PROJECT1

Attempting to Predict the Causes of sales growth based on available data

Team Members : Amal ,Aryam,Ghada,Hanuf

STEPS IN DATA UNDERSTANDING, CLEANING, AND MODEL SELECTION

- :Initial Exploration .1

We began by understanding the data, examining its type, size, and identifying the key columns. We also assessed .which columns could be excluded

- :Data Cleaning and Transformation .2

We then proceeded with cleaning the data, converting the “Date” column into three separate columns. Additionally, .we transformed object values into numeric formats (using “float” where applicable)

- :Model Selection and Experimentation .3

Each team member chose a model to work with. The results for each model are attached along

- :Best Model Selection .4

The best performing model was Random Forest, which we applied to the real-data file

INSTALL LIBRARIES,DOWNLOAD CSV

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

[101] ✓ 0.0s Python

```
df = pd.read_csv('sales.csv', index_col=0)
import pandas as pd
rd = pd.read_csv('REAL_DATA.csv', index_col=0)
```

[102] ✓ 0.4s

SHOW FIRST 5 DATA SALES AND SHOW THE INFORMATION FOR EACH COLUMN

```
df.head()
```

[103] ✓ 0.0s

	store_ID	day_of_week	date	nb_customers_on_day	open	promotion	state_holiday	school_holiday	sales
425390	366	4	2013-04-18	517	1	0	0	0	4422
291687	394	6	2015-04-11	694	1	0	0	0	8297
411278	807	4	2013-08-29	970	1	1	0	0	9729
664714	802	2	2013-05-28	473	1	1	0	0	6513
540835	726	4	2013-10-10	1068	1	1	0	0	10882

```
df.info()
```

[108] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
Index: 640840 entries, 425390 to 305711
Data columns (total 11 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   store_ID              640840 non-null  int64
 1   day_of_week           640840 non-null  int64
 2   nb_customers_on_day   640840 non-null  int64
 3   open                  640840 non-null  int64
 4   promotion             640840 non-null  int64
 5   state_holiday         640840 non-null  int64
 6   school_holiday        640840 non-null  int64
 7   sales                 640840 non-null  int64
 8   day                   640840 non-null  int32
 9   month                 640840 non-null  int32
 10  year                  640840 non-null  int32
dtypes: int32(3), int64(8)
memory usage: 51.3 MB
```

CHECK EMPTY VALUS AND SHOW DESCRIPTION

```
[109]  df.isnull().sum()
       ✓ 0.0s
```

```
store_ID                0
day_of_week             0
nb_customers_on_day     0
open                    0
promotion               0
state_holiday           0
school_holiday          0
sales                   0
day                     0
month                   0
year                    0
dtype: int64
```

```
[110]  df.describe()
       ✓ 0.2s                                                                    Python
```

	store_ID	day_of_week	nb_customers_on_day	open	promotion	state_holiday	school_holiday	sales	day	640
count	640840.000000	640840.000000	640840.000000	640840.000000	640840.000000	640840.00000	640840.000000	640840.000000	640840.000000	
mean	558.211348	4.000189	633.398577	0.830185	0.381718	0.03071	0.178472	5777.469011	15.711689	
std	321.878521	1.996478	464.094416	0.375470	0.485808	0.17253	0.382910	3851.338083	8.791182	
min	1.000000	1.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	1.000000	
25%	280.000000	2.000000	405.000000	1.000000	0.000000	0.00000	0.000000	3731.000000	8.000000	
50%	558.000000	4.000000	609.000000	1.000000	0.000000	0.00000	0.000000	5746.000000	16.000000	
75%	837.000000	6.000000	838.000000	1.000000	1.000000	0.00000	0.000000	7860.000000	23.000000	
max	1115.000000	7.000000	5458.000000	1.000000	1.000000	1.00000	1.000000	41551.000000	31.000000	

CONVORT DATE, STATE HOLIDAY

```
df['date'] = pd.to_datetime(df['date'])

df['day'] = df['date'].dt.day
df['month'] = df['date'].dt.month
df['year'] = df['date'].dt.year

df=df.drop(columns= ['date'], axis=1)
```

✓ 0.1s

```
df['state_holiday'] = df['state_holiday'].map({'0': 0, 'a': 1, 'b': 1, 'c': 1})
df['school_holiday'] = df['school_holiday'].astype(int)
```

✓ 0.0s

SHOW DATA AFTER CHANGE

