

Lab 01 - Creating a Python Model

Lab 01 - Creating a Python Model

Welcome to the first lab. In this lab we are just going to build a basic machine learning model.

We are going to build a regression model. We will use an inbuilt dataset. In this session I am not trying to show you how to create machine learning models, that enough content for another week. We will do something basic and prove the point that we can deploy it with a few simple techniques.

In this lab we will build a model together and explore a little about the dataset.

Step 1

For this model we will be using Python.

We are going to incrementally build our model. You can choose to do this in your tool of choice. A jupyter notebook might a good idea.

Start by creating a folder called **ProductionDiabetes** create a new file call ProductionDiabetes.py

Step 2

We will need to import a few new modules.

If running the following results in an error, then run the scripts in the next section.

```
# Import the required libraries
import numpy as np
from sklearn import datasets, linear_model
```

We will use PIP to install the missing modules.

```
python -m pip install numpy==1.15.4
python -m pip install scikit-learn==0.19.1
```

Step 3

Load in the datasets from SKlearn. We will be using the diabetes dataset.

```
diabetes = datasets.load_diabetes()
```

We are going to use only one feature. To make the prediction simple.

```
diabetes_X = diabetes.data[:, np.newaxis, 2]
```

Lab 01 - Creating a Python Model

Split the data into training/testing sets

```
diabetes_X_train = diabetes_X[:-20]  
diabetes_X_test = diabetes_X[-20:]
```

Split the targets into training/testing sets

```
diabetes_y_train = diabetes.target[:-20]  
diabetes_y_test = diabetes.target[-20:]
```

Create linear regression object

```
regr = linear_model.LinearRegression()
```

Train the model using the training sets

```
regr.fit(diabetes_X_train, diabetes_y_train)
```

Make predictions using the testing set

```
diabetes_y_pred = regr.predict(diabetes_X_test)  
# Examples to show.  
print(diabetes_y_pred)  
NewValue = 12  
NewValuePrediction = regr.predict(NewValue)  
  
print(NewValuePrediction)
```

Completed script:

```
# Import the required libraries  
import numpy as np  
from sklearn import datasets, linear_model  
  
# Load in the datasets from SKlearn. We will be using the diabetes dataset.  
diabetes = datasets.load_diabetes()  
  
# Use only one feature. To make the prediction simple.  
diabetes_X = diabetes.data[:, np.newaxis, 2]  
  
# Split the data into training/testing sets  
diabetes_X_train = diabetes_X[:-20]  
diabetes_X_test = diabetes_X[-20:]  
  
# Split the targets into training/testing sets  
diabetes_y_train = diabetes.target[:-20]  
diabetes_y_test = diabetes.target[-20:]  
  
# Create linear regression object  
regr = linear_model.LinearRegression()  
  
# Train the model using the training sets  
regr.fit(diabetes_X_train, diabetes_y_train)  
  
# Make predictions using the testing set  
diabetes_y_pred = regr.predict(diabetes_X_test)  
  
# Examples to show.  
# print(diabetes_y_pred)  
  
NewValue = [[12]]  
NewValuePrediction = regr.predict(NewValue)
```

Lab 01 - Creating a Python Model

```
print(NewValuePrediction)
```