In [376...
```
import numpy as np
import pandas as pd
```

In [377...
```
x.isna().value_counts()
```

Out[377...
```
GRE Score  TOEFL Score  University Rating  SOP    LOR    CGPA   Research  Chance of Admit
False      False        False              False  False  False  False     False             500
dtype: int64
```

In [378...
```
x = pd.read_csv(r"D:\PG-DAI\MachineLearning\Dec 23 SVM\Admission_Prediction.csv")
```

In [379...
```
x['GRE Score'].replace(to_replace=np.nan, value=x['GRE Score'].mean(), inplace=True, limit=None, regex=False, method='pad')
```

In [380...
```
x['TOEFL Score'].replace(to_replace=np.nan, value=x['TOEFL Score'].mean(), inplace=True, limit=None, regex=False, method='pad')
```

In [381...
```
x['University Rating'].replace(to_replace=np.nan, value=x['University Rating'].mean(), inplace=True, limit=None, regex=False, meth
```

In [382...
```
x
```

Out[382...

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337.000000	118.0	4.0	4.5	4.5	9.65	1	0.92
1	2	324.000000	107.0	4.0	4.0	4.5	8.87	1	0.76
2	3	316.558763	104.0	3.0	3.0	3.5	8.00	1	0.72
3	4	322.000000	110.0	3.0	3.5	2.5	8.67	1	0.80
4	5	314.000000	103.0	2.0	2.0	3.0	8.21	0	0.65
...	...	...	...	...	...	...	...	...	...
495	496	332.000000	108.0	5.0	4.5	4.0	9.02	1	0.87
496	497	337.000000	117.0	5.0	5.0	5.0	9.87	1	0.96

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
497	498	330.000000	120.0	5.0	4.5	5.0	9.56	1	0.93
498	499	312.000000	103.0	4.0	4.0	5.0	8.43	0	0.73
499	500	327.000000	113.0	4.0	4.5	4.5	9.04	0	0.84

500 rows × 9 columns

In [383...
```
del x['Serial No.']
```

In [384...
```
from sklearn.model_selection import train_test_split
data=x.iloc[:,:-1]
target=x['Chance of Admit']
x_train,x_test,y_train,y_test=train_test_split(data,target,test_size=0.20)
```

In [385...
```
from sklearn.svm import SVR

model = SVR()
```

In [386...
```
# x_train = x_train.values.reshape(-1,1)
# x_test = x_train.reshape(-1,1)
```

In [387...
```
x_train
```

Out[387...

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
239	299.0	100.0	1.0	1.5	2.0	7.89	0
173	323.0	113.0	4.0	4.0	4.5	9.23	1
453	319.0	103.0	3.0	2.5	4.0	8.76	1
104	326.0	112.0	3.0	3.5	3.0	9.05	1
21	325.0	114.0	4.0	3.0	2.0	8.40	0

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
...	...	...	...	...	...	...	...
139	318.0	109.0	1.0	3.5	3.5	9.12	0
347	299.0	94.0	1.0	1.0	1.0	7.34	0
457	295.0	99.0	1.0	2.0	1.5	7.57	0
325	326.0	116.0	3.0	3.5	4.0	9.14	1
25	340.0	120.0	5.0	4.5	4.5	9.60	1

400 rows × 7 columns

In [388...
```
model.fit(x_train,y_train)
```

Out[388...
```
SVR()
```

In [389...
```
predict = model.predict(x_test)
```

In [390...
```
predict
```

Out[390...
```
array([0.79842413, 0.58785628, 0.75693053, 0.68694758, 0.68489116,
       0.75563508, 0.73214108, 0.86149761, 0.78459232, 0.67119331,
       0.66435528, 0.79417761, 0.70075444, 0.85759148, 0.62911953,
       0.53445218, 0.79430984, 0.61530444, 0.69219656, 0.67207249,
       0.69748132, 0.58970289, 0.76752116, 0.65780597, 0.56181875,
       0.72920365, 0.70372174, 0.90871333, 0.66506713, 0.88791939,
       0.65774145, 0.64619949, 0.59560372, 0.5780853 , 0.61990769,
       0.78735025, 0.69802988, 0.69777894, 0.81749576, 0.73918516,
       0.82216408, 0.77943644, 0.49541252, 0.59510754, 0.80947986,
       0.75177606, 0.74531938, 0.891966  , 0.68666727, 0.79361227,
       0.66097938, 0.71080809, 0.81733552, 0.68740492, 0.77152761,
       0.76125066, 0.57187683, 0.68360568, 0.67463896, 0.7780645 ,
       0.70242086, 0.6649329 , 0.66981752, 0.89036453, 0.74824798,
       0.57323302, 0.6616027 , 0.82884112, 0.66047946, 0.87770475,
       0.52071183, 0.63669407, 0.66188152, 0.67374324, 0.64457602,
       0.69714466, 0.69878669, 0.58159421, 0.62456121, 0.83061415,
```

```
       0.78165276, 0.87979063, 0.92095342, 0.62546014, 0.66732396,
       0.77094878, 0.73701187, 0.61819942, 0.90888935, 0.88000087,
       0.76475098, 0.66271068, 0.57706631, 0.59505779, 0.69385221,
       0.66701755, 0.56332064, 0.65567332, 0.6322232 , 0.72297643])
```

In [391…
```
from sklearn.metrics import r2_score

r2_score(y_test, predict)
```

Out[391…
```
0.7152184496819456
```

In [392…
```
model.score(x_test,y_test)
```

Out[392…
```
0.7152184496819456
```

In [393…
```
from sklearn.model_selection import GridSearchCV

param_grid={'C':[0.01 , 1], 'kernel':['rbf' , 'poly', 'linear'] }

grid= GridSearchCV(model,param_grid,cv =5)
```

In [394…
```
grid.fit(x_train,y_train)
```

Out[394…
```
GridSearchCV(cv=5, estimator=SVR(),
             param_grid={'C': [0.01, 1], 'kernel': ['rbf', 'poly', 'linear']})
```

In [395…
```
grid.best_score_
```

Out[395…
```
0.7765732659953086
```

In [396…
```
grid.best_params_
```

Out[396…
```
{'C': 1, 'kernel': 'linear'}
```