```
import pandas as pd
import numpy as np
```

```
data = pd.read_csv("Engineering_graduate_salary.csv")
```

```
data
```

	ID	Gender	DOB	10percentage	10board	12graduation	12percentage	1
0	604399	f	1990-10-22	87.80	cbse	2009	84.00	
1	988334	m	1990-05-15	57.00	cbse	2010	64.50	
2	301647	m	1989-08-21	77.33	maharashtra state board,pune	2007	85.17	a d
3	582313	m	1991-05-04	84.30	cbse	2009	86.00	
4	339001	f	1990-10-30	82.00	cbse	2008	75.00	
...	...	...	...	...	...	...	...	
2993	103174	f	1989-04-17	75.00	0	2005	73.00	
2994	352811	f	1991-07-22	84.00	state board	2008	77.00	
2995	287070	m	1988-11-24	91.40	bsemp	2006	65.56	
2996	317336	m	1988-08-25	88.64	karnataka education board	2006	65.16	ka e
2997	993701	m	1992-05-27	77.00	state board	2009	75.50	

2998 rows × 34 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2998 entries, 0 to 2997
Data columns (total 34 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   ID                     2998 non-null   int64
```

```
 1   Gender                 2998 non-null   object
 2   DOB                    2998 non-null   object
 3   10percentage           2998 non-null   float64
 4   10board                2998 non-null   object
 5   12graduation           2998 non-null   int64
 6   12percentage           2998 non-null   float64
 7   12board                2998 non-null   object
 8   CollegeID              2998 non-null   int64
 9   CollegeTier            2998 non-null   int64
 10  Degree                 2998 non-null   object
 11  Specialization         2998 non-null   object
 12  collegeGPA             2998 non-null   float64
 13  CollegeCityID          2998 non-null   int64
 14  CollegeCityTier        2998 non-null   int64
 15  CollegeState           2998 non-null   object
 16  GraduationYear         2998 non-null   int64
 17  English                2998 non-null   int64
 18  Logical                2998 non-null   int64
 19  Quant                  2998 non-null   int64
 20  Domain                 2998 non-null   float64
 21  ComputerProgramming    2998 non-null   int64
 22  ElectronicsAndSemicon  2998 non-null   int64
 23  ComputerScience        2998 non-null   int64
 24  MechanicalEngg         2998 non-null   int64
 25  ElectricalEngg         2998 non-null   int64
 26  TelecomEngg            2998 non-null   int64
 27  CivilEngg              2998 non-null   int64
 28  conscientiousness      2998 non-null   float64
 29  agreeableness          2998 non-null   float64
 30  extraversion           2998 non-null   float64
 31  nueroticism            2998 non-null   float64
 32  openess_to_experience  2998 non-null   float64
 33  Salary                 2998 non-null   int64
dtypes: float64(9), int64(18), object(7)
memory usage: 796.5+ KB
```

```
data.head()
```

	ID	Gender	DOB	10percentage	10board	12graduation	12percentage	12bo
0	604399	f	1990-10-22	87.80	cbse	2009	84.00	c
1	988334	m	1990-05-15	57.00	cbse	2010	64.50	c
2	301647	m	1989-08-21	77.33	maharashtra state board,pune	2007	85.17	amra divisi bc
3	582313	m	1991-05-04	84.30	cbse	2009	86.00	c
4	339001	f	1990-10-30	82.00	cbse	2008	75.00	c

5 rows × 34 columns

```
d=data.drop(['ID','10board','12board','Degree','CollegeState','DOB','12graduation','Gender
```

