

## SageMaker Studio Lab: Machine Learning

# Read and practice the examples below  
# submit a screenshot of your source code and output.

### a. Introduction/Overview:

What's Amazon SageMaker Studio Lab?

Amazon SageMaker Studio Lab is a free cloud service provided by AWS. It is a no-charge (no credit card need) and no-setup (no AWS account) notebook environment (Jupyter notebooks). It's designed for learning and experimenting with Data Science and Machine Learning. You can get started with sign-up for an account using your email address only.

In this lab, we will introduce a simple Machine Learning example using Python. How to build, train and deploy a Machine Learning program in a simple example. There are many Python libraries used for ML. Our example uses the Python libraries numpy, pandas, pickle-mixin and scikit-learn to train our model using the 'Linear Regression' algorithm. Our example will predict a salary based on the years of experience, using the sample training dataset SalaryData.csv.

# The ML Model – Linear Regression (create a model, train and use it).

Training Dataset -> [ LINEAR REGRESSION MODEL ] <- Test Dataset  
=> Prediction (Outcome)

- Sample Training Dataset (SalaryData):

YearsExperience	Salary
1.0	28000
2.0	31500
5.5	78950
10.9	112500

- **Split Dataset and Train the Model:** (Use the training dataset to train a Linear Regression Model)

```
year_train, year_test, salary_train, salary_test = train_test_split(year, salary,  
test_size = 1/2, random_state = 0)
```

```
regressor = LinearRegression()  
  
regressor.fit(year_train, salary_train)
```

- **Test/Evaluate the Model:** (Use the testing dataset to calculate the testing error)

```
testing_error = mean_squared_error(salary_test, salary_pred, squared = False)
```

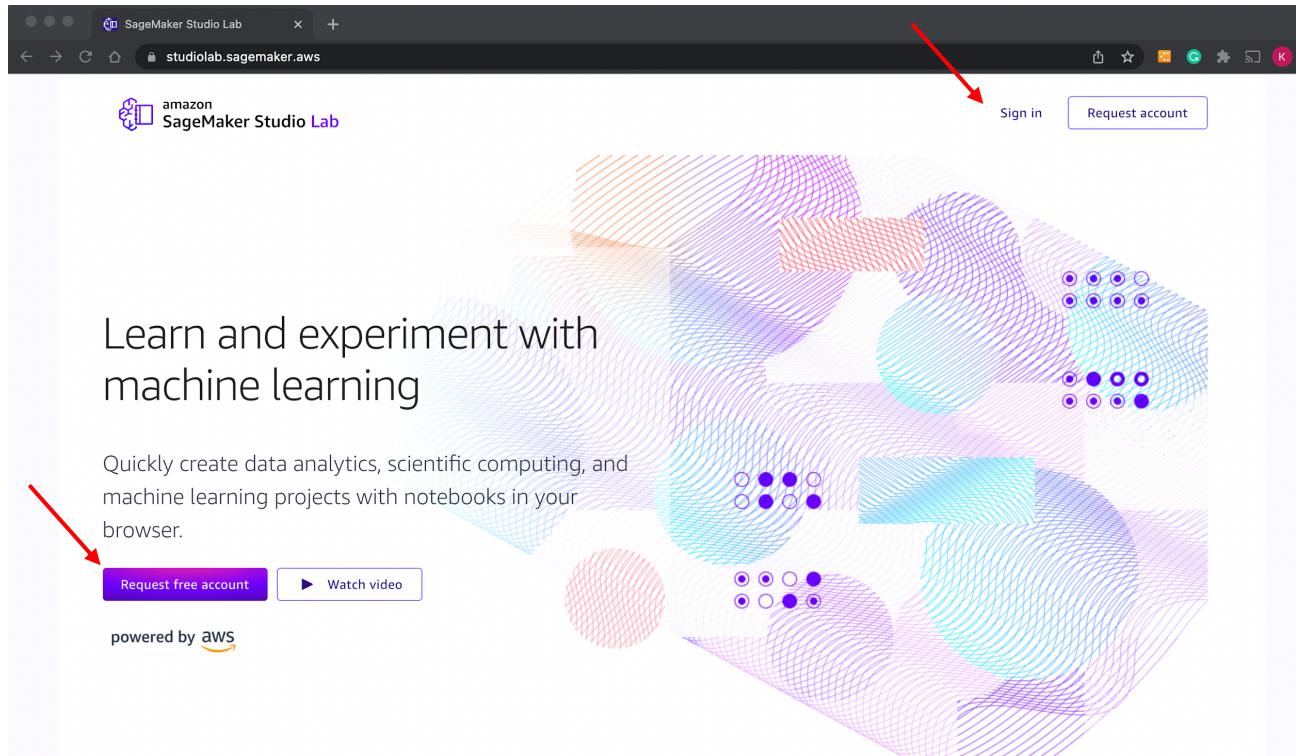
- **Outcome:** (Use model to predict the salary of someone with 20.8 years of experience)

```
mlmodel.predict([[20.8]]) # with 20.8 years experiences  
print("Predicated Salary: $ {:.2f}\n".format(prediction[0]))
```