```
df.head()
```

✓ 0.0s

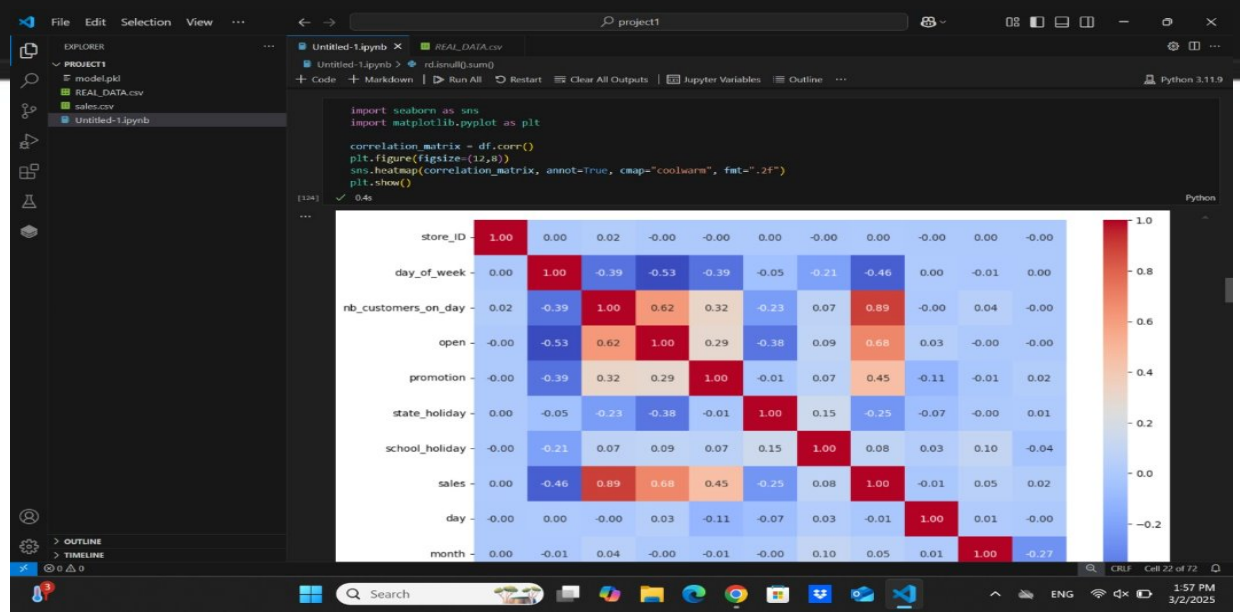
	store_ID	day_of_week	nb_customers_on_day	open	promotion	state_holiday	school_holiday	sales	day	month	year
425390	366	4	517	1	0	0	0	4422	18	4	2013
291687	394	6	694	1	0	0	0	8297	11	4	2015
411278	807	4	970	1	1	0	0	9729	29	8	2013
664714	802	2	473	1	1	0	0	6513	28	5	2013
540835	726	4	1068	1	1	0	0	10882	10	10	2013

TO KNOW THE IMPORTANT COLMNS

```
import seaborn as sns
import matplotlib.pyplot as plt

correlation_matrix = df.corr()
plt.figure(figsize=(12,8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.show()
```

✓ 0.4s

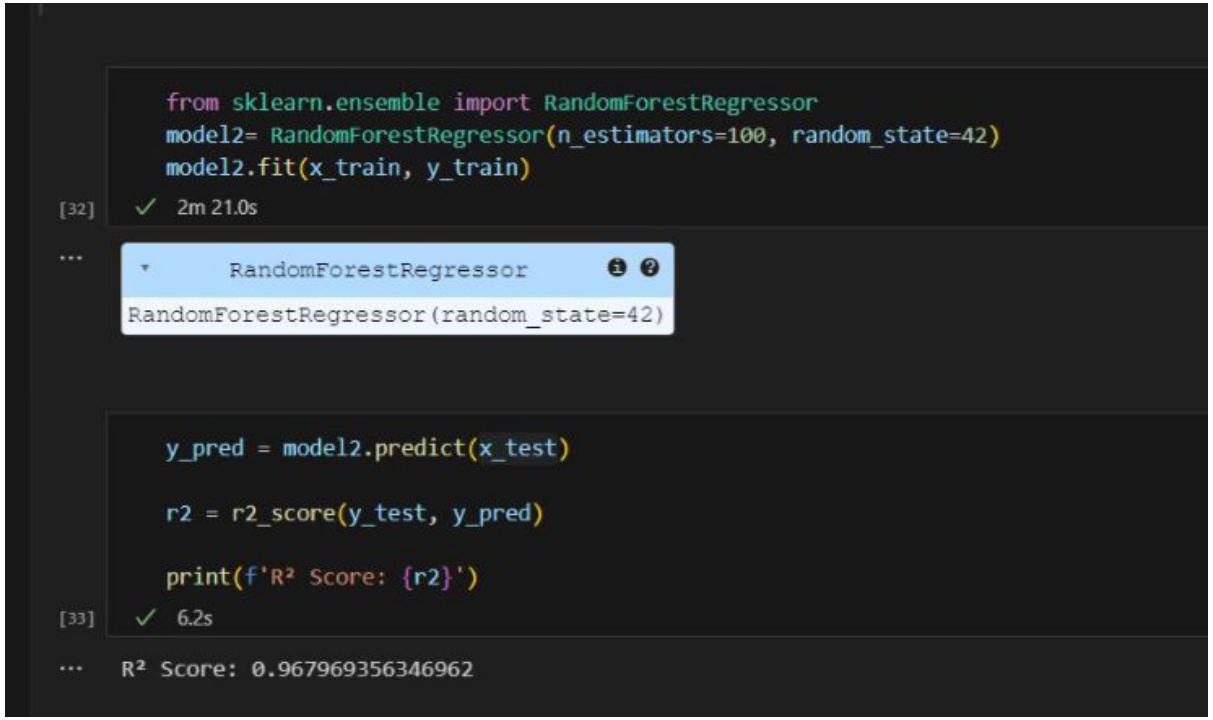


SPLIT