```
d.head()
```

	10percentage	12percentage	CollegeID	CollegeTier	collegeGPA	CollegeCityID	Co
0	87.80	84.00	6920	1	73.82	6920	
1	57.00	64.50	6624	2	65.00	6624	
2	77.33	85.17	9084	2	61.94	9084	
3	84.30	86.00	8195	1	80.40	8195	
4	82.00	75.00	4889	2	64.30	4889	

5 rows × 25 columns

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2998 entries, 0 to 2997
Data columns (total 25 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   10percentage           2998 non-null   float64
 1   12percentage           2998 non-null   float64
 2   CollegeID              2998 non-null   int64  
 3   CollegeTier            2998 non-null   int64  
 4   collegeGPA             2998 non-null   float64
 5   CollegeCityID          2998 non-null   int64  
 6   CollegeCityTier        2998 non-null   int64  
 7   GraduationYear         2998 non-null   int64  
 8   English                2998 non-null   int64  
 9   Logical                2998 non-null   int64  
 10  Quant                  2998 non-null   int64  
 11  Domain                 2998 non-null   float64
 12  ComputerProgramming    2998 non-null   int64  
 13  ElectronicsAndSemicon  2998 non-null   int64  
 14  ComputerScience        2998 non-null   int64  
 15  MechanicalEngg         2998 non-null   int64  
 16  ElectricalEngg         2998 non-null   int64  
 17  TelecomEngg            2998 non-null   int64  
 18  CivilEngg              2998 non-null   int64  
 19  conscientiousness      2998 non-null   float64
 20  agreeableness          2998 non-null   float64
 21  extraversion           2998 non-null   float64
 22  nueroticism            2998 non-null   float64
 23  openess_to_experience  2998 non-null   float64
 24  Salary                 2998 non-null   int64  
dtypes: float64(9), int64(16)
memory usage: 585.7 KB
```

```
X,Y=d.iloc[:,:-1],d.iloc[:,[-1]]
```

X

	10percentage	12percentage	CollegeID	CollegeTier	collegeGPA	CollegeCityID
0	87.80	84.00	6920	1	73.82	6920
1	57.00	64.50	6624	2	65.00	6624
2	77.33	85.17	9084	2	61.94	9084
3	84.30	86.00	8195	1	80.40	8195
4	82.00	75.00	4889	2	64.30	4889
...	...	...	...	...	...	...
2993	75.00	73.00	1263	2	70.00	1263
2994	84.00	77.00	9481	2	75.20	9481
2995	91.40	65.56	547	2	73.19	547
2996	88.64	65.16	1629	2	74.81	1629
2997	77.00	75.50	1111	2	69.30	1111

2998 rows × 24 columns

Y

	Salary

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
X=scaler.fit_transform(X)
```

2	255000

```
Y=Y/10000
```

4	200000

Double-click (or enter) to edit

```
from sklearn.model_selection import train_test_split
```

2994	120000

```
X_train,X_test,Y_train,Y_test= train_test_split(X,Y,test_size=1/4,random_state=0)
```

2996	530000

```
print(X_train.shape,X_test.shape,Y_train.shape,Y_test.shape)
```

```
(2248, 24) (750, 24) (2248, 1) (750, 1)
```

Double-click (or enter) to edit

```
Y_train.head()
```

	Salary
487	33.0
2820	25.0
1093	43.0
438	24.0
748	36.0

```
from sklearn.linear_model import LinearRegression

regressor=LinearRegression()

