

AndrewBreslauer_A5_PolynomialRegression

October 4, 2019

```
[1]: %matplotlib inline

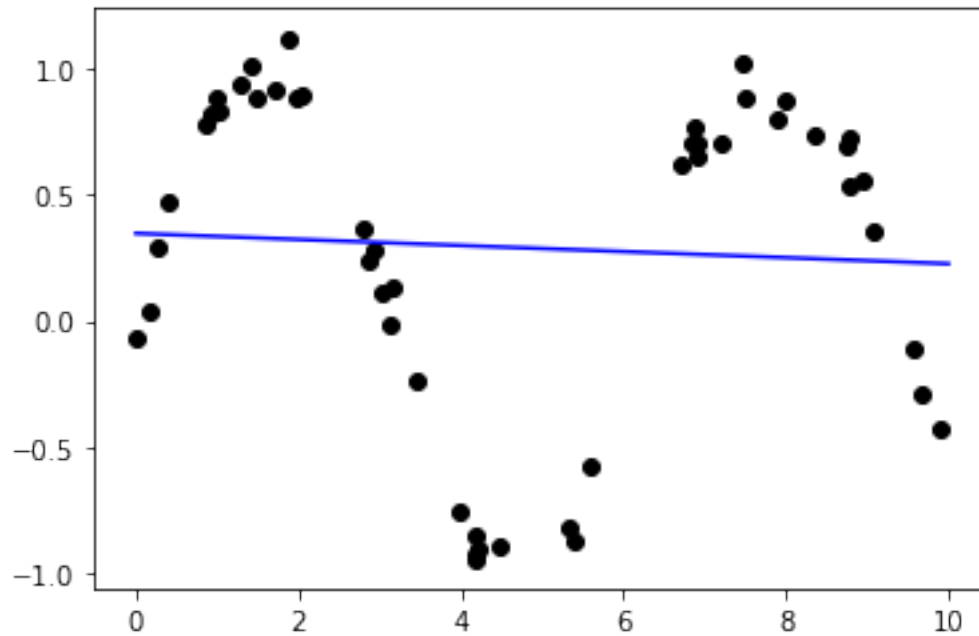
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LinearRegression, Ridge, Lasso
import matplotlib.pyplot as plt
import numpy as np

poly_model = make_pipeline(PolynomialFeatures(1), LinearRegression())
rng = np.random.RandomState(1)
x = 10 * rng.rand(50)
y = np.sin(x) + 0.1*rng.randn(50)

poly_model.fit(x[:,np.newaxis],y)
xfit = np.linspace(0, 10, 1000)
yfit = poly_model.predict(xfit[:,np.newaxis])

plt.scatter(x,y,color='black')
plt.plot(xfit,yfit,color='blue')
plt.show()

y_ground = np.sin(xfit)
print("1 Feature MSE =",mean_squared_error(yfit,y_ground))
```



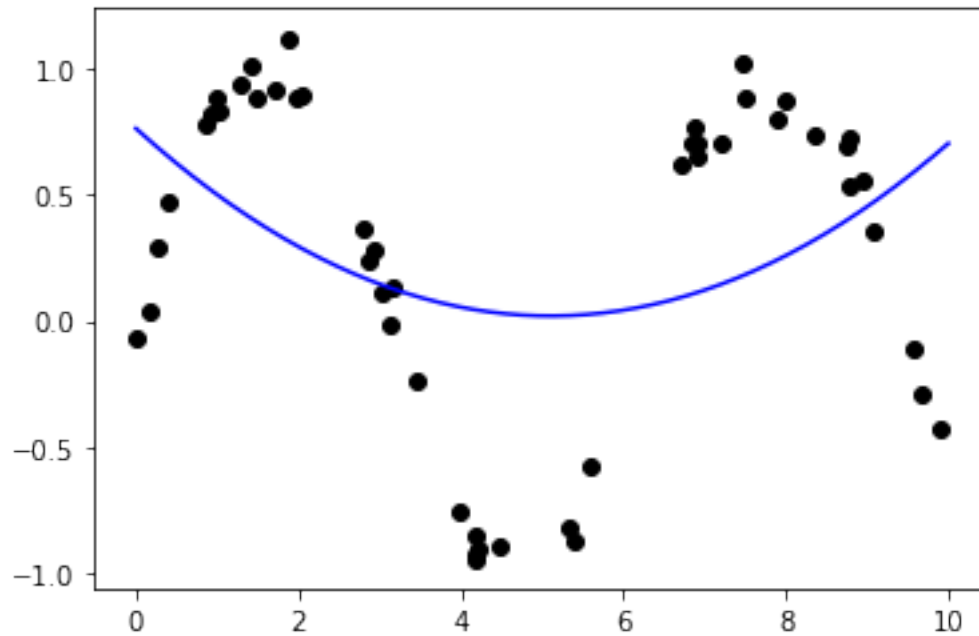
1 Feature MSE = 0.4520070794410817

```
[2]: poly_model = make_pipeline(PolynomialFeatures(2), LinearRegression())

poly_model.fit(x[:,np.newaxis],y)
xfit = np.linspace(0, 10, 1000)
yfit = poly_model.predict(xfit[:,np.newaxis])

plt.scatter(x,y,color='black')
plt.plot(xfit,yfit,color='blue')
plt.show()

y_ground = np.sin(xfit)
print("2 Feature MSE =",mean_squared_error(yfit,y_ground))
```



