**Assignment-2**

Name: Dasari Abhiram

REG NO.: 41733021

Branch: CSE-DS

University: Sathyabama University

Aim: Building an AI model to predict best-selling manga.

Algorithm:

1. Import all the required libraries.
2. Take the dataset, remove all the duplicates.
3. Print the information of the dataframe.
4. Calculate the percentage of data that is missing and print it.
5. Plot a graph on the sales of manga.
6. Drop the unused columns.
7. Build a model, train it , test it and plot the performance graph of the model.

Code:

**I/p-1:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import mean\_absolute\_error

from sklearn.metrics import mean\_squared\_error

from sklearn.model\_selection import train\_test\_split

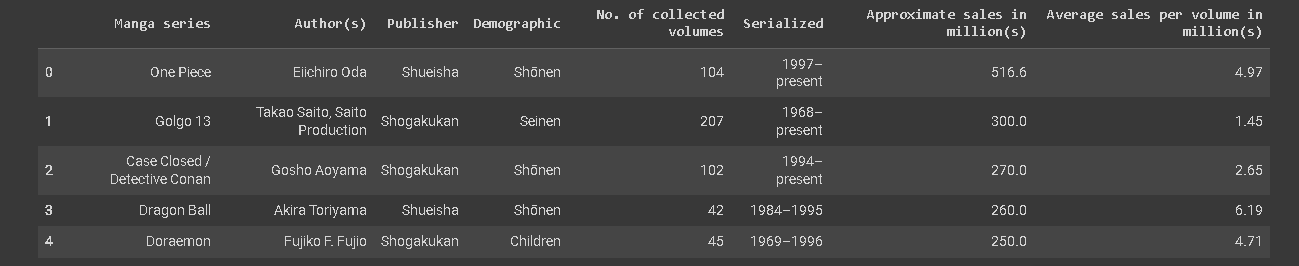
from sklearn.ensemble import RandomForestRegressor

**I/p-2:**

df = pd.read\_csv("/content/best-selling-manga.csv").drop\_duplicates()

df.head()

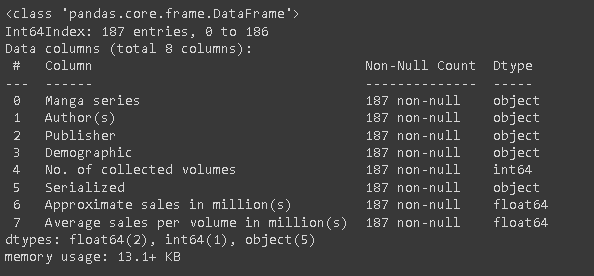
**O/P-1:**



I/p-3:

df.info()

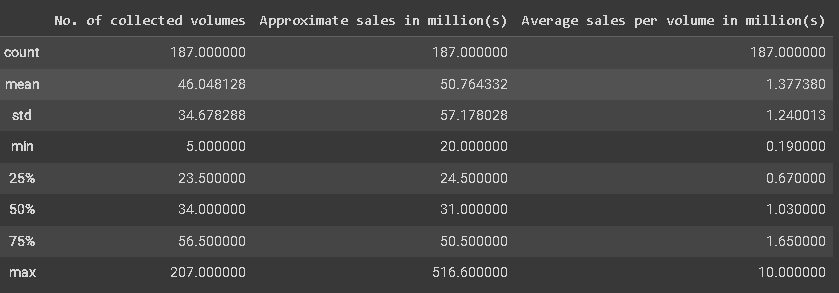
**O/p-2:**

****

**I/p-4:**

df.describe()

**O/p-3:**

****

**I/p-5:**

total\_missing = (df.isnull().sum()).sum()

total\_cells = np.product(df.shape)

# percent of data that is missing

percent\_missing = (total\_missing/total\_cells) \* 100

print("Percent of data that is missing : %d " % percent\_missing)

**O/p-4:**

Percent of data that is missing : 0

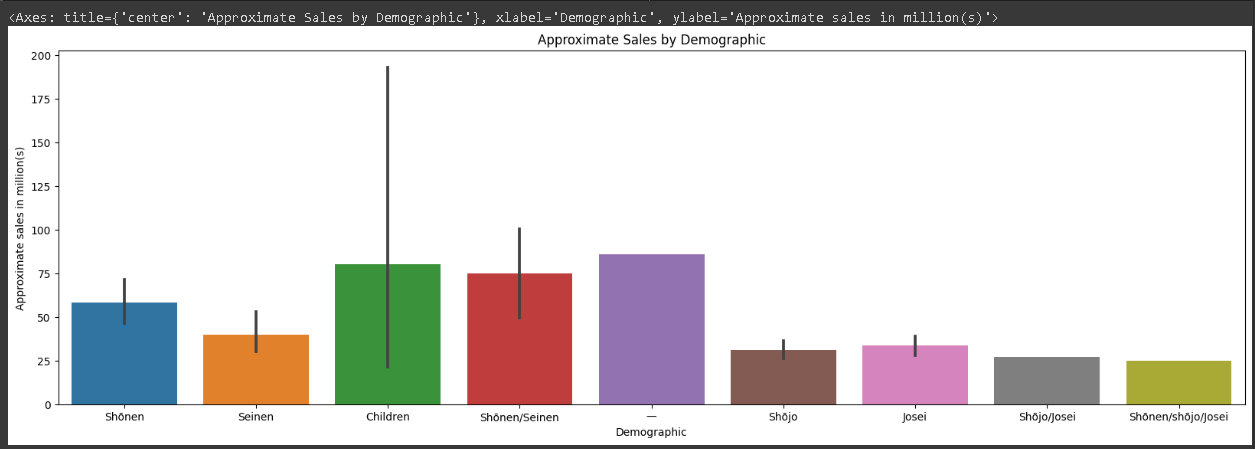
**I/p-6:**

plt.figure(figsize = (20,6))

plt.title('Approximate Sales by Demographic')

sns.barplot(data=df, x='Demographic' , y='Approximate sales in million(s)')

**O/p-5:**

****

**I/p-7:**

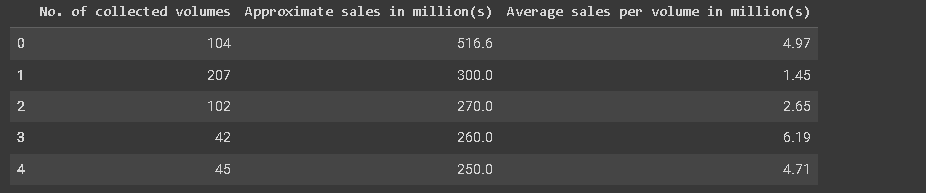
#Drop unused columns

target = 'Approximate sales in million(s)'

df = df.drop(['Manga series','Author(s)','Publisher','Demographic','Serialized'], axis=1)

df.head()

**O/p-6:**

****

**I/p-7:**

X = df.drop([target],axis=1)

y = df[target]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.5, random\_state=0)

model = RandomForestRegressor(n\_estimators=100, random\_state=0, min\_samples\_split=2)

model.fit(X\_train,y\_train)

y\_train\_predict = model.predict(X\_train)

train\_mae = mean\_absolute\_error(y\_train\_predict, y\_train)

print("MAE", train\_mae)

y\_test\_predict = model.predict(X\_test)

test\_mae = mean\_absolute\_error(y\_test\_predict, y\_test)

print("MAE", test\_mae)

**O/p-7:**

MAE 5.796972043010756

MAE 13.227430851063827

**I/p-8:**

train\_mse = mean\_squared\_error(y\_train, y\_train\_predict)

test\_mse = mean\_squared\_error(y\_test, y\_test\_predict)

# Placer les performances dans une liste

performance\_labels = ['MAE (Train)', 'MAE (Test)', 'MSE (train)', 'MSE (Test)']

performance\_values = [train\_mae, test\_mae, train\_mse, test\_mse]

plt.figure(figsize=(8, 5))

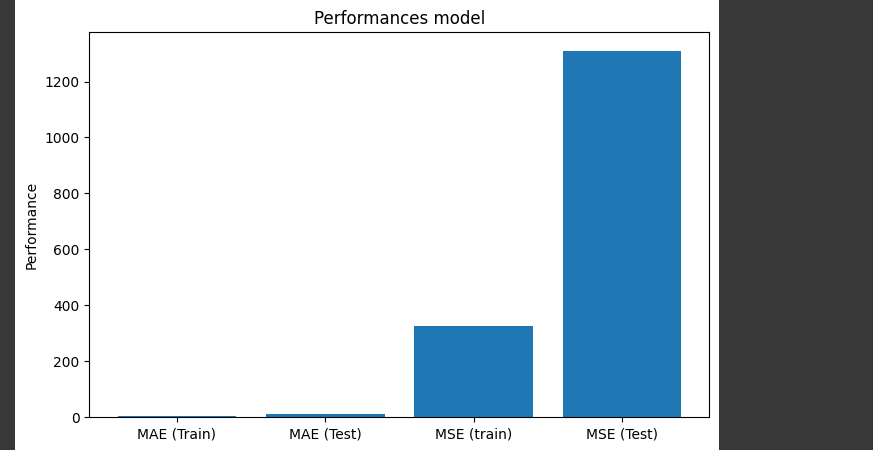
plt.bar(performance\_labels, performance\_values)

plt.ylabel('Performance')

plt.title('Performances model')

plt.show()

**O/p-7:**



**Result:**

Therefore, thebest-selling manga is one piece , 2nd best is Golgo 13.