TOÁN ỨNG DỤNG VÀ XÁC SUẤT

Project 01

1.Preproccessing data test

reading dadaTrain to take unquie value

In [64]:

```
import numpy as np
import pandas as pd
import matplotlib.pylab as plt
dataX_train = pd.read_csv("X_train.csv")
dataY_train = pd.read_csv("Y_train.csv")
data=pd.concat([dataX_train,dataY_train['price']],axis=1)
data=data.dropna()
```

In [65]:

```
dataUniqueManufacturer=dataX_train['manufacturer'].unique()
dataUniqueTransmission=dataX_train['transmission'].unique()
dataUniqueEngineFuel=dataX_train['engineFuel'].unique()
dataUniqueEngineType=dataX_train['engineType'].unique()
dataUniqueBodyType=dataX_train['bodyType'].unique()
dataUniqueDrivetrain=dataX_train['drivetrain'].unique()
dataUniqueFeature_0=dataX_train['feature_0'].unique()
dataUniqueColor=dataX_train['color'].unique()
```

supported function

function checkvalue : check input into colum

In [66]:

```
def checValue(input,colums):
    return input in colums
```

function fillingMissingData: to fill out value

In [67]:

```
def fillingMissingData(dataUnique,namFeature,valueFrequency):

    n=dataX_test[namFeature].size
    datafeature=dataX_test[namFeature]
    for i in range(n):
        if not checValue(datafeature.loc[i],dataUnique):
            dataX_test.loc[i,namFeature]=valueFrequency
```

Reading datatest

In [68]:

```
dataX_test= pd.read_csv("X_test.csv")
dataY_test= pd.read_csv("Y_test.csv")
```

handle missingdata and strange value

In [69]:

```
fillingMissingData(dataUniqueManufacturer,'manufacturer','Volkswagen')

fillingMissingData(dataUniqueTransmission,'transmission','mechanical')

fillingMissingData(dataUniqueEngineFuel,'engineFuel','gasoline')

fillingMissingData(dataUniqueEngineType,'engineType','gasoline')

fillingMissingData(dataUniqueBodyType,'bodyType','sedan')

fillingMissingData(dataUniqueDrivetrain,'drivetrain','front')

fillingMissingData(dataUniqueColor,'color','black')
```

In [70]:

```
dataX_test["feature_0"].fillna(dataX_train["feature_0"].value_counts().idxmax(), inplace = True)
dataX_test["feature_1"].fillna(dataX_train["feature_1"].value_counts().idxmax(), inplace = True)
dataX_test["feature_2"].fillna(dataX_train["feature_2"].value_counts().idxmax(), inplace = True)
dataX_test["feature_3"].fillna(dataX_train["feature_3"].value_counts().idxmax(), inplace = True)
dataX_test["feature_4"].fillna(dataX_train["feature_4"].value_counts().idxmax(), inplace = True)
dataX_test["feature_5"].fillna(dataX_train["feature_5"].value_counts().idxmax(), inplace = True)
dataX_test["feature_6"].fillna(dataX_train["feature_6"].value_counts().idxmax(), inplace = True)
dataX_test["feature_7"].fillna(dataX_train["feature_7"].value_counts().idxmax(), inplace = True)
dataX_test["feature_8"].fillna(dataX_train["feature_8"].value_counts().idxmax(), inplace = True)
dataX_test["feature_9"].fillna(dataX_train["feature_9"].value_counts().idxmax(), inplace = True)
```

reading format datatest

In [71]:

```
formData_test=pd.read_csv("formX_test.csv")
```

connecting data test with format datatest

In [72]:

```
arryManufacturer= np.concatenate((formData_test['manufacturer'], dataX_test['manufacturer']))
arryTransmission= np.concatenate((formData_test['transmission'], dataX_test['transmission']))
arryColor= np.concatenate((formData_test['color'], dataX_test['color']))
arryBodyType= np.concatenate((formData_test['bodyType'], dataX_test['bodyType']))
arryDrivetrain= np.concatenate((formData_test['drivetrain'], dataX_test['drivetrain']))
arryEngineType= np.concatenate((formData_test['engineType'], dataX_test['engineType']))
arryEngineFuel= np.concatenate((formData_test['engineFuel'], dataX_test['engineFuel']))
arryFeature_0= np.concatenate((formData_test['feature_0'], dataX_test['feature_0']))
arryFeature_1= np.concatenate((formData_test['feature_1'], dataX_test['feature_1']))
arryFeature_2= np.concatenate((formData_test['feature_2'], dataX_test['feature_2']))
arryFeature_3= np.concatenate((formData_test['feature_3'], dataX_test['feature_3']))
arryFeature_4= np.concatenate((formData_test['feature_4'], dataX_test['feature_4']))
arryFeature_5= np.concatenate((formData_test['feature_5'], dataX_test['feature_5']))
arryFeature_6= np.concatenate((formData_test['feature_6'], dataX_test['feature_6']))
arryFeature_7= np.concatenate((formData_test['feature_7'], dataX_test['feature_7']))
arryFeature_8= np.concatenate((formData_test['feature_8'], dataX_test['feature_8']))
arryFeature_9= np.concatenate((formData_test['feature_9'], dataX_test['feature_9']))
```

