In [1]:

```
import pandas as pd
from sklearn import metrics
from sklearn.linear_model import LogisticRegression
from sklearn import neighbors
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
dff=pd.read_csv(r"C:\Users\Akhilganga\OneDrive\Desktop\Stores.csv")
 dff.head()
```

In [3]:

```
dff.tail()
```

Out[3]:

	Store ID	Store_Area	Items_Available	Daily_Customer_Count	Store_Sales
891	892	1582	1910	1080	66390
892	893	1387	1663	850	82080
893	894	1200	1436	1060	76440
894	895	1299	1560	770	96610
895	896	1174	1429	1110	54340

In [4]:

```
dff.isnull().sum()
```

Out[4]:

```
Store ID                0
Store_Area              0
Items_Available         0
Daily_Customer_Count    0
Store_Sales             0
dtype: int64
```

In [5]:

```
dff.duplicated().sum()
```

Out[5]:

```
0
```

In [6]:

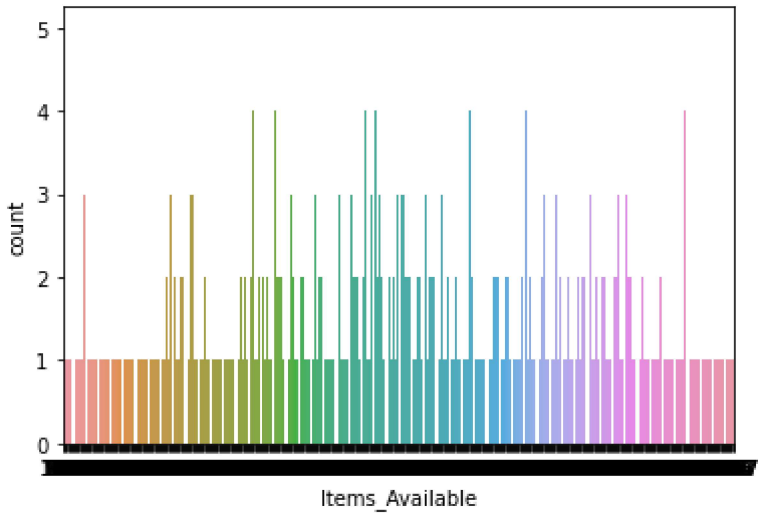
```
dff.shape
```

Out[6]:

```
(896, 5)
```

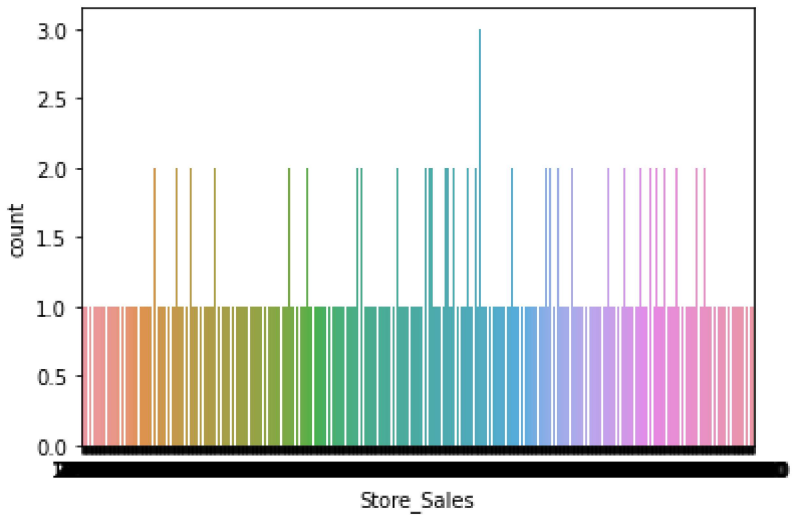
In [7]:

```
ax = sns.countplot(x="Items_Available",data=dff)
```



In [8]:

```
ax = sns.countplot(x="Store_Sales",data=dff)
```



In [9]:

```
dff.dtypes
```

Out[9]:

```
Store ID                int64
Store_Area              int64
Items_Available         int64
Daily_Customer_Count    int64
Store_Sales             int64
dtype: object
```

In [15]:

```
x = dff.drop(['Store_Area'],axis=1)
y = dff.Store_Area
```

In [16]:

```
x_train = dff.drop(['Store_Area'],axis=1)
y_train = dff.Store_Area
```

In [12]:

```
from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score
```

In [17]:

```
 x_train, x_test, y_train, y_test=train_test_split(x, y, random_state=0, test_size=0.2)
```

In [32]:

```
from sklearn.metrics import mean_squared_error
import numpy as np
```