## b. Workshop/practice:

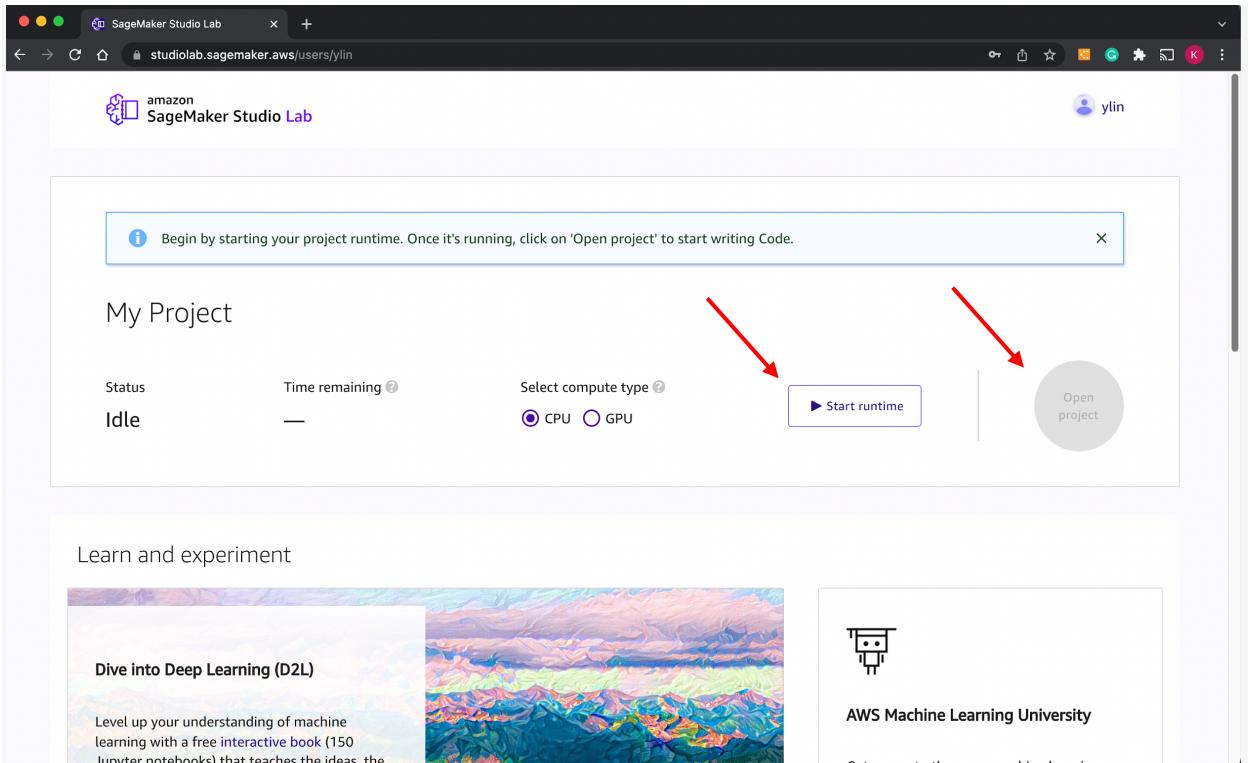
### Step 1: create a SageMaker Studio Lab account

To get started, you need to request a new account from the SageMaker Studio Lab website (<http://studiolab.sagemaker.aws>). You might add to a waitlist (around 1 to 2 day waiting time) and once it is approved, you can sign in.



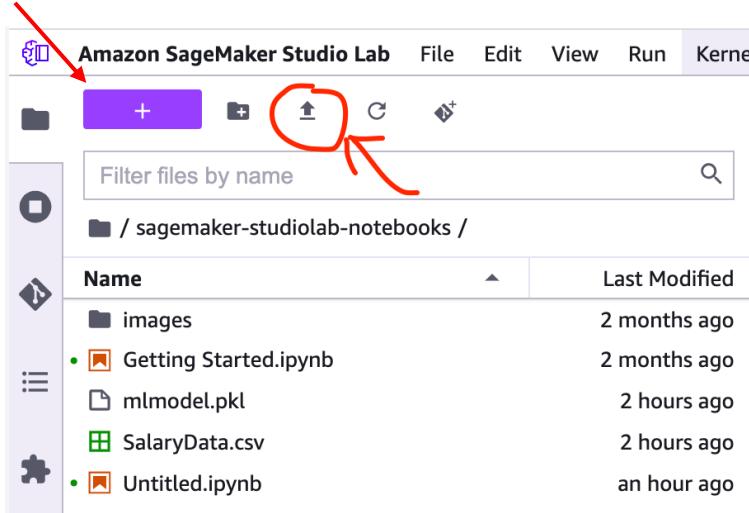
### Step 2: sign in, start runtime and create a project

Once you sign in (<http://studiolab.sagemaker.aws>), click ‘Start runtime’ to enable your instance. Click ‘Open project’, you will be taken to your project where you can create notebooks and many more.



