

APP BUILDING :

```
import pickle

import joblib
import numpy as np
import sklearn
from flask import Flask, render_template, request
from sklearn.preprocessing import StandardScaler

app = Flask(__name__)
model = pickle.load(open("vot_reg.pkl", "rb"))
@app.route('/', methods=['GET'])
def Home():
    return render_template('resalepredict.html')

# @app.route('/', methods=['GET'])
# def predictpage():
#     return render_template('resalepredict.html')

standard_to = StandardScaler()
@app.route("/predict", methods=['GET', 'POST'])
def predict():

    Fuel_Type_Diesel=0

    if request.method == 'POST':
```

```

Year = int(request.form['year'])
Present_Price=float(request.form['Present_Price'])
Kms_Driven=int(request.form['Kms_Driven'])
Owner=int(request.form['Owner'])
Fuel_Type_Petrol=request.form['fueltype']
if(Fuel_Type_Petrol=='petrol'):
    Fuel_Type_Petrol=1
    Fuel_Type_Diesel=0
else:
    Fuel_Type_Petrol=0
    Fuel_Type_Diesel=1
Year=2020-Year

Seller_Type_Individual=request.form['Seller_Type_Individual']
if(Seller_Type_Individual=='Individual'):
    Seller_Type_Individual=1
else:
    Seller_Type_Individual=0

Transmission_Mannual=request.form['Transmission_Mannual']
if(Transmission_Mannual=='Mannual'):
    Transmission_Mannual=1
else:
    Transmission_Mannual=0
prediction=model.predict(np.array([[Year,
                                     Present_Price,
                                     Kms_Driven,

```

```

Owner,
Fuel_Type_Diesel,
Fuel_Type_Petrol,

Seller_Type_Individual,
Transmission_Mannual]])
    output=round(prediction[0],2)
    if output<0:
        return
render_template('resaleresult.html',prediction_texts="Sorry
you cannot sell this car")
    else:
        return
render_template('resaleresult.html',prediction_texts="You can
sell the Car at {} lakhs".format(output))
    else:
        return render_template('resaleresult.html')

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

```

```
app.py X Car_Price_Pred.ipynb resaleintro.html resaleresult.html resalepredict.html
app.py > ...
1 import pickle
2
3 import joblib
4 import numpy as np
5 import sklearn
6 from flask import Flask, render_template, request
7 from sklearn.preprocessing import StandardScaler

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Dharun R\CRV> & "C:\Users\Dharun R\CRV\env\Scripts\Activate.ps1"
(env) PS C:\Users\Dharun R\CRV> flask run
C:\Users\Dharun R\CRV\env\Lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.0.2 when u
sing version 1.1.3. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\Dharun R\CRV\env\Lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.0.2 when u
sing version 1.1.3. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\Dharun R\CRV\env\Lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator VotingRegressor from version 1.0.2 when using v
ersion 1.1.3. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
█
```

App Building