

Week 4 ML Deployment Document

Name: Mike Wang

Batch Code: LISUM21

Date: 5/26/2023

Submitted to: Data Glacier

Introduction

This project aims to predict job salary using Extreme Gradient Boosting (XGB) ML model and then deploy the model on Flask API through a web app service.

Data Information

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

Data columns (total 6 columns):

#	Column	Non-Null Count	Dtype
---	-----	-----	-----
0	Age	373 non-null	float64
1	Gender	373 non-null	object
2	Education Level	373 non-null	object
3	Job Title	373 non-null	object
4	Years of Experience	373 non-null	float64
5	Salary	373 non-null	float64

Data Preprocessing

- Split categorical variables to get job level and job title features;
- Use dummy coding to code Gender, Education level, and job titles for better model performance;

```
df.loc[df['job_level'] == 'Senior', 'Senior'] = 1
df.loc[df['job_level'] != 'Senior', 'Senior'] = 0
df.loc[df['job_level'] == 'Junior', 'Junior'] = 1
df.loc[df['job_level'] != 'Junior', 'Junior'] = 0
```

```
df = df.drop(['job_level'], axis = 1)
```

```
from sklearn import preprocessing
```

```
encode = preprocessing.LabelEncoder()
```

```
df['Education'] = encode.fit_transform(df['Education Level'])
df['jobs'] = encode.fit_transform(df['title'])
```

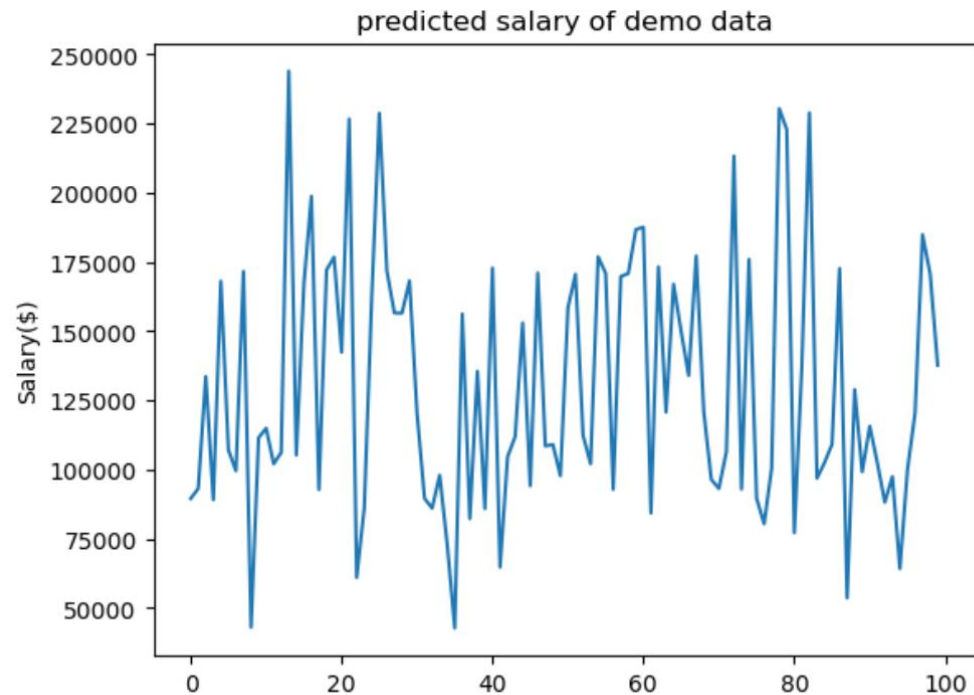
```
df['Male'] = pd.get_dummies(df['Gender'], drop_first=True)
```

```
df.head()
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary	job title	title	Senior	Junior	Education	jobs	Male
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0	[Engineer]	Software Engineer	0.0	0.0	0	101	1
1	28.0	Female	Master's	Data Analyst	3.0	65000.0	[Analyst]	Data Analyst	0.0	0.0	1	23	0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0	[Manager]	Manager	1.0	0.0	2	63	1
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0	[Associate]	Sales Associate	0.0	0.0	0	90	0
4	52.0	Male	Master's	Director	20.0	200000.0	[]	Director	0.0	0.0	1	31	1

Model Build

After data processing, we implement XGB Regressor to fit the dataset, predict job salary, and then save the model for later deployment.



Model Deployment

Use Flask to deploy web applications allow users inputs. Then take the inputs to predict model outcomes.

```
: app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    int_features = [int(x) for x in request.form.values()]
    if int_features[2] == 1:
        int_features.insert(0, 3)
    elif int_features[2] == 0:
        int_features.insert(1, 3)
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

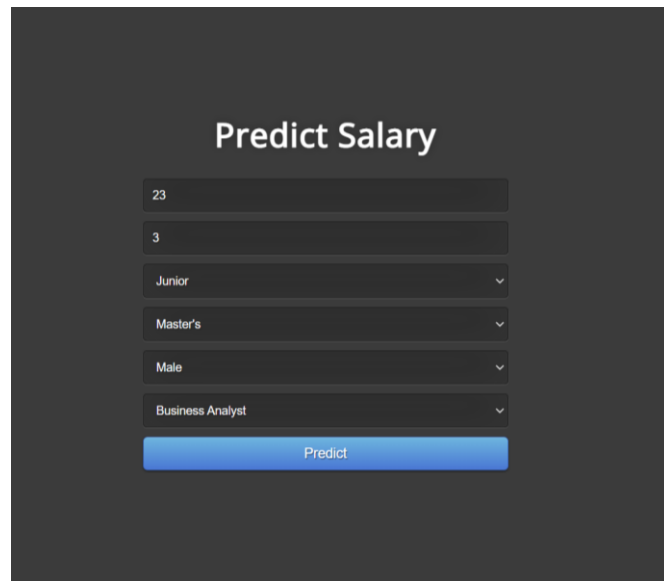
    return render_template('index.html', prediction_text='The estimated salary is $ {}'.format(output))

@app.route('/results', methods=['POST'])
def results():
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)
```

Run on Website

Create a simple html web that is based on the given template. Testing the API to check if it's working.



Predict Salary

23

3

Junior

Master's

Male

Business Analyst

Predict

Predict Salary



The estimated salary is \$ 64260.41