### Step 3: Create a notebook instance and import data

1. Click 'File' -> 'New'->'Notebook' and select 'Python' with default settings.
2. Click the 'upload' button shown below and upload the 'SalaryData.csv' in the same or subfolder.



**Step 4: Copy and run the example below**

1. Copy the title and place it in a 'Markdown' cell. You need to make sure the dropdown is changed to 'Markdown' and then run the cell by either clicking the 'run' button or 'Shift' + 'Return'.



## ## Linear Regression Demo v.1.0.0

### Purpose: This notebook uses Linear Regression Model to predict/forecasting the salary

### Note: Before running this file, you should have SalaryData.csv uploaded to the same folder

2. Copy into a new 'Code' cell and run the code below for install libraries

```
# first, install the required Python libraries  
!pip install numpy pandas scikit-learn pickle-mixin
```

3. Copy and run the code to import the libraries and dataset into your program (check the format of your line after copy/paste)

```
# import the libraries and dataset  
import numpy  
import pandas  
from sklearn.model_selection import train_test_split  
from sklearn.linear_model import LinearRegression  
from sklearn.metrics import mean_squared_error  
import pickle  
# import the dataset using pandas  
dataset = pandas.read_csv('SalaryData.csv')
```

4. Copy and run the markdown for linear regression

```
## Linear Regression (create a model, train and use it)
```

## 5. Copy and run the code for building the model with training data

```
# step to build your model with training and test dataset
# parse dataset into year and salary
year = dataset.iloc[:, :-1].values
salary = dataset.iloc[:, 1].values
# splitting the dataset in half into the training dataset (used to train model)
# and test dataset (used to evaluate how well model does)
year_train, year_test, salary_train, salary_test = train_test_split(year, salary,
test_size = 1/2, random_state = 0)
# fitting simple Linear Regression to the training dataset
regressor = LinearRegression()
regressor.fit(year_train, salary_train)
coefficients = regressor.coef_
# predicting the salary test dataset results
salary_pred = regressor.predict(year_test)
# save our object regressor to the file named mlmodel.pkl
pickle.dump(regressor, open('mlmodel.pkl','wb'))
# loading ml model to compare the outcome
mlmodel = pickle.load(open('mlmodel.pkl','rb'))
```

## 6. Copy the code to run to calculate and display the testing error of our model

```
# Computes and displaying testing error, which is on average how far off our
# predictions are from the actual salaries in the testing data; error is high since we
# have a small dataset.
testing_error = mean_squared_error(salary_test, salary_pred, squared = False)
print("Testing Error of Model: {:.2f}\n".format(testing_error))
```

You should see the following output.

Testing Error of Model: 4579.73

## 7. Copy the code to run and display the prediction

```
# Use model to predict one's salary if they have 20.8 years of experience
# Final step is to run and display the forecasting for salary
prediction = mlmodel.predict([[20.8]])
print('Coefficient : {}'.format(coefficients[0]))
print("Predicted Salary: $ {:.2f}\n".format(prediction[0]))
```

You should see the following output.

Coefficients : 8970.573239339064

Predicted Salary: \$ 210,494.08

### c. Additional Resources/References:

- Amazon SageMaker Studio Lab FAQ  
<https://studiolab.sagemaker.aws/faq>
- Introduction To Machine Learning using Python –  
<https://www.geeksforgeeks.org/introduction-machine-learning-using-python/>
- Build, train, and deploy a machine learning model with Amazon SageMaker  
<https://aws.amazon.com/getting-started/hands-on/build-train-deploy-machine-learning-model-sagemaker>
- Your First Machine Learning Project in Python Step-By-Step –  
<https://machinelearningmastery.com/machine-learning-in-python-step-by-step/>
- Simple Machine Learning Model in Python -  
<https://towardsdatascience.com/simple-machine-learning-model-in-python-in-5-lines-of-code-fe03d72e78c6>
- ML Algorithms For Beginners – <https://medium.com/towards-artificial-intelligence/machine-learning-algorithms-for-beginners-with-python-code-examples-ml-19c6af60daa>
- 8 Fun Machine Learning Projects for Beginners –  
<https://elitedatascience.com/machine-learning-projects-for-beginners>