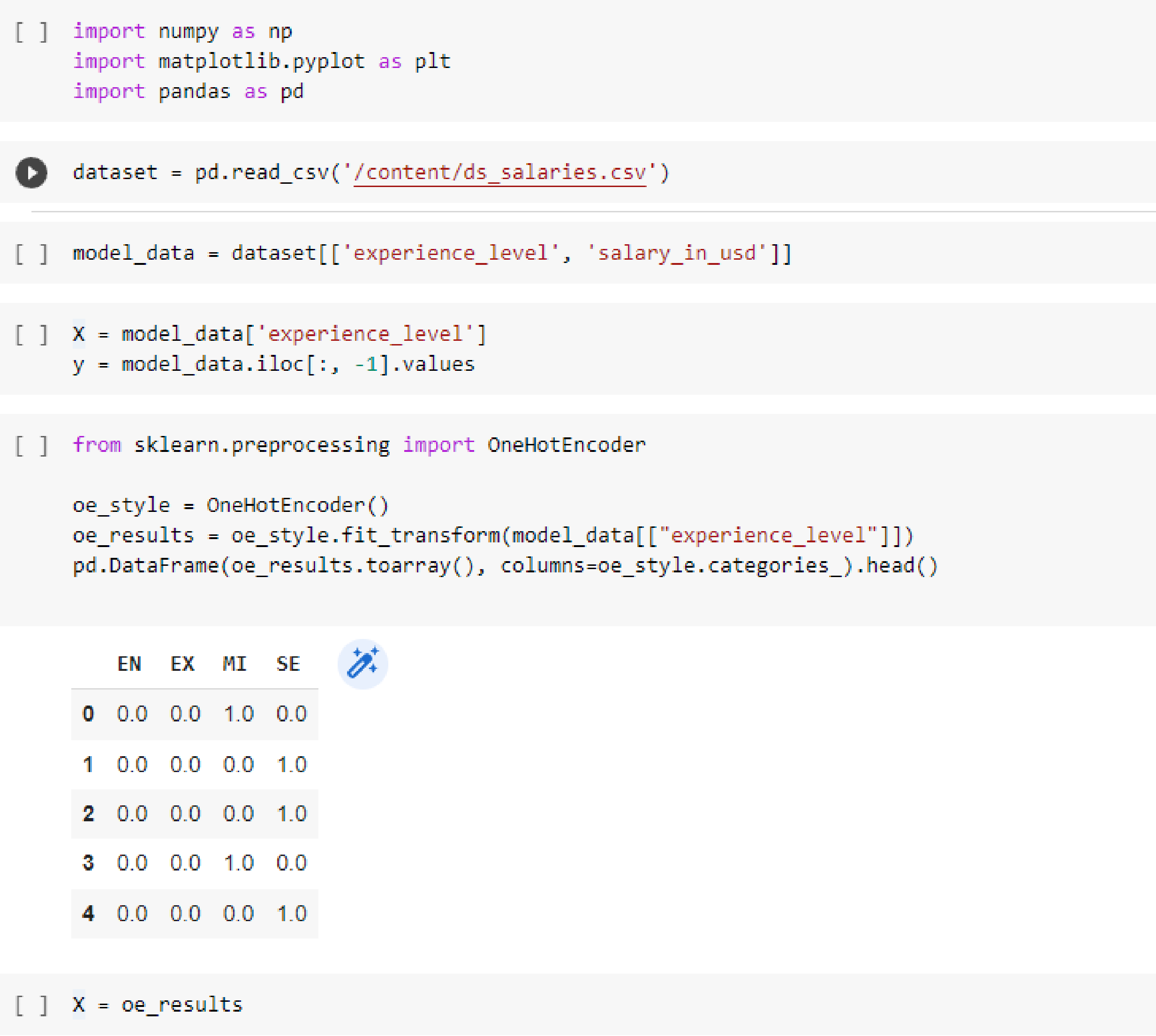
Kenneth Fields
Batch Code: LISUM11
Week 4: Flask Deployment
Submission Date: 8/3/22

1. Data from Kaggle for Data Scientist Salaries - ‘https://www.kaggle.com/datasets/ruchi798/data-science-job-salaries’
2. Model developed on Colab - Simple Linear regression model to predict salary level based on experience level.
3. Flask Web App
4. 

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
dataset = pd.read_csv('/content/ds_salaries.csv')
```

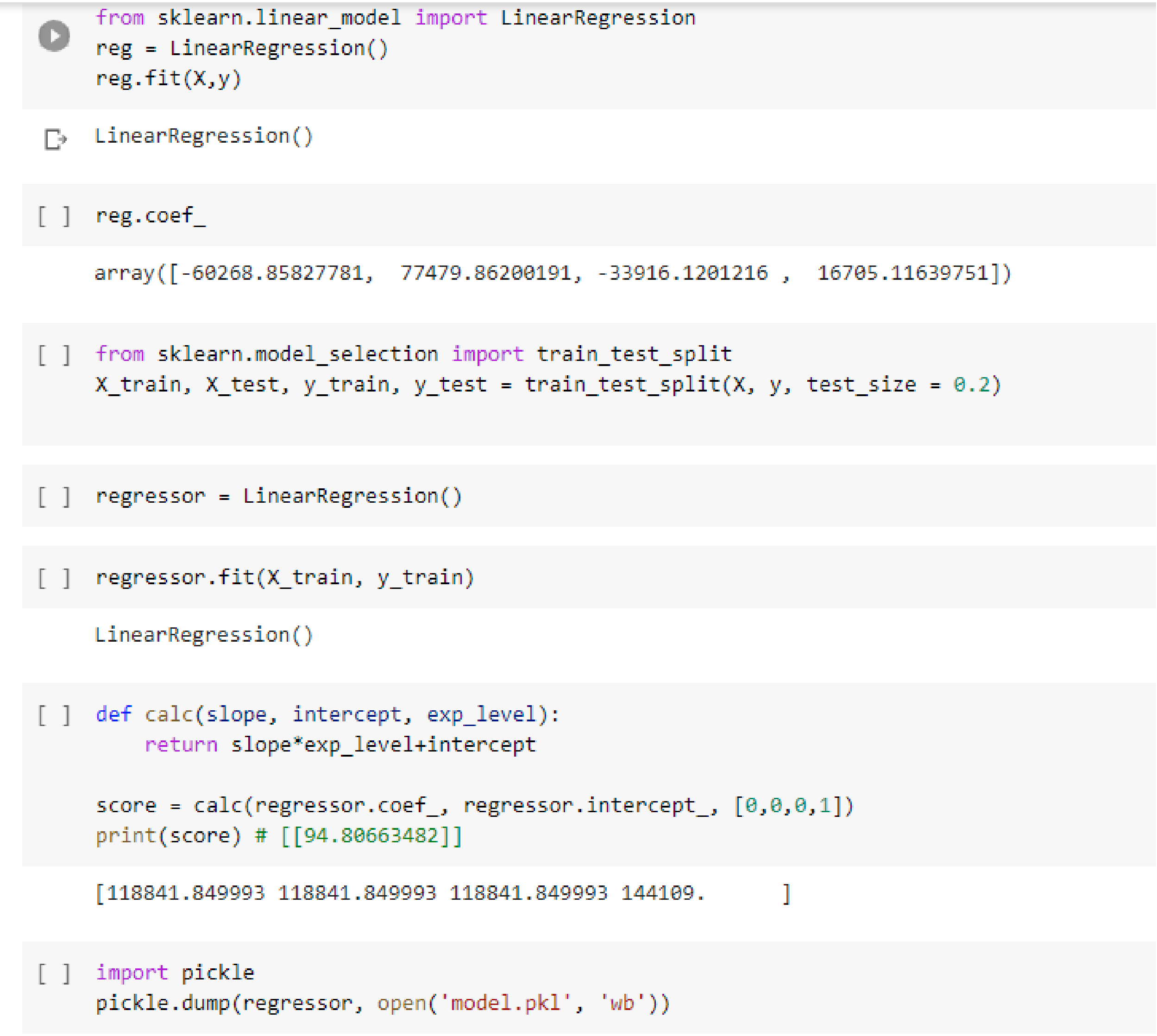
```
model_data = dataset[['experience_level', 'salary_in_usd']]
```

```
X = model_data['experience_level']
y = model_data.iloc[:, -1].values
```

```
from sklearn.preprocessing import OneHotEncoder

oe_style = OneHotEncoder()
oe_results = oe_style.fit_transform(model_data[["experience_level"]])
pd.DataFrame(oe_results.toarray(), columns=oe_style.categories_).head()
```

	EN	EX	MI	SE
0	0.0	0.0	1.0	0.0
1	0.0	0.0	0.0	1.0
2	0.0	0.0	0.0	1.0
3	0.0	0.0	1.0	0.0
4	0.0	0.0	0.0	1.0