regressor.fit(X_train,Y_train)
```

```
LinearRegression()
```

```
y_pred.shape
```

```
(750, 1)
```

```
y_pred
```

```
array([[ 3.38660895e-03],
       [ 1.63434092e-03],
       [ 3.06641566e-03],
       [ 8.82524739e-04],
       [ 5.03246360e-03],
       [ 2.59535445e-03],
       [ 3.62880130e-03],
       [ 2.80408094e-03],
       [ 3.01973208e-03],
       [ 4.40253356e-03],
       [ 3.38848008e-03],
       [ 4.68101396e-03],
       [ 3.71957116e-03],
       [ 1.62604984e-03],
       [ 2.45631900e-03],
       [ 3.21095260e-03],
       [ 3.17886779e-03],
       [ 3.00002991e-03],
       [ 2.22359135e-03],
       [ 2.60966533e-03],
       [ 3.13928124e-03],
       [ 2.83306883e-03],
       [ 4.19182434e-03],
       [ 4.97112608e-03],
       [ 2.62268031e-03],
       [ 3.84087067e-03],
       [ 2.32565801e-03],
       [ 1.68550577e-03],
       [ 2.10079178e-03],
       [ 3.30297959e-03],
       [ 4.49853591e-03],
       [ 1.97993675e-03],
       [ 3.95819843e-03],
       [ 3.04621652e-03],
       [ 2.57195494e-03],
       [ 4.97858135e-03],
       [ 4.06063303e-03],
       [ 3.03281846e-03],
       [ 2.23685134e-03],
       [ 2.56726036e-03],
       [ 3.06749144e-03],
       [ 4.26900997e-03],
       [ 7.51625160e-01],
       [ 2.93336693e-03],
       [ 1.91409860e-03],
       [ 2.14098401e-03],
       [ 3.20624326e-03],
       [ 2.74087382e-03],
       [ 3.83693201e-03],
       [ 4.28316841e-03],
       [ 2.12103047e-03],
       [ 2.76773606e-03],
       [ 2.85177305e-03],
       [ 4.68733181e-03],
       [ 3.78334553e-03],
       [ 3.92424284e-03],
```

```
            [ 3.60090684e-03],
            [ 4.60044091e-03]
```

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

```
y_pred.shape
```

```
(750, 1)
```

```
y_pred
```

```
array([[ 3.38660895e+01],
       [ 1.63434092e+01],
       [ 3.06641566e+01],
       [ 8.82524739e+00],
       [ 5.03246360e+01],
       [ 2.59535445e+01],
       [ 3.62880130e+01],
       [ 2.80408094e+01],
       [ 3.01973208e+01],
       [ 4.40253356e+01],
       [ 3.38848008e+01],
       [ 4.68101396e+01],
       [ 3.71957116e+01],
       [ 1.62604984e+01],
       [ 2.45631900e+01],
       [ 3.21095260e+01],
       [ 3.17886779e+01],
       [ 3.00002991e+01],
       [ 2.22359135e+01],
       [ 2.60966533e+01],
       [ 3.13928124e+01],
       [ 2.83306883e+01],
       [ 4.19182434e+01],
       [ 4.97112608e+01],
       [ 2.62268031e+01],
       [ 3.84087067e+01],
       [ 2.32565801e+01],
       [ 1.68550577e+01],
       [ 2.10079178e+01],
       [ 3.30297959e+01],
       [ 4.49853591e+01],
       [ 1.97993675e+01],
       [ 3.95819843e+01],
       [ 3.04621652e+01],
       [ 2.57195494e+01],
       [ 4.97858135e+01],
       [ 4.06063303e+01],
       [ 3.03281846e+01],
       [ 2.23685134e+01],
       [ 2.56726036e+01],
       [ 3.06749144e+01],
       [ 4.26900997e+01],
       [ 7.51625160e+03],
       [ 2.93336693e+01],
       [ 1.91409860e+01],
       [ 2.14098401e+01],
       [ 3.20624326e+01],
```

```
[ 2.74087382e+01],
[ 3.83693201e+01],
[ 4.28316841e+01],
[ 2.12103047e+01],
[ 2.76773606e+01],
[ 2.85177305e+01],
[ 4.68733181e+01],
[ 3.78334553e+01],
[ 3.92424284e+01],
[ 3.60090684e+01],
[ 4.60044091e+01],
```

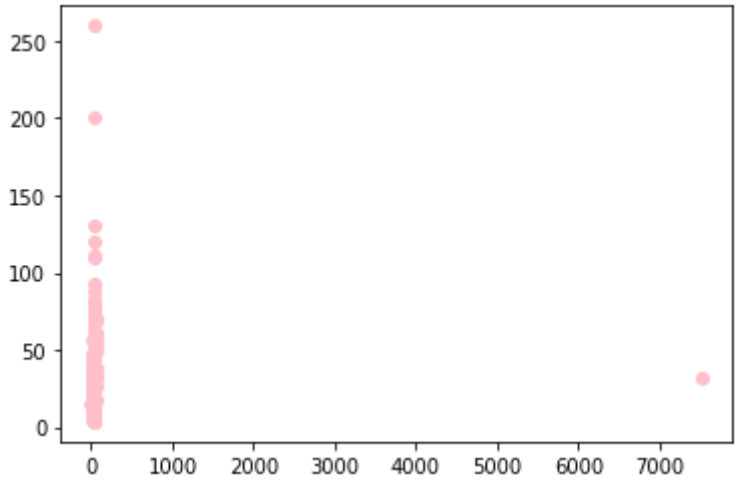
```
import matplotlib.pyplot as plt
```

```
print(y_pred.shape,Y_train.shape)
```

```
(750, 1) (2248, 1)
```

```
plt.scatter(y_pred,Y_test,color='pink')
```

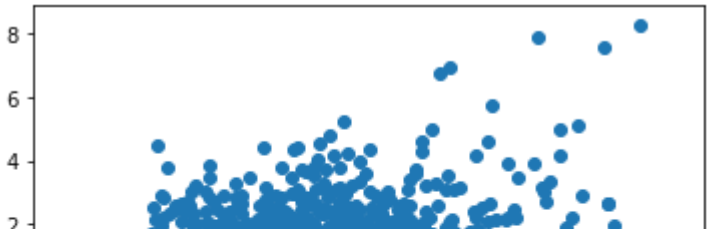
```
<matplotlib.collections.PathCollection at 0x7f7544017d50>
```



Double-click (or enter) to edit

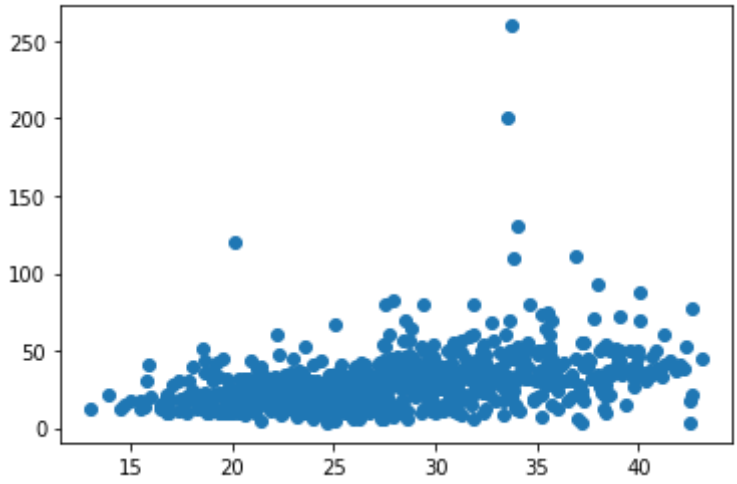
```
from sklearn.decomposition import PCA
pca=PCA(n_components=2)
X_pca=pca.fit_transform(X_train)
plt.scatter(X_pca[:,0],X_pca[:,1])
```

```
<matplotlib.collections.PathCollection at 0x7f7541e36e10>
```




```
from sklearn.svm import SVR
model=SVR()
model.fit(X_train,Y_train)
y_pred=model.predict(X_test)
plt.scatter(y_pred,Y_test)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993: DataConversic
  y = column_or_1d(y, warn=True)
<matplotlib.collections.PathCollection at 0x7f7541e430d0>
```




```
score=model.score(X_train,Y_train)
score
```

```
0.1084330628881951
```

```
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(X_train,Y_train)
y_pred=rf.predict(X_test)
score=rf.score(X_train,Y_train)
score
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: DataConversionWarning
  This is separate from the ipykernel package so we can avoid doing imports until
0.8622259002906525
```

```
plt.scatter(y_pred,Y_test)
```

```
<matplotlib.collections.PathCollection at 0x7f754123db50>
```