Q1. Read it the .xlsx file using Panda's read_excel() function

In [1]:

```
import pandas as pd

electricity = pd.read_excel('DataSet(Assignment3).xlsx')
```

Q2. Please set the three train sizes in your python programming

In [2]:

```
train_sizes = [100, 500, 2000]
```

Q3. Please use several models (including linear and nonlinear models) to specify the 'learning_curve()' function in Scikit-learn.

In [3]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import learning_curve

features = ['AT', 'V', 'AP', 'RH']
target = 'PE'
train_sizes, train_scores, validation_scores = learning_curve(
estimator = LinearRegression(),
X = electricity[features],
y = electricity[target], train_sizes = train_sizes, cv = 5,
scoring = 'neg_mean_squared_error')
```

Q4. Please calculate the error scores (training and validation) at the three train_size ={100, 500,2000}

In [4]:

```
print('Training scores:₩n₩n', train_scores)
print('₩n', '-' * 70)
print('₩nValidation scores:₩n₩n', validation_scores)
```

```
Training scores:

 [[-19.71230701 -18.31492642 -18.31492642 -18.31492642 -18.31492642]
 [-18.14420459 -19.63885072 -19.63885072 -19.63885072 -19.63885072]
 [-21.53603444 -20.18568787 -19.98317419 -19.98317419 -19.98317419]]

 ----------------------------------------------------------------------

Validation scores:

 [[-21.80224219 -23.01103419 -20.81350389 -22.88459236 -23.44955492]
 [-19.96005238 -21.2771561  -19.75136596 -21.4325615  -21.89067652]
 [-19.92863783 -21.35440062 -19.62974239 -21.38631648 -21.811031  ]]
```

Q5. Please calculate the mean error scores (training and validation) at the three train_size = {100, 500, 2000}

In [5]:

```
train_scores_mean = -train_scores.mean(axis = 1)
validation_scores_mean = -validation_scores.mean(axis = 1)

print('Mean training scores₩n₩n', pd.Series(train_scores_mean, index = train_sizes))
print('₩n', '-' * 20)
print('₩nMean validation scores₩n₩n',pd.Series(validation_scores_mean, index = train_sizes))
```

```
Mean training scores

 100      18.594403
500       19.339921
2000     20.334249
dtype: float64

 --------------------

Mean validation scores

 100      22.392186
500       20.862362
2000     20.822026
dtype: float64
```

Q6. Plot the learning curve using matplotlib, and explain what regression model and what train_size make the 'bias-variance' trade-off balanced off.

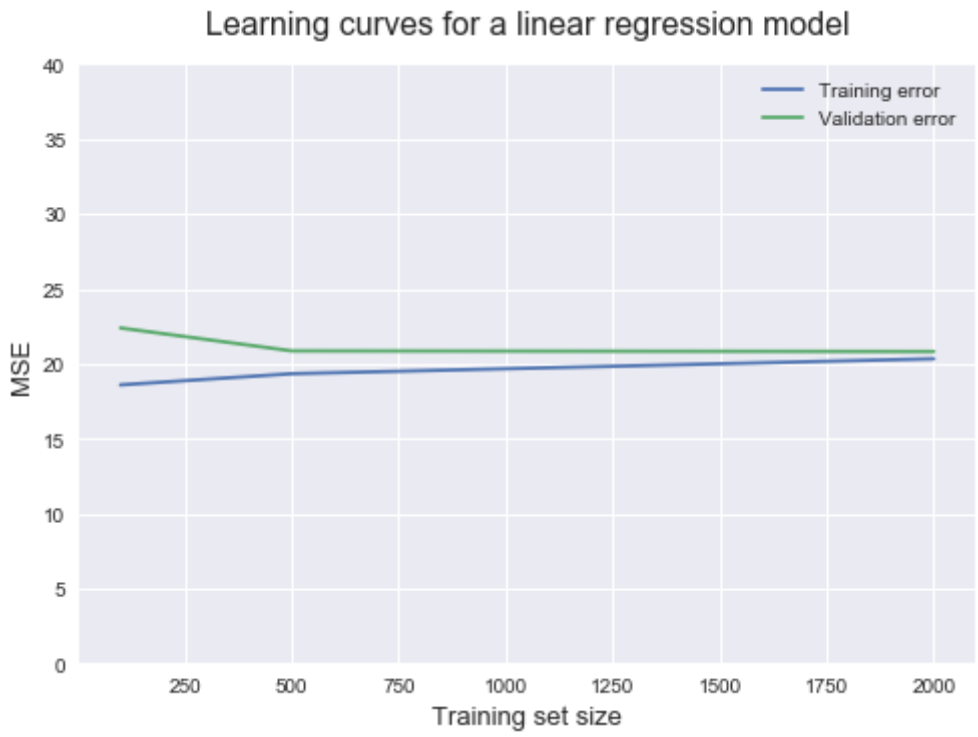
In [6]:

```
import matplotlib.pyplot as plt

plt.style.use('seaborn')
plt.plot(train_sizes, train_scores_mean, label = 'Training error')
plt.plot(train_sizes, validation_scores_mean, label = 'Validation error')
plt.ylabel('MSE', fontsize = 13)
plt.xlabel('Training set size', fontsize = 13)
plt.title('Learning curves for a linear regression model', fontsize = 16, y = 1.03)
plt.legend()
plt.ylim(0,40)
```

Out[6]:

(0, 40)



In [ ]:

```

```