

```
In [1]: %matplotlib inline
import operator
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_validate
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
```

```
In [2]: df = pd.read_csv("dataset.csv")
```

```
In [3]: x = df[['x']].values
y = df.y.values
```

```
In [4]: x_train, x_val, y_train, y_val = train_test_split(x,y, train_size=.75, random_state=1)
```

```
In [5]: maxdeg = 20

training_error, validation_error, cross_validation_error = [],[],[]

for d in range(maxdeg):

    x_poly = PolynomialFeatures(degree=d).fit_transform(x)

    x_poly_train = PolynomialFeatures(degree=d).fit_transform(x_train)

    x_poly_val = PolynomialFeatures(degree=d).fit_transform(x_val)

    lreg = LinearRegression(fit_intercept=False )

    lreg.fit(x_poly_train, y_train)

    y_train_pred = lreg.predict(x_poly_train)

    y_val_pred = lreg.predict(x_poly_val)

    training_error.append(mean_squared_error(y_train, y_train_pred))

    validation_error.append(mean_squared_error(y_val, y_val_pred))

    mse_score = cross_validate(lreg, x_poly, y, cv=10, scoring= 'neg_mean_squared_error', return_train_score= T

    cross_validation_error.append(abs(np.mean(mse_score['test_score'])))
```

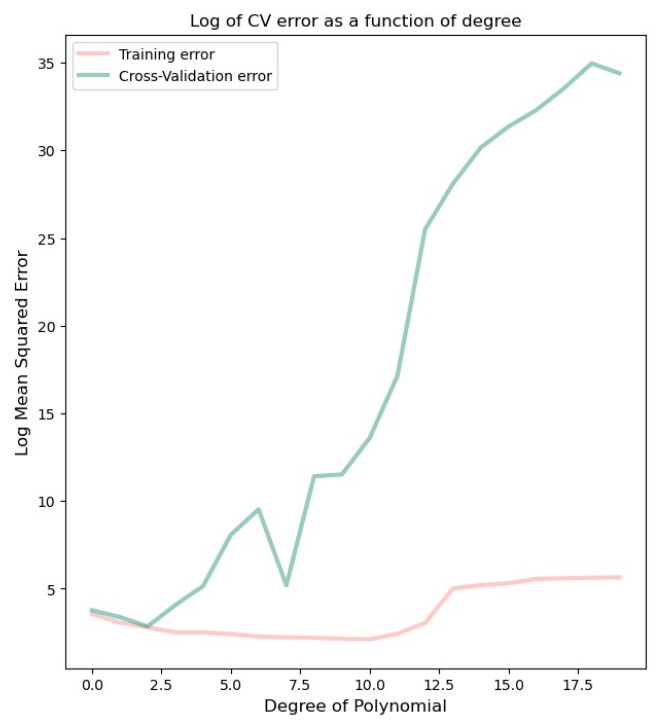
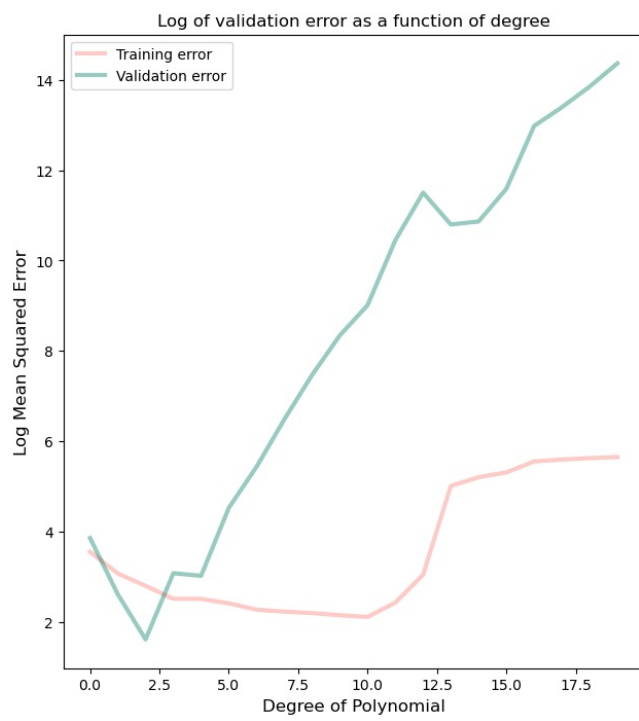
```
In [6]: fig, ax = plt.subplots(1,2, figsize=(16,8))

ax[0].plot(range(maxdeg), np.log(training_error), label = 'Training error', linewidth=3, color='#FF7E79', alpha=0.5)
ax[0].plot(range(maxdeg), np.log(validation_error), label = 'Validation error', linewidth=3, color="#007D66", alpha=0.5)
ax[1].plot(range(maxdeg), np.log(training_error), label = 'Training error', linewidth=3, color='#FF7E79', alpha=0.5)
ax[1].plot(range(maxdeg), np.log(cross_validation_error), label = 'Cross-Validation error', linewidth=3, color="#007D66", alpha=0.5)

ax[0].set_xlabel('Degree of Polynomial', fontsize=12)
ax[0].set_ylabel('Log Mean Squared Error', fontsize=12)
ax[0].set_title("Log of validation error as a function of degree")

ax[1].set_xlabel('Degree of Polynomial', fontsize=12)
ax[1].set_ylabel('Log Mean Squared Error', fontsize=12)
ax[1].set_title("Log of CV error as a function of degree")

ax[0].legend()
ax[1].legend()
plt.show();
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js