In [33]:

```

for i in range(200):
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=
from sklearn.tree import DecisionTreeRegressor
lm=DecisionTreeRegressor()
lm.fit(x_train,y_train)
y_pred=lm.predict(x_test)
mse=mean_squared_error(y_test,y_pred)
rmse = np.sqrt(mse)
print(rmse)


```

```
19.47546967133062
```

In [37]:

```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
regressor.fit(x_train,y_train)
y_pred_regressor = regressor.predict(x_test)
y_pred = regressor.predict(x_test)
y_pred
dt_accuracy = round(regressor.score(x_train,y_train)*100,2)
dt_accuracy
```

Out[37]:

```
100.0
```

In [38]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train,y_train)
```

Out[38]:

```
LinearRegression()
```

In [39]:

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

In [40]:

```
y_pred
```

Out[40]:

```
array([1463.14349695,  982.8520913 , 1490.83342611, 1917.28521292,
       1548.50435342, 1414.49283229, 1713.21780771, 1252.71210852,
       1475.84260774, 1748.59287128, 1238.59991049, 1771.56193094,
       1599.29356686, 1486.54056212, 1577.58949089, 1461.10417586,
       1229.15756962, 1619.52250148, 1791.78428468, 2063.20671976,
       1686.40124571, 1258.46434382, 1266.91772799, 1216.10098129,
       1979.39062035, 1302.39968031, 1148.5967616 , 1590.46347255,
       1471.58693504, 1612.75504954, 1775.26188855, 1195.17679491,
       1400.81090681, 1382.91309548, 1971.033423  , 1575.49816072,
       1403.24809784, 1290.59963386, 2054.07056018, 1126.02074833,
       1560.43901918, 1146.18771673, 1635.3957501 , 1694.98529288,
       1463.19514282, 1641.08723374, 1384.49910567, 1530.36877268,
       1075.24815243, 1727.04440235, 1209.81557808, 1312.30237425,
       1244.47372973, 1142.36952972, 1424.16538893, 1590.73866605,
       1443.90039457, 1952.32366549, 1177.33770786, 1103.4433988 ,
       1336.38634583, 1408.90614014, 1876.65852628, 1778.82802404,
       1479.51934764, 1531.2361397 , 1176.89666485, 1031.03972428,
       1467.28369349, 1066.17065038, 1307.90353772, 1641.55203366,
       1380.35188634, 1978.16190969, 1332.71796457, 1924.89951923,
       1678.60576678, 1536.84563011, 1341.53843183, 1448.51417993,
       1231.1483995 , 1594.78181162, 1552.72790161, 1459.37735263,
       1777.41504976, 1683.92626447, 1196.89706626, 1521.39034243,
        946.70363613, 1258.68071298, 1359.24914225, 1500.40165735,
       1886.34500479, 1766.75561696, 1578.61978387, 1217.67875451,
       1439.81922933, 1631.54170117, 1451.43463009, 1344.01115952,
       1418.04633994, 1341.8233957 , 1164.66663138, 1588.03870558,
       1622.38406035, 1597.11691654, 1512.45069885, 1614.74483968,
       1573.74807538, 1839.54098395, 1795.14124488, 1702.93554873,
       1251.20965838, 1796.71332914, 1302.87748   , 1469.53028101,
       1269.33595222, 1576.71694576, 1215.71298155, 1873.97081747,
       1949.71879478, 1415.72497202, 1189.74245012, 1442.0934499 ,
       1829.23986509, 1146.8700205 , 1494.05399786,  896.78084607,
       1566.60098822, 1712.96811734, 1192.74580528, 1598.9350158 ,
       1541.06820022, 1649.3234403 , 1475.34213564, 1871.91409584,
       1836.68501883, 1491.38111291, 1201.6441487 , 1216.62421245,
       1432.81949538, 1299.47763069, 1626.60343901, 1451.98106617,
       1146.12134778, 1291.79765673, 1243.26476781, 1298.01504464,
       1269.46102595, 1295.80287055, 1444.98797123, 1102.57713643,
       1407.97287347, 1634.71884532, 1999.99712608, 1362.81903021,
       1951.39488825, 1474.40194633, 1572.22153923, 1758.52887163,
       1870.57008918, 1455.02036422, 1549.75573069, 1454.93145614,
       1897.04315081, 1399.50967773, 1433.51861157, 1868.45836175,
       1360.61650353, 1447.66746232, 1564.40174342, 1534.68962263,
       1227.24123247, 1477.1157859 , 1530.68586209, 1063.88238571,
       1622.32072405, 1438.58722261, 1636.35998462, 1547.84405115,
       1340.68400606, 1491.37494168, 1592.99584399, 1242.14980158,
       1777.5912009 , 1878.05111061, 1750.08135815,  907.25161634,
       1411.64306793, 1997.56364907, 1622.59601022, 1715.49962697,
       1596.97223215, 1585.78002661, 1455.64225663, 1568.49165813,
       1244.0831181 , 1503.28462525, 1425.22633264, 1622.006557  ,
       1966.98387033, 1486.14741535, 1589.62309102, 1583.59179015,
       1240.61307485, 1717.31471519, 1435.15296256, 1603.47026683,
       1320.3381459 ,  778.49228802, 1340.39235794, 1536.23004253,
       1406.19094677, 1515.07754533, 1211.91057953, 1821.28686139,
```

```
       1620.89297665, 1682.30387095, 1901.1886497 , 1729.05680594,
       1230.01770748, 1205.00200185, 1706.16032374, 1337.13292041,
       1374.94988201, 2006.91232956, 1878.35024406, 1504.94225043,
       1437.40014697, 1338.12252268, 1297.7413889 , 2182.04955232,
       1790.35860567, 1886.78042365, 1927.14915337, 2075.55677192,
       1782.76811791, 1518.42550307,  977.84688824, 1322.25092285,
       1706.51069855, 1878.4327075 , 1456.22815862, 1537.94484108,
       1424.88943945, 1457.22120353, 1251.00969369, 1627.53947825,
       1855.88884597, 1513.58065757, 1438.85974349, 1459.33959438,
       1621.14656819, 1693.80737399, 1275.28789748, 1811.46493602,
       1119.42483288, 1719.8795898 , 1549.64763838, 1738.46639532,
       1541.538761  , 1746.30729894, 1528.55731723, 1151.03736859,
       1470.26117409, 1657.62134995, 1447.22144396, 1062.20353505,
       1128.47667694, 1526.6371537 , 1255.24556346, 1703.02237373,
       1419.47308234, 1498.00888097, 1995.99355344, 1120.93826108,
       1519.54172654, 1475.61849231,  935.13106002, 1665.04749533,
       1455.95730882, 2222.40296198, 1776.84584244, 1805.62162469,
       1535.67685389, 1996.35945191, 1886.98582657, 1722.19060388,
       1506.33784196, 1178.07437766, 1693.5431619 , 1382.91240553,
       1195.85948083, 1526.35837195, 1221.63580956, 1557.21872164])
```

In [41]:

```
lr_accuracy = round(regressor.score(x_train,y_train)*100,2)
lr_accuracy
```

Out[41]:

```
99.78
```

In [42]:

```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor()
regressor.fit(x_train,y_train)
```

Out[42]:

```
RandomForestRegressor()
```

In [43]:

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

Out[43]:

```
array([1456.94, 1002.72, 1500.46, 1913.2 , 1547.9 , 1414.57, 1700.67,
       1256.78, 1483.44, 1744.71, 1243.36, 1776.1 , 1597.13, 1481.58,
       1581.05, 1449.95, 1221.09, 1619.67, 1785.2 , 2044.  , 1690.9 ,
       1258.5 , 1263.11, 1218.69, 1970.09, 1300.16, 1148.48, 1593.39,
       1469.48, 1612.93, 1781.58, 1189.76, 1393.95, 1382.25, 1961.32,
       1578.02, 1391.04, 1292.15, 2049.44, 1127.84, 1558.26, 1142.6 ,
       1641.65, 1696.34, 1467.  , 1632.93, 1377.68, 1526.72, 1073.22,
       1732.13, 1214.96, 1318.61, 1244.03, 1144.26, 1429.65, 1588.18,
       1446.84, 1942.26, 1170.59, 1105.72, 1338.33, 1403.25, 1866.87,
       1786.81, 1481.17, 1534.31, 1169.61, 1024.82, 1471.52, 1062.63,
       1306.46, 1631.35, 1373.13, 1972.92, 1332.6 , 1915.07, 1668.55,
       1545.29, 1336.48, 1447.02, 1231.55, 1595.4 , 1545.49, 1455.24,
       1775.62, 1663.57, 1192.26, 1512.48,  939.75, 1256.89, 1363.74,
       1495.68, 1881.76, 1783.75, 1579.8 , 1211.12, 1430.6 , 1637.16,
       1445.21, 1340.41, 1426.28, 1339.06, 1161.86, 1594.09, 1630.73,
       1590.02, 1527.6 , 1612.81, 1576.99, 1840.06, 1797.44, 1712.34,
       1250.12, 1800.89, 1298.82, 1468.99, 1262.76, 1580.31, 1218.87,
       1868.45, 1940.35, 1422.18, 1186.16, 1454.87, 1839.51, 1143.39,
       1501.94,  904.13, 1561.16, 1701.38, 1186.86, 1599.21, 1546.83,
       1646.18, 1483.34, 1867.74, 1842.1 , 1494.62, 1209.41, 1213.67,
       1429.52, 1298.03, 1630.63, 1456.22, 1140.69, 1293.25, 1245.9 ,
       1299.96, 1265.12, 1297.08, 1441.4 , 1105.03, 1402.79, 1642.82,
       2007.66, 1368.18, 1939.87, 1482.32, 1575.61, 1764.11, 1864.07,
       1455.75, 1551.16, 1455.69, 1892.17, 1393.36, 1426.15, 1866.48,
       1367.32, 1444.45, 1570.28, 1537.47, 1220.61, 1485.25, 1532.2 ,
       1064.6 , 1624.54, 1430.41, 1642.92, 1549.18, 1330.42, 1507.27,
       1600.78, 1245.47, 1782.45, 1886.21, 1751.9 ,  913.56, 1410.91,
       2006.86, 1625.16, 1720.49, 1596.18, 1590.43, 1454.24, 1564.22,
       1246.44, 1498.29, 1431.99, 1624.19, 1962.75, 1481.16, 1586.71,
       1578.22, 1244.69, 1728.34, 1432.89, 1605.35, 1319.87,  848.93,
       1332.61, 1546.67, 1395.34, 1524.15, 1217.59, 1822.05, 1622.58,
       1666.18, 1902.79, 1733.77, 1231.56, 1209.68, 1712.47, 1336.76,
       1373.43, 2006.13, 1874.76, 1511.33, 1425.43, 1335.7 , 1301.04,
       2152.73, 1784.71, 1882.73, 1917.2 , 2045.59, 1773.71, 1523.71,
        954.29, 1324.95, 1710.64, 1875.47, 1459.27, 1549.37, 1426.2 ,
       1451.77, 1253.73, 1626.78, 1858.84, 1525.17, 1430.78, 1457.31,
       1621.27, 1701.2 , 1280.03, 1821.99, 1120.31, 1722.62, 1542.05,
       1746.81, 1542.78, 1727.83, 1536.53, 1153.21, 1467.12, 1660.14,
       1450.94, 1064.33, 1133.53, 1535.57, 1250.69, 1708.54, 1425.83,
       1499.55, 2007.17, 1116.84, 1515.98, 1488.79,  921.32, 1660.65,
       1454.86, 2154.65, 1783.59, 1810.02, 1540.11, 2006.23, 1875.5 ,
       1720.68, 1516.07, 1177.42, 1695.25, 1376.04, 1186.03, 1533.73,
       1211.44, 1557.72])
```

In [44]:

```
rf_accuracy = round(regressor.score(x_train,y_train)*100,2)
rf_accuracy
```

Out[44]:

```
99.94
```

In [45]:

```
from sklearn.model_selection import cross_val_score
```

In [46]:

```
print(cross_val_score(regressor,x,y,cv=5).mean())
```

```
0.9969102486755658
```

In [47]:

```
from sklearn.model_selection import GridSearchCV
```

In [48]:

```
parameter = {"max_depth":[1,3,5,7,9,11,12],
             'criterion':['mse','friedman_mse']}
```

In [49]:

```
GCV = GridSearchCV(DecisionTreeRegressor(),parameter,cv=5)
```

In [50]:

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

Out[50]:

```
GridSearchCV(cv=5, estimator=DecisionTreeRegressor(),
             param_grid={'criterion': ['mse', 'friedman_mse'],
                         'max_depth': [1, 3, 5, 7, 9, 11, 12]})
```

In [51]:

```
GCV.best_params_
```

Out[51]:

```
{'criterion': 'mse', 'max_depth': 7}
```

In [52]:

```
Final_mod = DecisionTreeRegressor()
Final_mod.fit(x_train,y_train)
pred = Final_mod.predict(x_test)
print((regressor.score(x_test,y_test)*100))
```

```
99.67199500256596
```

In [53]:

```
import joblib
joblib.dump(Final_mod,"FinalModle.pkl")
```

Out[53]:

```
['FinalModle.pkl']
```

In [54]:

```
preds = regressor.predict(x_test)
print(preds[:36])
print(y_test[:36].values)
```

```
[1456.94 1002.72 1500.46 1913.2  1547.9  1414.57 1700.67 1256.78 1483.44
 1744.71 1243.36 1776.1  1597.13 1481.58 1581.05 1449.95 1221.09 1619.67
 1785.2  2044.   1690.9  1258.5  1263.11 1218.69 1970.09 1300.16 1148.48
 1593.39 1469.48 1612.93 1781.58 1189.76 1393.95 1382.25 1961.32 1578.02]
[1461  986 1481 1936 1526 1415 1703 1235 1462 1733 1236 1797 1611 1485
 1613 1442 1222 1631 1807 2044 1671 1249 1270 1218 1978 1307 1150 1582
 1455 1633 1770 1199 1406 1378 1979 1566]
```

In [ ]:

In [ ]:

In [ ]: