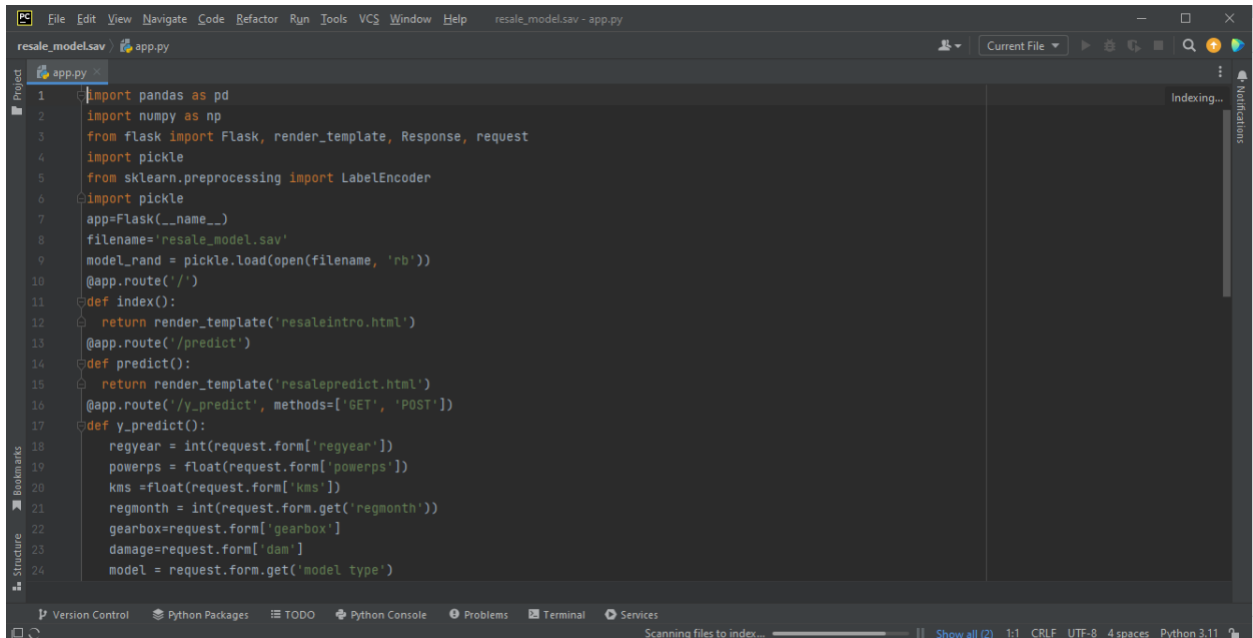
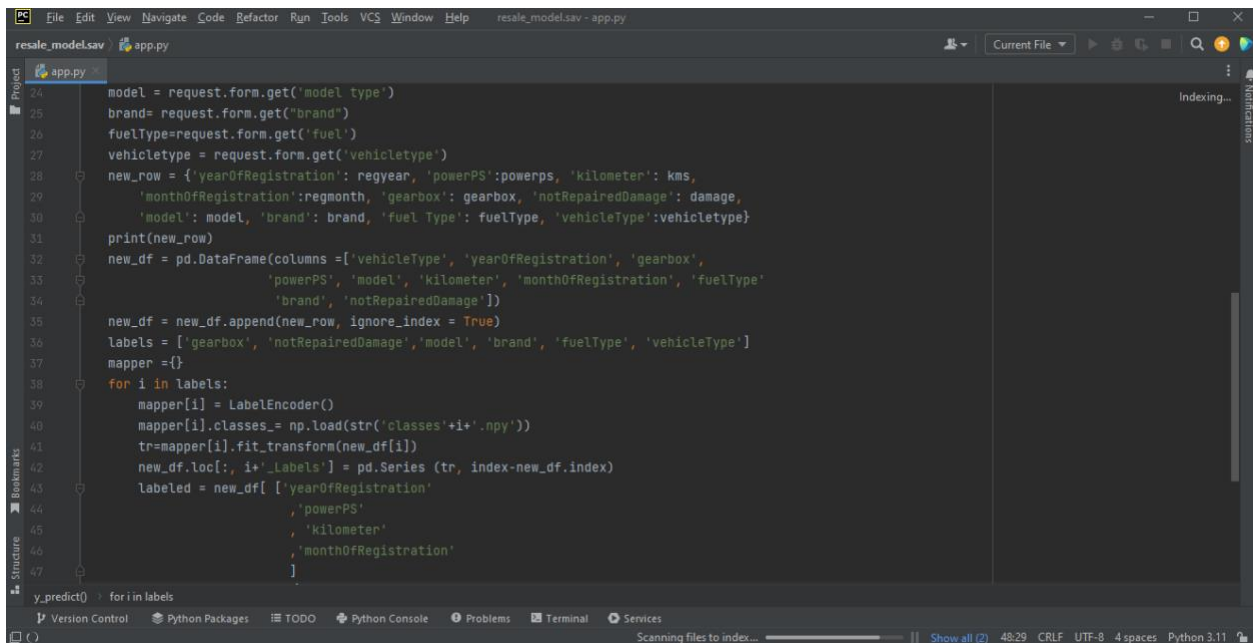


INTEGRATING FLASK WITH SCORING POINT



```
1 import pandas as pd
2 import numpy as np
3 from flask import Flask, render_template, Response, request
4 import pickle
5 from sklearn.preprocessing import LabelEncoder
6 import pickle
7 app=Flask(__name__)
8 filename='resale_model.sav'
9 model_rand = pickle.load(open(filename, 'rb'))
10 @app.route('/')
11 def index():
12     return render_template('resaleintro.html')
13 @app.route('/predict')
14 def predict():
15     return render_template('resalepredict.html')
16 @app.route('/y_predict', methods=['GET', 'POST'])
17 def y_predict():
18     regyear = int(request.form['regyear'])
19     powerps = float(request.form['powerps'])
20     kms = float(request.form['kms'])
21     regmonth = int(request.form.get('regmonth'))
22     gearbox=request.form['gearbox']
23     damage=request.form['dam']
24     model = request.form.get('model type')
```



```
24     model = request.form.get('model type')
25     brand= request.form.get("brand")
26     fuelType=request.form.get('fuel')
27     vehicletype = request.form.get('vehicletype')
28     new_row = {'yearOfRegistration': regyear, 'powerPS':powerps, 'kilometer': kms,
29               'monthOfRegistration':regmonth, 'gearbox': gearbox, 'notRepairedDamage': damage,
30               'model': model, 'brand': brand, 'fuel Type': fuelType, 'vehicleType':vehicletype}
31     print(new_row)
32     new_df = pd.DataFrame(columns=['vehicleType', 'yearOfRegistration', 'gearbox',
33                                   'powerPS', 'model', 'kilometer', 'monthOfRegistration', 'fuelType',
34                                   'brand', 'notRepairedDamage'])
35     new_df = new_df.append(new_row, ignore_index = True)
36     labels = ['gearbox', 'notRepairedDamage','model', 'brand', 'fuelType', 'vehicleType']
37     mapper ={}
38     for i in labels:
39         mapper[i] = LabelEncoder()
40         mapper[i].classes_ = np.load(str('classes'+i+'.npy'))
41         tr=mapper[i].fit_transform(new_df[i])
42         new_df.loc[:, i+'Labels'] = pd.Series(tr, index=new_df.index)
43     labeled = new_df[['yearOfRegistration',
44                       'powerPS',
45                       'kilometer',
46                       'monthOfRegistration',
47                       'yearOfRegistrationLabels',
48                       'powerPSLabels',
49                       'kilometerLabels',
50                       'monthOfRegistrationLabels',
51                       'gearboxLabels',
52                       'notRepairedDamageLabels',
53                       'modelLabels',
54                       'brandLabels',
55                       'fuelTypeLabels',
56                       'vehicleTypeLabels']
57     y_predict() for i in labels
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help resale_model.sav - app.py
resale_model.sav app.py
new_df = new_df.append(new_row, ignore_index = True)
labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']
mapper = {}
for i in labels:
    mapper[i] = LabelEncoder()
    mapper[i].classes_ = np.load(str('classes'+i+'.npy'))
    tr=mapper[i].fit_transform(new_df[i])
    new_df.loc[:, i+'_'+Labels] = pd.Series (tr, index=new_df.index)
labeled = new_df[ ['yearOfRegistration'
                  , 'powerPS'
                  , 'kilometer'
                  , 'monthOfRegistration'
                  ]
                +[x+'_'+Labels' for x in labels]]
X=labeled.values
print(X)
y_prediction=model_rand.predict(X)
print(y_prediction)
return render_template('resalepredict.html',ypred='The resale value predicted is {:.2f}$'.format(y_prediction[0]))
if __name__ == '__main__':
    app.run(host='localhost', debug=True, threaded=False)
```

Car resale value Predictor

Welcome! To predict your used car price click the below button!

[Check price](#)



Get the Accurate Resale Value of Your Car

Registration year :

Registration Month :

Power of car in PS:

Kilometers that car have driven :

Gear type : ☒ Manual ☐ Automatic ☐ Not declared

Your car is repaired or damaged : ☒ Yes ☐ No ☐ Not declared

Model Type :

Brand :

Fuel Type :

Vehicle type:

