future_sales_prediction

December 23, 2025

1 Future Sales Prediction with Machine Learning

[6]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import plotly.express as px
import plotly.graph_objects as go
```

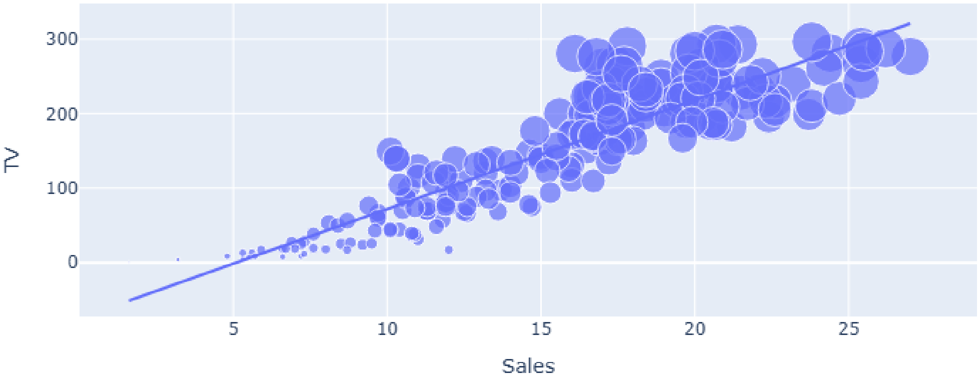
[4]:
```
data = pd.read_csv("advertising.csv")
data.head()
```

[4]:
```
      TV  Radio  Newspaper  Sales
0  230.1   37.8       69.2   22.1
1   44.5   39.3       45.1   10.4
2   17.2   45.9       69.3   12.0
3  151.5   41.3       58.5   16.5
4  180.8   10.8       58.4   17.9
```

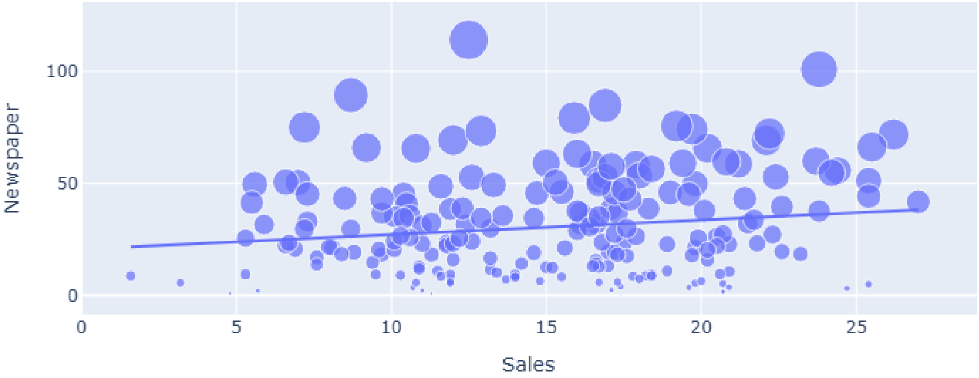
[5]:
```
# Lets check check if this dataset contains any Null values or not
print(data.isnull().sum())
```

```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

[7]:
```
#relationship between the Sales and the advertising on TV
figure = px.scatter(data_frame = data, x = "Sales",
                    y = "TV", size = "TV", trendline = "ols"
                    )
figure.show()
```

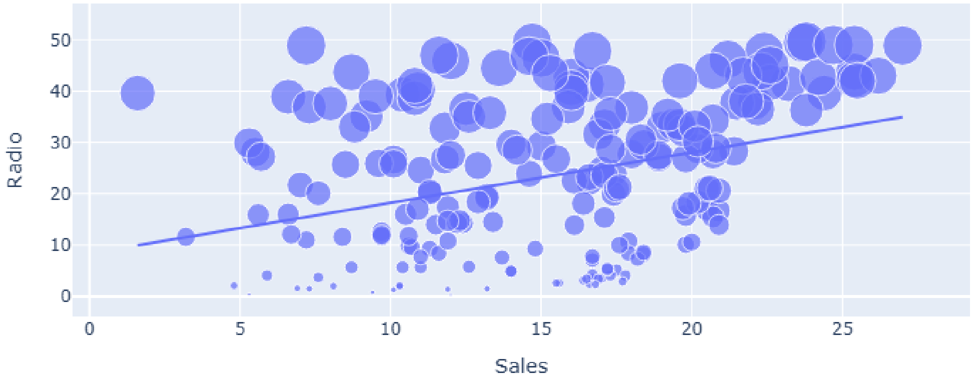


[8]:
```
#relationship between the Sales and the advertising on Newspaper
figure = px.scatter(data_frame = data, x = "Sales",
                    y = "Newspaper", size = "Newspaper", trendline = "ols")
figure.show()
```



[9]:
```
##relationship between the Sales and the advertising on Radio
figure = px.scatter(data_frame = data, x = "Sales",
                    y = "Radio", size = "Radio", trendline = "ols")

figure.show()
```



[10]:

```
#lets look at the correlation of all the columns with the sales column
correlation  = data.corr()
print(correlation["Sales"].sort_values(ascending = False))
```

```
Sales        1.000000
TV           0.901208
Radio        0.349631
Newspaper    0.157960
Name: Sales, dtype: float64
```

1.1 Future Sales Prediction Model

[14]:

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

xtrain, xtest, ytrain, ytest =  train_test_split(x,y,
                                                test_size = 0.2,
                                                random_state = 42)
```

[15]:

```
model = LinearRegression()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
```

```
0.9059011844150826
```

[16]:

```
# Lets do some prediction, accroding the training features
#features = [[TV, Radio, Newspaper]]
features = np.array([[230.1, 37.8, 69.2]])
print(model.predict(features))
```

```
[21.37254028]
```

[ ]:

```
```