
74.4990
Epoch 44/50
2552/2552 [==============================] - 16s 6ms/step - loss: 9680709.0000 - mae: 23
58.2114
Epoch 45/50
2552/2552 [==============================] - 15s 6ms/step - loss: 9639672.0000 - mae: 23
51.5410
Epoch 46/50
2552/2552 [==============================] - 17s 7ms/step - loss: 9650552.0000 - mae: 23
55.7195
Epoch 47/50
2552/2552 [==============================] - 16s 6ms/step - loss: 9555350.0000 - mae: 23
42.2375
Epoch 48/50
2552/2552 [==============================] - 16s 6ms/step - loss: 9541521.0000 - mae: 23
42.1260
Epoch 49/50
2552/2552 [==============================] - 19s 7ms/step - loss: 9361684.0000 - mae: 23
22.9670
Epoch 50/50
2552/2552 [==============================] - 19s 8ms/step - loss: 9397202.0000 - mae: 23
24.3860
```

In [190…

```
score_kf_ann
RMSE = sum(score_kf_ann)/len(score_kf_ann)
RMSE
```

Out[190…

```
1964.6614023809518
```

In [ ]:

```

```