

CAR RESALE VALUE PREDICTION

```
import libraries

import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
import pickle
from lightgbm import LGBMRegressor

read preprocessed data

data = pd.read_csv("autos_preprocessed.csv")

metrics evaluation

def find_scores(Y_actual, Y_pred, X_train):
    scores = dict()
    mae = mean_absolute_error(Y_actual, Y_pred)
    mse = mean_squared_error(Y_actual, Y_pred)
    rmse = np.sqrt(mse)
    rmsle = np.log(rmse)
    r2 = r2_score(Y_actual, Y_pred)
    n, k = X_train.shape
    adj_r2_score = 1 - ((1-r2)*(n-1)/(n-k-1))

    scores['mae']=mae
    scores['mse']=mse
    scores['rmse']=rmse
    scores['rmsle']=rmsle
    scores['r2']=r2
    scores['adj_r2_score']=adj_r2_score

    return scores

traing and testing

X = labeled.iloc[:,1:].values
Y = labeled.iloc[:,0].values.reshape(-1,1)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4,
```

```
random_state=42)
model =
LGBMRegressor(boosting_type="gbdt", learning_rate=0.07, metric="rmse", n_estimators=300, objective="root_mean_squared_error", random_state=42, reg_sqrt=True)

model.fit(X_train, Y_train)

Y_pred = model.predict(X_test)

find_scores(Y_test, Y_pred, X_train)

save the model

pickle.dump(model, open('resale_model.sav', 'wb'))
```