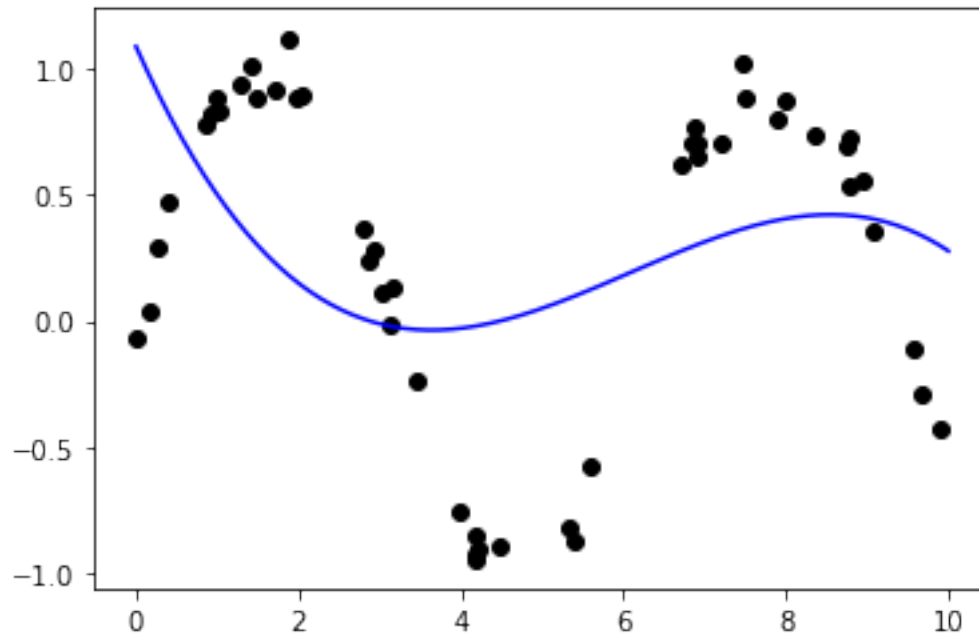
2 Feature MSE = 0.37063661531424125

```
[3]: poly_model = make_pipeline(PolynomialFeatures(3), LinearRegression())

poly_model.fit(x[:,np.newaxis],y)
xfit = np.linspace(0, 10, 1000)
yfit = poly_model.predict(xfit[:,np.newaxis])

plt.scatter(x,y,color='black')
plt.plot(xfit,yfit,color='blue')
plt.show()

y_ground = np.sin(xfit)
print("3 Feature MSE =",mean_squared_error(yfit,y_ground))
```



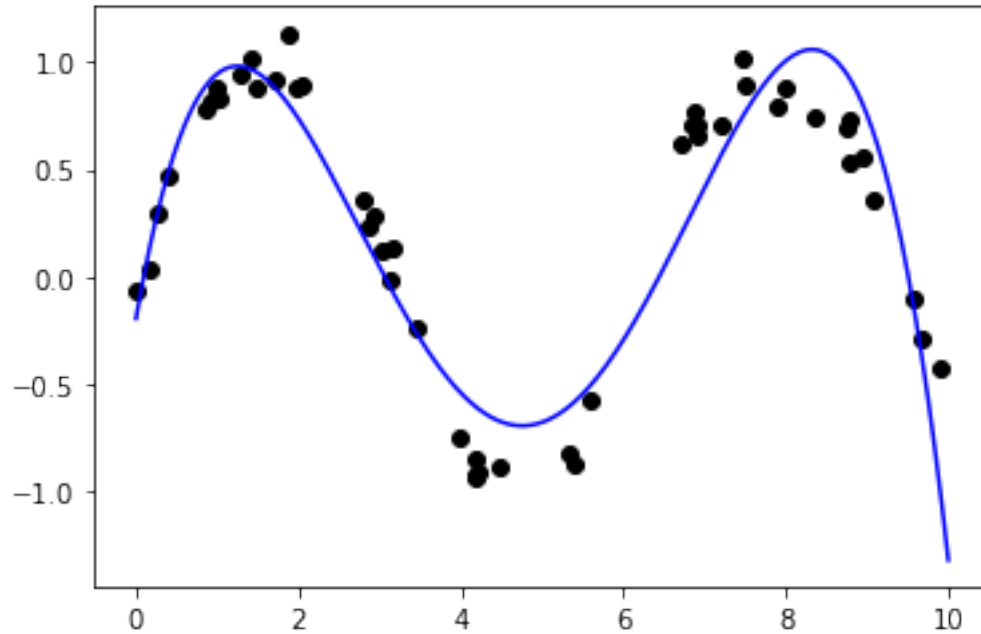
3 Feature MSE = 0.3465879905824595

```
[4]: poly_model = make_pipeline(PolynomialFeatures(4), LinearRegression())

poly_model.fit(x[:,np.newaxis],y)
xfit = np.linspace(0, 10, 1000)
yfit = poly_model.predict(xfit[:,np.newaxis])

plt.scatter(x,y,color='black')
plt.plot(xfit,yfit,color='blue')
plt.show()

y_ground = np.sin(xfit)
print("4 Feature MSE =",mean_squared_error(yfit,y_ground))
```



4 Feature MSE = 0.040276698745374546

Once we achieve 4+ polynomial features, the graph remains the same.

As we can see above, 4 polynomial features/variables fits best to the randomly generated values. Despite the possibility of overfitting with the 4 features, the mean squared error difference is just too large to prevent picking the 4-feature model over the 3-feature model, so you have to pick the 4-feature model.