Inna Soltsman-Groysman
LISUM03
09/13/2021

```python
app = Flask(__name__)


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


@app.route('/predict/')
def price_predict():
    model = pickle.load(open('model.pickle', 'rb'))
    age = request.args.get('Age')
    experience = request.args.get('Experience')
    income = request.args.get('Income')
    zip_code = request.args.get('Zip_code')
    family = request.args.get('Family')
    cCAvg = request.args.get('CCAvg')
    education = request.args.get('Education')

    test_df = pd.DataFrame(
        {'Age': [age], 'Experience': [experience], 'Income': [income], 'Zip_code': [zip_code], 'Family': [family],
         'CCAvg': [cCAvg],
         'Education': [education]})

    predict = model.predict(test_df)

    return jsonify({'Loan approved': str(predict)})


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

```html
<!DOCTYPE html>
<html >
<head>
    <meta charset="UTF-8">
    <title>ML API</title>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">


</head>

<body>
 <div class="Login">
     <h1>Predict Loan Approval </h1>
     <form action="{{url_for('predict')}}"method="post">
         <input type="text" name="age" placeholder="Age" required="required" />
         <input type="text" name="experience" placeholder="experience" required="required" />
         <input type="text" name="income" placeholder="income" required="required" />
         <input type="text" name="zip_code" placeholder="zip_code" required="required" />
         <input type="text" name="family" placeholder="family" required="required" />
         <input type="text" name="cCAvg" placeholder="cCAvg" required="required" />
         <input type="text" name="education" placeholder="education" required="required" />

         <button type="submit" name="btn btn-primary btn-large">Predict</button>
     </form>



|
     <br>
     <br>
     {{ prediction_text }}
 </div>

</body>
</html>
```

```python
import pandas as pd
import pickle
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression


Bank_data = pd.read_csv('Bank_Personal_Loan_Modelling.csv')
X = Bank_data[['Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg', 'Education']]


y = Bank_data['Personal Loan']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)


lm = LogisticRegression()
lm.fit(X_train, y_train)


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

```python
import ...


# %%
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():
    """
    For rendering results on HTML GUI
    :return:
    """

    int_feature = []
    for x in request.form.values():
        if x != '':
            int_feature.append(x)

    final_feature = [np.array(int_feature)]
    prediction = model.predict(final_feature)

    return render_template('index.html', prediction_text='Loan should be {0}'.format(prediction[0]))


if __name__ == '__main__':
    app.run(port=5555, debug=True)
```

	KEY	VALUE	DESCRIPTION		Bulk Edit
☑	Age	31			
☑	Experience	3			
☑	Income	10000			
☑	Zip_code	02026			
☑	Family	2			
☑	CCAvg	2			
☑	Education	2			
	Key	Value	Description		

```
1  {
2      "Loan approved": "[1]"
3  }
```

Predict Loan Approval

31

2

10000

02026

2

2

2

Predict

Loan should be 1