In [ ]:

```
# pip install scikit-learn
```

```
Collecting scikit-learn
  Downloading scikit_learn-1.0.2-cp310-cp310-win_amd64.whl (7.2 MB)
Requirement already satisfied: scipy>=1.1.0 in c:\users\javeria\appdata\local\programs\p
ython\python310\lib\site-packages (from scikit-learn) (1.7.3)
Collecting joblib>=0.11
  Downloading joblib-1.1.0-py2.py3-none-any.whl (306 kB)
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-3.0.0-py3-none-any.whl (14 kB)
Requirement already satisfied: numpy>=1.14.6 in c:\users\javeria\appdata\local\programs
\python\python310\lib\site-packages (from scikit-learn) (1.22.0)
Installing collected packages: threadpoolctl, joblib, scikit-learn
Successfully installed joblib-1.1.0 scikit-learn-1.0.2 threadpoolctl-3.0.0
Note: you may need to restart the kernel to use updated packages.

WARNING: You are using pip version 21.2.4; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\Javeria\AppData\Local\Programs\Python\Py
thon310\python.exe -m pip install --upgrade pip' command.
```

1- Simple linear regression

step 1 import data set as df

In [ ]:

```
import pandas as pd
df= pd.read_csv("ml_data.csv")
df.head()
```

Out[ ]:

	age	height	weight	gender	likeness
0	27	170.688	76.0	Male	Biryani
1	41	165.000	70.0	Male	Biryani
2	29	171.000	80.0	Male	Biryani
3	27	173.000	102.0	Male	Biryani
4	29	164.000	67.0	Male	Biryani

step 2 splitting of dataset into training data (80%) and testing data (20%)

In [ ]:

```
X = df[["age"]]
y = df[["height"]]
```

In [ ]:

```
X.head()
```

Out[ ]:

	age
0	27
1	41

	age
2	29
3	27
4	29

In [ ]:

```
y.head()
```

Out[ ]:

	height
0	170.688
1	165.000
2	171.000
3	173.000
4	164.000

In [ ]:

```
# import library and split data
from sklearn.model_selection import train_test_split
X_test, X_train, y_test, y_train = train_test_split(X, y, test_size= 0.2, random_state=
```

step 3 fit linear regression model

In [ ]:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression().fit(X_train, y_train)
model
```

Out[ ]:

```
LinearRegression()
```

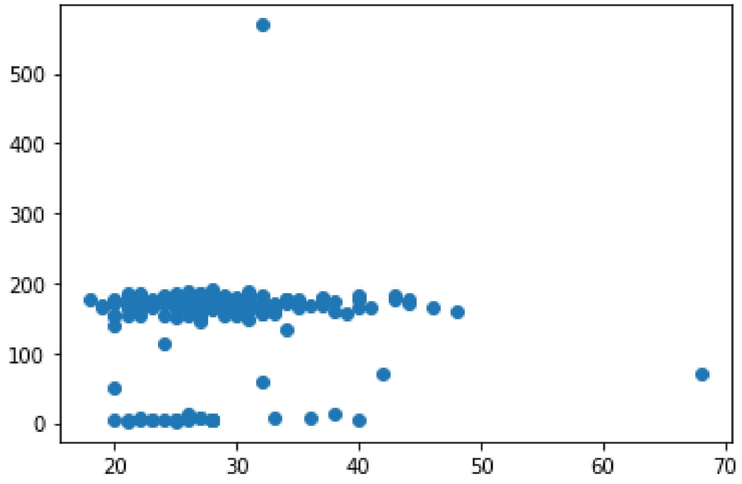
step 4 Plotting

In [ ]:

```
import matplotlib.pyplot as plt
plt.scatter(X_test, y_test)
```

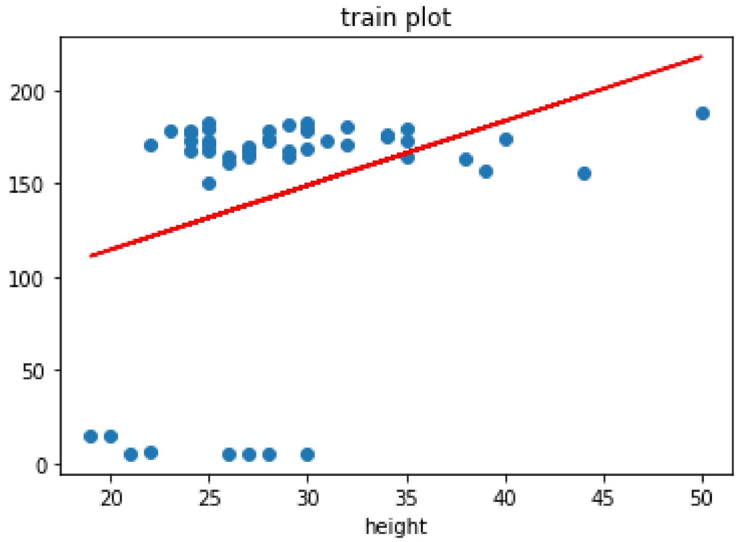
Out[ ]:

```
<matplotlib.collections.PathCollection at 0x26b7abde080>
```



In [ ]:

```
plt.scatter(X_train, y_train)
plt.plot(X_train, model.predict(X_train), color = "red")
plt.xlabel("age")
plt.xlabel("height")
plt.title("train plot")
plt.show()
```



step 5 Testing or evaluating model

In [ ]:

```
# model fitness
model.score(X_train, y_train)
```

Out[ ]:

```
0.11580049821224159
```

In [ ]:

```
model.score(X_test, y_test)
```

Out[ ]:

```
-0.08026580589459331
```

In [ ]:

```
print('score for training data= ', model.score(X_train, y_train))
print('score for test data= ', model.score(X_test, y_test))
```

```
score for training data=  0.11580049821224159
score for test data=  -0.08026580589459331
```

step 6 Predicting unknown values

In [ ]:

```
model.predict([[41]])
```

```
C:\Users\Javeria\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.
py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitte
d with feature names
  warnings.warn(
```

Out[ ]:

```
array([[186.98989693]])
```

In [ ]:

```
model.predict([[27], [41], [29]])
```

```
C:\Users\Javeria\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.
py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitte
d with feature names
  warnings.warn(
```

Out[ ]:

```
array([[138.69448635],
       [186.98989693],
       [145.59383072]])
```

In [ ]:

```
X= ([27], [29], [41])
model.predict(X)
```

```
C:\Users\Javeria\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.
py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitte
d with feature names
  warnings.warn(
```

Out[ ]:

```
array([[138.69448635],
       [145.59383072],
       [186.98989693]])
```