```
X = oe_results
```



```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(X,y)
```

```
LinearRegression()
```

```
reg.coef_
```

```
array([-60268.85827781,  77479.86200191, -33916.1201216 ,  16705.11639751])
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

```
regressor = LinearRegression()
```

```
regressor.fit(X_train, y_train)
```

```
LinearRegression()
```

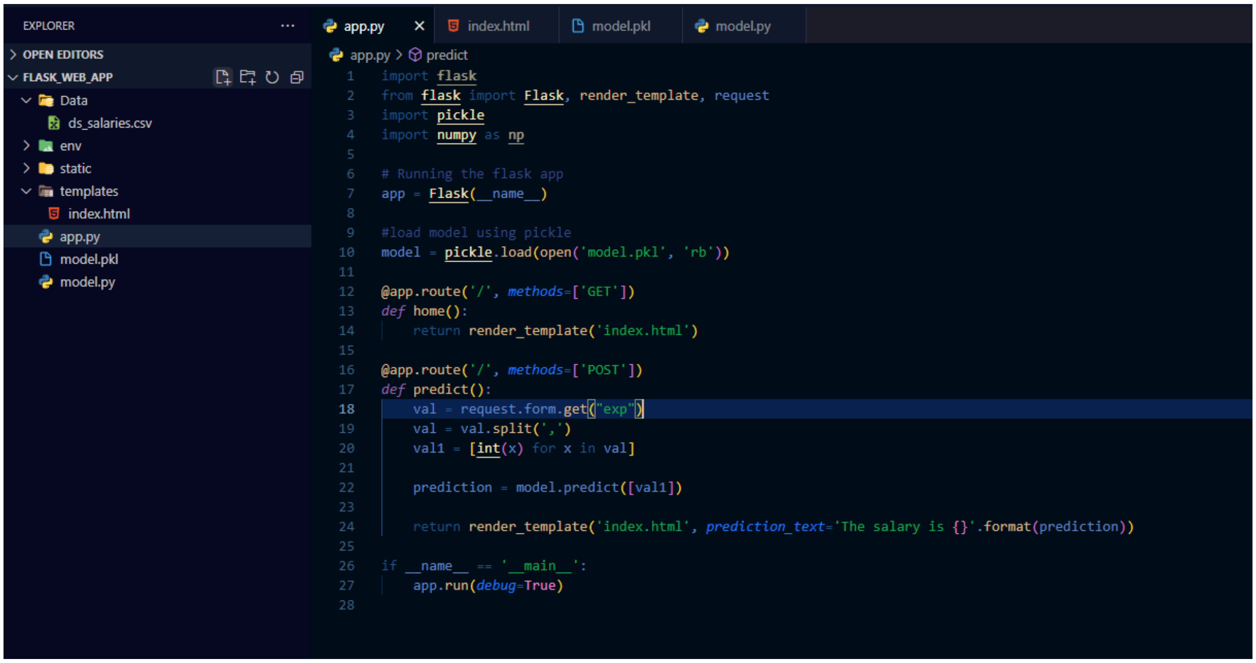
```
def calc(slope, intercept, exp_level):
    return slope*exp_level+intercept

score = calc(regressor.coef_, regressor.intercept_, [0,0,0,1])
print(score) # [[94.80663482]]
```

```
[118841.849993 118841.849993 118841.849993 144109.      ]
```

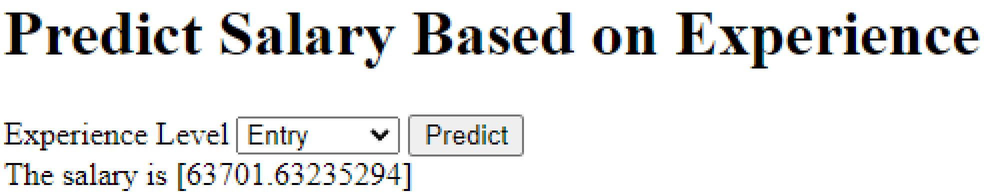
```
import pickle
pickle.dump(regressor, open('model.pkl', 'wb'))
```

5.

6.



```python
import flask
from flask import Flask, render_template, request
import pickle
import numpy as np

# Running the flask app
app = Flask(__name__)

#load model using pickle
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/', methods=['GET'])
def home():
    return render_template('index.html')

@app.route('/', methods=['POST'])
def predict():
    val = request.form.get("exp")
    val = val.split(',')
    val1 = [int(x) for x in val]

    prediction = model.predict([val1])

    return render_template('index.html', prediction_text='The salary is {}'.format(prediction))

if __name__ == '__main__':
    app.run(debug=True)
```



Predict Salary Based on Experience

Experience Level Entry

The salary is [63701.63235294]

7.