In [64]:

```
import numpy as np
import matplotlib.pyplot as ais
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures
from sklearn import linear_model
```

In [65]:

```
x=[0.54569285, 0.66713963, 1.8821619, 2.62643799, 2.77755533, 0.62036703, 
     1.46367319, 2.89139908, 1.28584076, 0.91031887, 1.58378549, 2.70337431
     1.84942398, 1.02288247, 1.32818655, 0.77015435, 1.32683635, 2.1469977
     1.45684425, 0.93492731, 2.17288046, 2.38366803, 0.18553194, 1.1605587
     2.05537434, 0.73697772, 1.67134223, 1.27495446, 2.05094272, 2.7166306
     2.30635976, 2.34910576, 2.52596201, 1.81093635, 0.61504781, 2.2543239
     2.28977926, 1.59435303, 0.16220664, 2.74076023, 2.13411313, 0.2488640
     0.8626656, 2.57910037, 1.4577751, 0.69469835, 1.1755677, 2.30071142, 
     1.63616482, 2.07380807, 0.27889959, 1.26687733, 0.19751214, 1.7340717
     1.02458033, 0.33707579, 2.34180852, 0.4797847,  1.75803667, 2.3355744
     0.93783358, 1.73717707, 0.46471257, 0.64567871, 0.72231835, 0.2588409

y=[0.53514393, 0.58600211, 0.93310423, 0.52689839, 0.40936582, 0.50996668,
     1.04168546, 0.09719999, 1.09654248, 0.85966593, 1.24210298, 0.1496748
     0.66311695, 0.81421226, 0.91617078, 0.8182873, 1.10528536, 0.82232858
     0.91921348, 0.88720014, 0.68272353, 0.78465141, 0.18418327, 0.8996447
     0.86465198, 0.48713479, 1.01252095, 0.9007005, 0.86324039, 0.51321692
     0.80306315, 0.52843247, 0.56275205, 0.9746029, 0.5096404, 0.91763387,
     0.74395972, 0.97298009, 0.24682351, 0.34909826, 0.58169838, 0.2156589
     0.79015231, 0.67503296, 1.25182691, 0.76448087, 1.18548521, 0.6790966
     0.95385419, 0.909671, 0.31834095, 1.01003781, 0.21067932, 0.85834788,
     0.92155107, 0.31528422, 0.77240702, 0.62354821, 0.89624496, 0.7095987
     0.99032025, 1.08341817, 0.53103793, 0.58917849, 0.67510364, 0.1681668
```

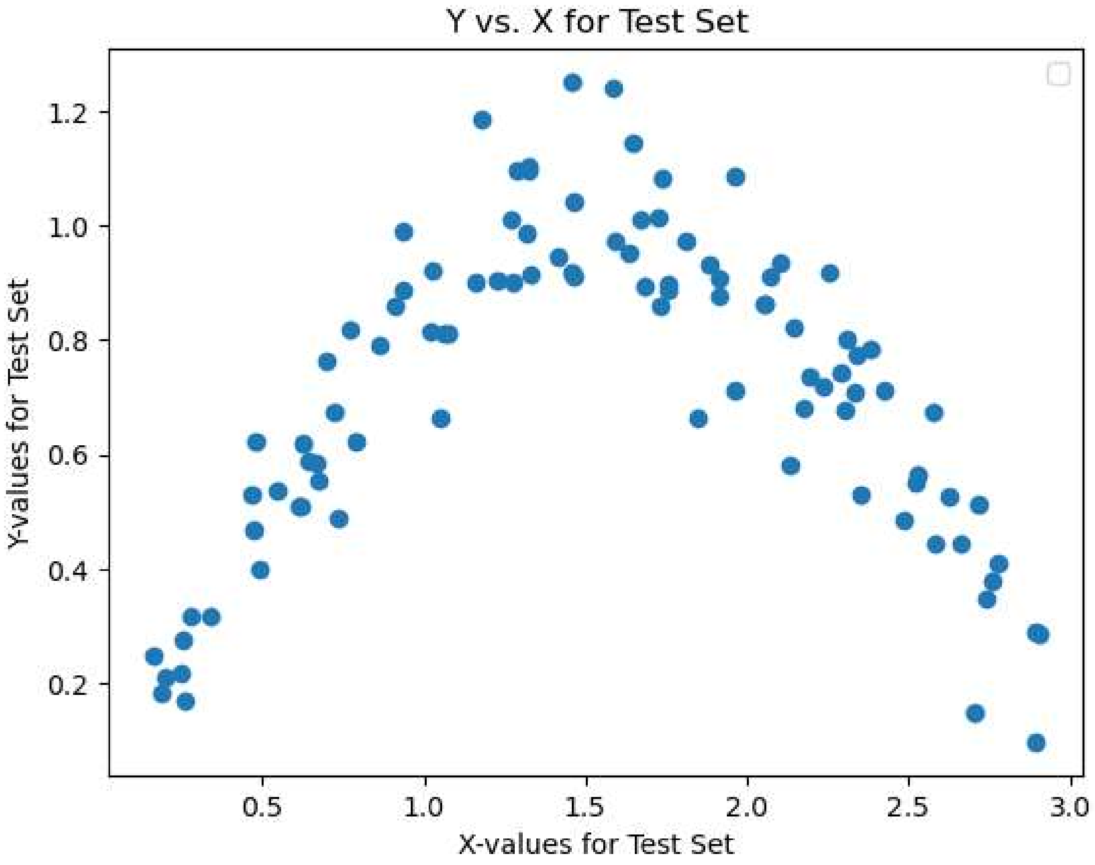
In [66]:

```
a_x=np.array(x)
a_y=np.array(y)
```

