In [7]:

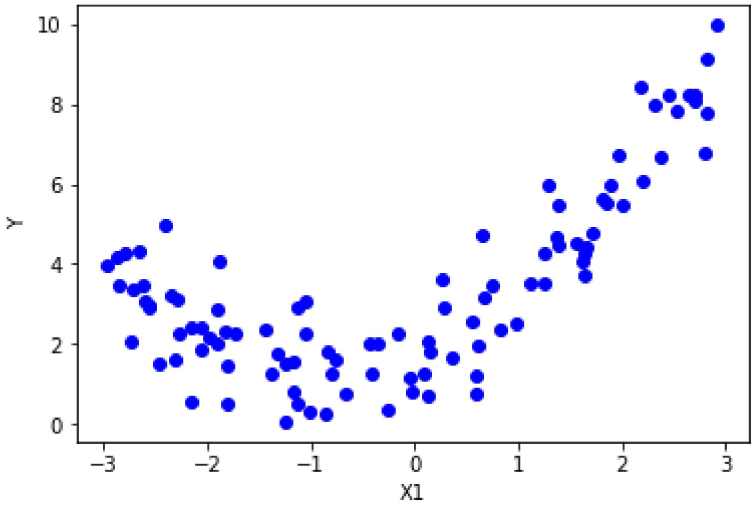
```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
playtennis=pd.read_csv(r'C:\Users\y22acm475\Desktop\playtennis.csv')
print("given dataset is:\n", playtennis, "\n")
le=LabelEncoder() # convert non-numerical value into numerical values
playtennis['outlook'] = le.fit_transform(playtennis['outlook'])
playtennis['Temperature'] = le.fit_transform(playtennis['Temperature'])
playtennis['Humidity'] = le.fit_transform(playtennis['Humidity'])
playtennis['Wind'] = le.fit_transform(playtennis['Wind'])
playtennis['PlayTennis'] = le.fit_transform(playtennis['PlayTennis'])
print("the Encoding dataset is \n", playtennis)
x = playtennis.drop(['PlayTennis'], axis=1)
y = playtennis['PlayTennis']
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.20)
print("\n x_train: \n", x_train)
print("\n y_train: \n", y_train)
model=GaussianNB()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
print("\n Accuracy of Naive Bayes classifier: ", accuracy)
```

```
given dataset is:
      outlook Temperature Humidity    Wind PlayTennis
0       Sunny         Hot     High    Weak         No
1       Sunny         Hot     High  Strong         No
2    Overcast         Hot     High    Weak        Yes
3        Rain        Mild     High    Weak        Yes
4        Rain        Cool   Normal    Weak        Yes
5        Rain        Cool   Normal  Strong         No
6    Overcast        Cool   Normal  Strong        Yes
7       Sunny        Mild     High    Weak         No
8       Sunny        Cool   Normal    Weak        Yes
9        Rain        Mild   Normal    Weak        Yes
10      Sunny        Mild   Normal  Strong        Yes
11   Overcast        Mild     High  Strong        Yes
12   Overcast         Hot   Normal    Weak        Yes
13       Rain        Mild     High  Strong         No

the Encoding dataset is
     outlook  Temperature  Humidity  Wind  PlayTennis
0         2            1         0     1           0
1         2            1         0     0           0
2         0            1         0     1           1
3         1            2         0     1           1
4         1            0         1     1           1
5         1            0         1     0           0
6         0            0         1     0           1
7         2            2         0     1           0
8         2            0         1     1           1
9         1            2         1     1           1
10        2            2         1     0           1
11        0            2         0     0           1
12        0            1         1     1           1
13        1            2         0     0           0

 x_train:
     outlook  Temperature  Humidity  Wind
11        0            2         0     0
13        1            2         0     0
10        2            2         1     0
5         1            0         1     0
0         2            1         0     1
6         0            0         1     0
3         1            2         0     1
2         0            1         0     1
9         1            2         1     1
4         1            0         1     1
1         2            1         0     0

 y_train:
 11    1
13    0
10    1
5     0
0     0
6     1
3     1
2     1
9     1
4     1
1     0
Name: PlayTennis, dtype: int64
```

```
Accuracy of Naive Bayes classifier:  1.0
```

In [22]:

```python
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(42)
m=100
X=6*np.random.rand(m,1)-3
y=0.5*X**2+X+2+np.random.randn(m,1)
plt.plot(X,y,"bo")
plt.xlabel("X1")
plt.ylabel("Y")
plt.show()
```



In [29]:

```python
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

from sklearn.preprocessing import PolynomialFeatures
poly_features=PolynomialFeatures(degree=2,include_bias=False)
X_poly=poly_features.fit_transform(X)
X[0]
```

Out[29]:

```
array([-0.75275929])
```

In [26]:

```python
X_poly[0]
```

Out[26]:

```
array([-0.75275929,  0.56664654])
```

In [27]:

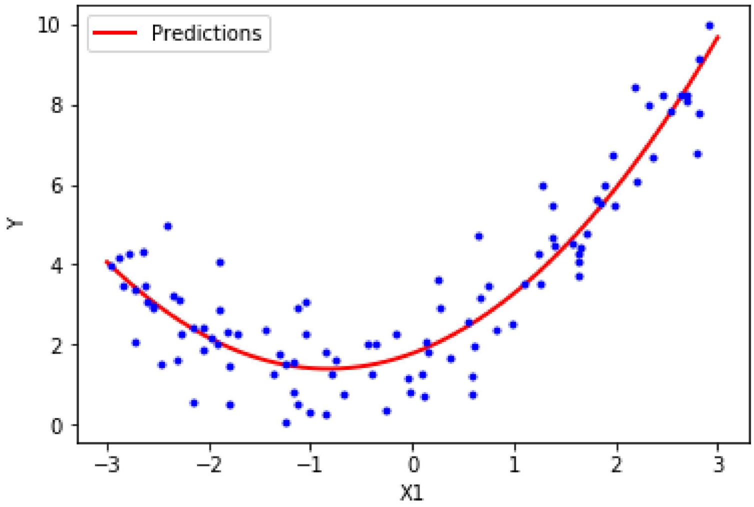
```python
from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
lin_reg.fit(X_poly,y)
lin_reg.intercept_,lin_reg.coef_
```

Out[27]:

```
(array([1.78134581]), array([[0.93366893, 0.56456263]]))
```

In [28]:

```
X_new=np.linspace(-3,3,200).reshape(200,1)
X_new_poly=poly_features.transform(X_new)
y_new=lin_reg.predict(X_new_poly)
plt.plot(X_new,y_new,"r-",linewidth=2,label="Predictions")
plt.plot(X_train,y_train,"b.")
plt.plot(X_test,y_test,"b.")
plt.xlabel("X1")
plt.ylabel("Y")
plt.legend()
plt.show()
```



In [ ]:

```
```