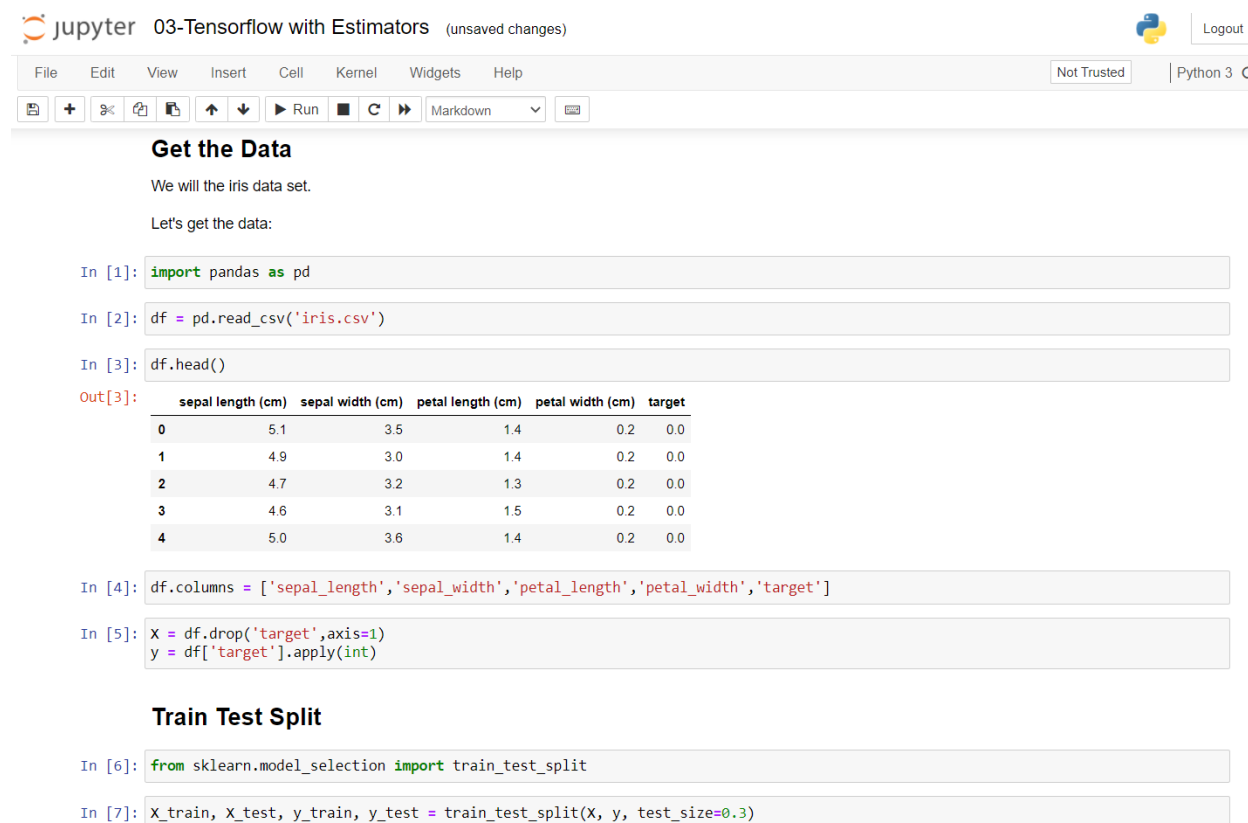


بنام خدا.

با توجه به **issue** تعریف شده، نسبت به نصب و یادگیری **Tensorflow** اقدام شد. در این گزارش سعی شده است مختصراً فعالیت های صورت گرفته شده، شرح داده شود.

تصاویر زیر، از محیط **jupyter notebook** بوده و نتیجه اجرای برخی از دستورات فرا گرفته شده را نشان میدهد.



The screenshot displays a Jupyter Notebook titled "03-Tensorflow with Estimators" with unsaved changes. The interface includes a top menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and output viewing. The notebook content is divided into two sections: "Get the Data" and "Train Test Split".

Get the Data

We will use the Iris data set.

Let's get the data:

```
In [1]: import pandas as pd
```

```
In [2]: df = pd.read_csv('iris.csv')
```

```
In [3]: df.head()
```

Out[3]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

```
In [4]: df.columns = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'target']
```

```
In [5]: X = df.drop('target', axis=1)
```

```
       y = df['target'].apply(int)
```

Train Test Split

```
In [6]: from sklearn.model_selection import train_test_split
```

```
In [7]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

Defining Operations

```
In [16]: add = tf.add(x,y)
sub = tf.subtract(x,y)
mul = tf.multiply(x,y)
```

Running operations with variable input:

```
In [17]: d = {x:20,y:30}
```

```
In [18]: with tf.Session() as sess:
print('Operations with Constants')
print('Addition',sess.run(add,feed_dict=d))
print('Subtraction',sess.run(sub,feed_dict=d))
print('Multiplication',sess.run(mul,feed_dict=d))
```

```
Operations with Constants
Addition 50
Subtraction -10
Multiplication 600
```

Now let's see an example of a more complex operation, using Matrix Multiplication. First we need to create the matrices:

```
In [19]: import numpy as np
# Make sure to use floats here, int64 will cause an error.
a = np.array([[5.0,5.0]])
b = np.array([[2.0],[2.0]])
```

```
In [20]: a
```

```
Out[20]: array([[5., 5.]])
```

Simple Constants

Let's show how to create a simple constant with Tensorflow, which TF stores as a tensor object:

```
In [2]: hello = tf.constant('Hello World')
```

```
In [3]: type(hello)
```

```
Out[3]: tensorflow.python.framework.ops.Tensor
```

```
In [4]: x = tf.constant(100)
```

```
In [5]: type(x)
```

```
Out[5]: tensorflow.python.framework.ops.Tensor
```

Running Sessions

Now you can create a TensorFlow Session, which is a class for running TensorFlow operations.

A `Session` object encapsulates the environment in which `Operation` objects are executed, and `Tensor` objects are evaluated. For example:

```
In [6]: sess = tf.Session()
```

```
In [7]: sess.run(hello)
```

```
Out[7]: b'Hello World'
```

```
In [8]: type(sess.run(hello))
```

```
Out[8]: bytes
```