```
x = df[['store_ID', 'day_of_week', 'nb_customers_on_day', 'open','state_holiday', 'promotion']]
y = df['sales']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

✓ 0.0s

FIT,PREDCTION ,SCORE AND THIS A BEST MODEL



SHOW REAL DATA

```
rd.head()
```

[112] ✓ 0.0s

...

index	store_ID	day_of_week	date	nb_customers_on_day	open	promotion	state_holiday	school_holiday
272371	415	7	01/03/2015	0	0	0	0	0
558468	27	7	29/12/2013	0	0	0	0	0
76950	404	3	19/03/2014	657	1	1	0	0
77556	683	2	29/01/2013	862	1	0	0	0
456344	920	3	19/03/2014	591	1	1	0	0

INFORMATION OF THE REAL DATA AND CHANGE COLUMN DATA AND STAT HOLIDAY TO INT

```
rd.info()
```

[113] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
Index: 71205 entries, 272371 to 85695
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   store_ID             71205 non-null  int64
 1   day_of_week          71205 non-null  int64
 2   date                 71205 non-null  object
 3   nb_customers_on_day  71205 non-null  int64
 4   open                 71205 non-null  int64
 5   promotion            71205 non-null  int64
 6   state_holiday        71205 non-null  object
 7   school_holiday       71205 non-null  int64
dtypes: int64(6), object(2)
memory usage: 4.9+ MB
```

```
rd['date'] = pd.to_datetime(rd['date'], format="%d/%m/%Y", errors='coerce')

rd['day'] = rd['date'].dt.day
rd['month'] = rd['date'].dt.month
rd['year'] = rd['date'].dt.year

rd = rd.drop(columns=['date'], axis=1)
```

[114] ✓ 0.1s

```
rd['state_holiday'] = rd['state_holiday'].map({'0': 0, 'a': 1, 'b': 1, 'c': 1})
rd['school_holiday'] = rd['school_holiday'].astype(int)
```

[115] ✓ 0.0s

AFTER MODIFICATION

```
rd.head()
```

[116] ✓ 0.0s

index	store_ID	day_of_week	nb_customers_on_day	open	promotion	state_holiday	school_holiday	day	month	year
272371	415	7	0	0	0	0	0	1	3	2015
558468	27	7	0	0	0	0	0	29	12	2013
76950	404	3	657	1	1	0	0	19	3	2014
77556	683	2	862	1	0	0	0	29	1	2013
456344	920	3	591	1	1	0	0	19	3	2014

```
rd.info()
```

[118] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
Index: 71205 entries, 272371 to 85695
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   store_ID            71205 non-null  int64
 1   day_of_week         71205 non-null  int64
 2   nb_customers_on_day 71205 non-null  int64
 3   open                71205 non-null  int64
 4   promotion           71205 non-null  int64
 5   state_holiday       71205 non-null  int64
 6   school_holiday      71205 non-null  int64
 7   day                 71205 non-null  int32
 8   month               71205 non-null  int32
 9   year                71205 non-null  int32
dtypes: int32(3), int64(7)
memory usage: 5.2 MB
```

CONFIRM THAT THERE ARE NO MISSING VALUE

```
rd.isnull().sum()
```

[119] ✓ 0.0s

```
store_ID              0
day_of_week           0
nb_customers_on_day   0
open                  0
promotion             0
state_holiday         0
school_holiday        0
day                   0
month                 0
year                  0
dtype: int64
```

CONCAT THA SALES DATA WITH REAL DATA AND SAVE SALES IN THA REAL DATA FILE

```
f=['store_ID', 'day_of_week', 'nb_customers_on_day', 'open','state_holiday', 'promotion']
X_new = rd [f]

predictions = model2.predict(X_new)


rd['sales'] = predictions
```

[34] ✓ 3.4s

```
rd.to_csv('REAL_DATA.csv')
```

[36] ✓ 0.1s

THANK YOU