In [72]:

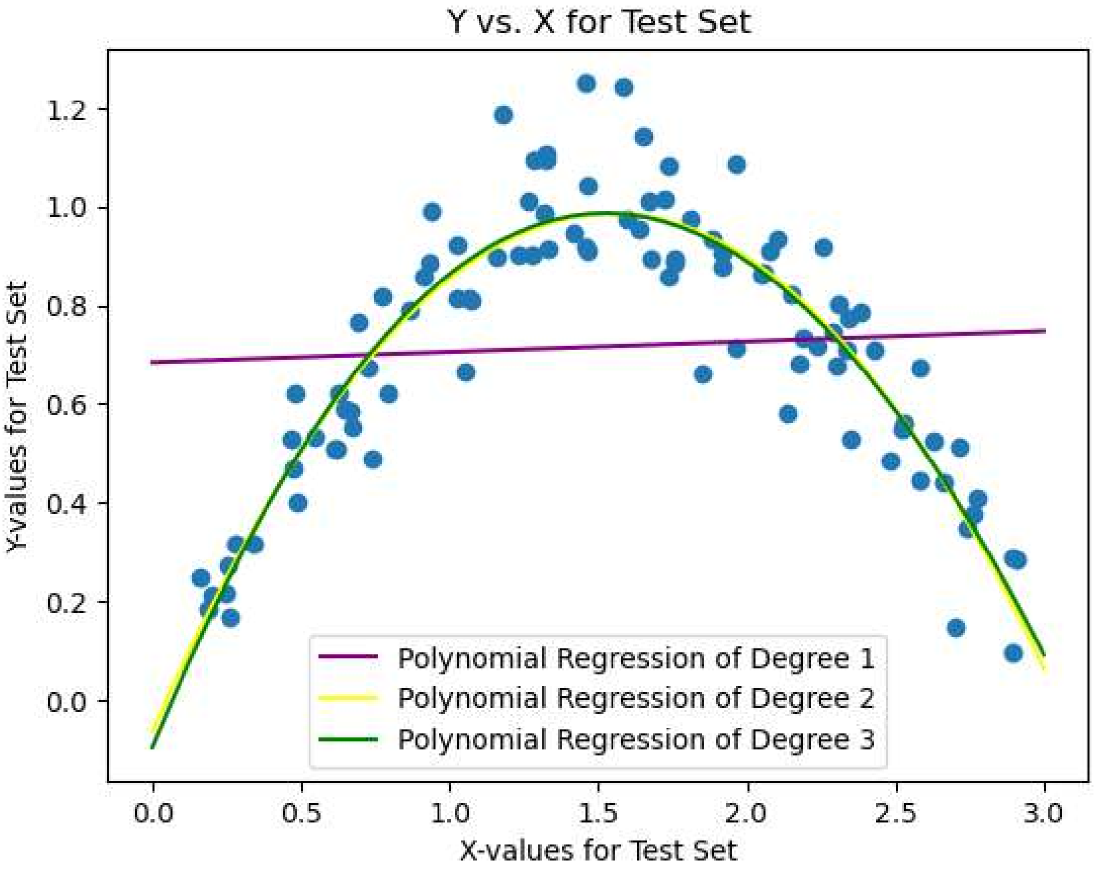
```
xn=np.linspace(0,3)
pr1 = Pipeline([('poly', PolynomialFeatures(degree=1)),('linear', linear_
pr1 =pr1.fit(a_x[:, np.newaxis], a_y[:, np.newaxis])
y1 = pr1.predict(xn[:, np.newaxis])

pr2 = Pipeline([('poly', PolynomialFeatures(degree=2)),('quadratic', linea
pr2 =pr2.fit(a_x[:, np.newaxis], a_y[:, np.newaxis])
y2 = pr2.predict(xn[:, np.newaxis])

pr3 = Pipeline([('poly', PolynomialFeatures(degree=3)),('cubic', linear_m
pr3 =pr3.fit(a_x[:, np.newaxis], a_y[:, np.newaxis])
y3 = pr3.predict(xn[:, np.newaxis])

ais.scatter(x,y)
ais.xlabel("X-values for Test Set")
ais.ylabel("Y-values for Test Set")
ais.title("Y vs. X for Test Set")

ais.legend()
ais.show()
```

```
No artists with labels found to put in legend.  Note that artists whose
label start with an underscore are ignored when legend() is called with
no argument.
```



In [70]:

```
ais.scatter(x,y)
ais.plot(xn,y1,'purple',label="Polynomial Regression of Degree 1")
ais.plot(xn,y2,'yellow',label="Polynomial Regression of Degree 2")
ais.plot(xn,y3,'green',label="Polynomial Regression of Degree 3")
ais.legend()
ais.xlabel("X-values for Test Set")
ais.ylabel("Y-values for Test Set")
ais.title("Y vs. X for Test Set")
ais.show()
```



In [ ]:

In [ ]: