

Machine Learning On AWS Platform (SageMaker & Jupyter Notebook)

By Subham Singh
MCA (PES University)

Machine Learning On AWS (with categories)

- ML Service → **Build train** and **deploy** ML Fast
- AI Service → Easily **add Intelligence** to your applications
- Frameworks → **Choice and flexibility** with broadest framework support
- Compute → **Fastest and Lowest Cost Compute Option**
- Analytics and Security → **Comprehensive capabilities**, no compromise
- Learning Tools → Get deep on ML with **AWS DeepRacer and DeepLens**

What is Machine Learning ?

Machine Learning is the idea

where there are some **generic algorithm** that can tell you something interesting about a set of data without you having write any custom code for specific to the problem

Instead of writing code, you feed data to the **generic algorithm** and it builds its own logic based on the data.

(taken from [Blog](#) on Medium)

Machine Learning On AWS Platform

AWS provides a diverse platform for the machine learning and its computation :

Amazon Sagemaker Steps

Step 1: Create an Amazon S3 Bucket

Step 2: Create an Amazon SageMaker Notebook Instance

Step 3: Create a Jupyter Notebook

Step 4: Download, Explore, and Transform the Training Data

Step 5: Train a Model

Step 6: Deploy the Model to Amazon SageMaker

Amazon Sagemaker Steps

Step 7: Validate the Model

Step 8: Clean Up

Step 9: Integrating Amazon SageMaker Endpoints into Internet-facing Applications

Step 1: Create an Amazon S3 Bucket

Download two dataset from below link

1. <https://s3.amazonaws.com/aml-sample-data/banking.csv> (training data)
2. <https://s3.amazonaws.com/aml-sample-data/banking-batch.csv>. (testing data)

Open your dataset banking.csv (mostly open in Excel)

You can see the **header row** contains the attribute names for each column

Here attribute means → property to describe particular characteristics of each customer.(banking dataset)

upload the **banking.csv** and **banking-batch.csv** files to Amazon S3

Step 2: Create an Amazon SageMaker Notebook Instance

1. Open the Amazon SageMaker console at <https://console.aws.amazon.com/sagemaker/>.
2. Choose **Notebook instances**, then choose **Create notebook instance**.
3. On the **Create notebook instance** page, provide the following information (if a field is not mentioned, leave the default values):
 - a. For **Notebook instance name**, type a name for your notebook instance.
 - b. For **Instance type**, choose ml.t2.medium. This is the least expensive instance type that notebook instances support, and it suffices for this exercise.
 - c. For **IAM role**, choose **Create a new role**, then choose **Create role**.
 - d. Choose **Create notebook instance**.

Step 3: Create a Jupyter Notebook

Open the Notebook Instances,

and then open the notebook instance you created by choosing

either **Open Jupyter** for classic Jupyter view

or **Open JupyterLab** for JupyterLab view

next to the name of the notebook instance.

Step 3: Create a Jupyter Notebook

Click on **Open Jupyter** Under **Action Column**.

Jupyter Notebook windows will Open

If you opened the notebook in Jupyter classic view, on the **Files** tab, choose **New**, and **conda_python3**.

This preinstalled environment includes the default Anaconda installation and Python 3.



Following steps will be explained in Jupyter Notebook

Create (S3)
Hoteltips.csv



meal_amt	tips
34	5
108	17
64	11
88	8
153	14
51	5
100	15
205	10
55	5
60	5
75	10
90	10
150	20
200	10
70	5
80	10

Code #import libraries to execute

#import libraries to execute

Draw the graph	import matplotlib.pyplot as plt
Calculation	import numpy as np
Implementing Technique	from sklearn import linear_model
import dataset in notebook	import pandas as pd
Source	from sklearn.metrics
Evaluating the model	import mean_squared_error
Find SQRT of the Number	from math import sqrt

jupyter MLAWSPESU Last Checkpoint: 3 hours ago (autosaved)



File Edit View Insert Cell Kernel Widgets Help Not Connected Trusted conda_python3

Save Add Reload Close Copy Paste Undo Redo Run Stop Restart Code nbdiff

```
In [1]: #import libraries to execute
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model
import pandas as pd
from sklearn.metrics import mean_squared_error
from math import sqrt
```

#Reading CSV from S3 Bucket

- #Reading CSV from S3 Bucket

```
df = pd.read_csv('https://mysmpesubbucket.s3.ap-south-1.amazonaws.com/ Hoteltips.csv', sep=',')
```

```
df.head()
```

or

```
df.tail()
```

Out[9]:

	meal_amt	tips
11	90	10
12	150	20
13	200	10
14	70	5
15	80	10

#Selecting features as X and Y

- #Selecting features as X and Y

```
X= df['meal_amt']
```

```
Y = df['tips']
```

```
print("Datatype of X: ", type(X))
```

```
print("Datatype of Y: ", type(Y))
```

```
#Selecting and reshaping the X and Y
```

```
X = X.values
```

```
X = X.reshape(len(X),1)
```

```
Y = Y.values
```

```
Y = Y.reshape(len(Y),1)
```

```
Datatype of X: <class 'pandas.core.series.Series'>
```

```
Datatype of Y: <class 'pandas.core.series.Series'>
```

#Split the data into training/testing sets

- #Split the data into training/testing sets

```
X_train = X[:-4]
```

```
X_test = X[-4:]
```

Split the targets into training/testing sets

```
Y_train = Y[:-4]
```

```
Y_test = Y[-4:]
```

#plot train Data

- #plot train Data

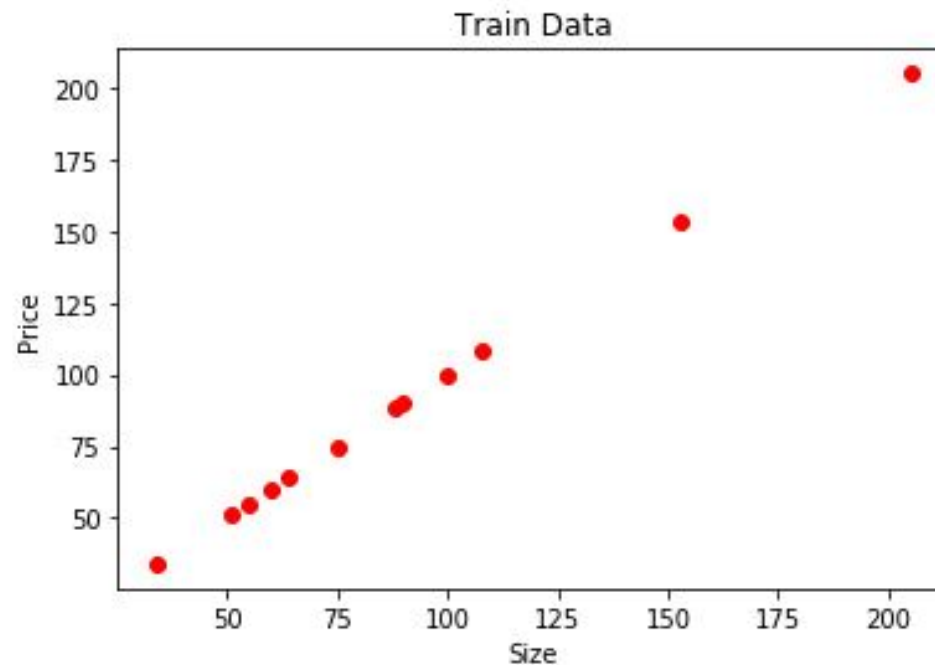
```
plt.scatter(X_train,Y_train, color='red')
```

```
plt.title('Train Data')
```

```
plt.xlabel('Size')
```

```
plt.ylabel('Price')
```

```
Text(0, 0.5, 'Price')
```



Create linear regression object

- # Create linear regression object

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

Train the model using the training sets

- # Train the model using the training sets

```
regr.fit(X_train, Y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,  
                 normalize=False)
```

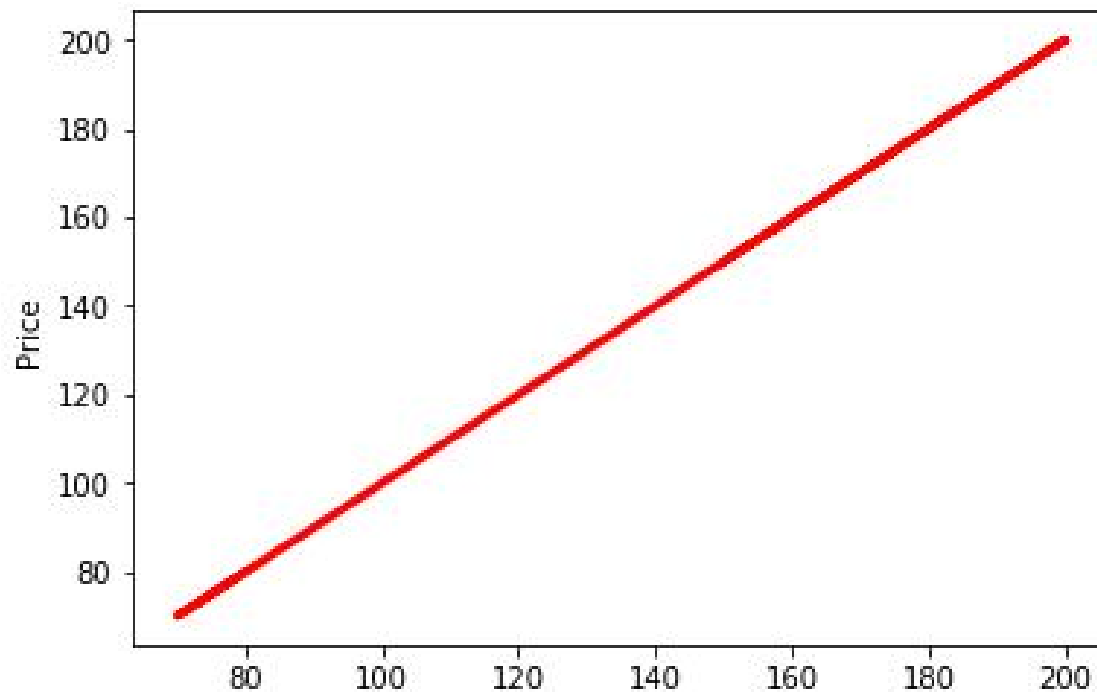
#Test the linear regression model on X_test Data

- #Test the linear regression model on X_test Data
`Y_pred=regr.predict(X_test)`

#Plot outputs->regression line is drawn

- #Plot outputs-> regression line is drawn
`plt.plot(X_test,Y_pred , color='red',linewidth=3)`
`plt.xlabel('Size')`
`plt.ylabel('Price')`

```
Text(0, 0.5, 'Price')
```



Root mean Squared Error

- `rms = sqrt(mean_squared_error(Y_test, Y_pred))`
`print("RMS : ",rms)`

RMS : 0.0

Reference

- <https://scikit-learn.org/>
- <https://docs.aws.amazon.com/sagemaker/index.html>
- <https://docs.aws.amazon.com/dlami/latest/devguide/setup-jupyter.html>

Thank You