In [73]:

```
arryFeature_0=arryFeature_0.astype(int)
arryFeature_1=arryFeature_1.astype(int)
arryFeature_2=arryFeature_2.astype(int)
arryFeature_3=arryFeature_3.astype(int)
arryFeature_4=arryFeature_4.astype(int)
arryFeature_5=arryFeature_5.astype(int)
arryFeature_6=arryFeature_6.astype(int)
arryFeature_7=arryFeature_7.astype(int)
arryFeature_8=arryFeature_8.astype(int)
arryFeature_9=arryFeature_9.astype(int)



arryFeature_0Dataframe=pd.DataFrame(arryFeature_0,columns = ['feature_0'])
arryFeature_1Dataframe=pd.DataFrame(arryFeature_1,columns = ['feature_1'])
arryFeature_2Dataframe=pd.DataFrame(arryFeature_2,columns = ['feature_2'])
arryFeature_3Dataframe=pd.DataFrame(arryFeature_3,columns = ['feature_3'])
arryFeature_4Dataframe=pd.DataFrame(arryFeature_4,columns = ['feature_4'])
arryFeature_5Dataframe=pd.DataFrame(arryFeature_5,columns = ['feature_5'])
arryFeature_6Dataframe=pd.DataFrame(arryFeature_6,columns = ['feature_6'])
arryFeature_7Dataframe=pd.DataFrame(arryFeature_7,columns = ['feature_7'])
arryFeature_8Dataframe=pd.DataFrame(arryFeature_8,columns = ['feature_8'])
arryFeature_9Dataframe=pd.DataFrame(arryFeature_9,columns = ['feature_9'])

dumyFeature_0_9=pd.concat([arryFeature_0Dataframe,
                           arryFeature_1Dataframe,
                          arryFeature_2Dataframe,
                           arryFeature_3Dataframe,
                          arryFeature_4Dataframe,
                           arryFeature_5Dataframe,
                          arryFeature_6Dataframe,
                           arryFeature_7Dataframe,
                          arryFeature_8Dataframe,
                           arryFeature_9Dataframe],axis=1)
```

getting dummy

In [74]:

```
DummyarryManufacturer=pd.get_dummies(arryManufacturer)
DummyarryTransmission=pd.get_dummies(arryTransmission)

DummyarryColor=pd.get_dummies(arryColor)


DummyarryBodyType=pd.get_dummies(arryBodyType)

DummyarryDrivetrain=pd.get_dummies(arryDrivetrain)


DummyarryEngineType=pd.get_dummies(arryEngineType)


DummyarryEngineFuel=pd.get_dummies(arryEngineFuel)
```

handle numberic attribute

In [75]:

```
dataOdometerAndYear=pd.concat([dataX_test['odometer'],dataX_test['year']],axis=1)
```

In [76]:

```
from sklearn.impute import KNNImputer
imputer = KNNImputer(n_neighbors=1)
df_filled = imputer.fit_transform(dataOdometerAndYear)
pd.isnull(df_filled).sum()

a = pd.DataFrame(df_filled)
arryodometer= np.concatenate((formData_test['odometer'],a[0]))

arryodometerdataframe=pd.DataFrame(arryodometer,columns = ['odometer'])

arryYear= np.concatenate((formData_test['year'],a[1]))

arryYeardataframe=pd.DataFrame(arryYear,columns = ['year'])
```

format datatest

In [77]:

```
datafinaltest=pd.concat([arryodometerdataframe,arryodometerdataframe**2,arryYeardataframe,
arryYeardataframe**2,
                         DummyarryManufacturer,DummyarryTransmission,DummyarryColor,
                         DummyarryBodyType,DummyarryDrivetrain,dumyFeature_0_9 ],axis=1)
```

In [78]:

```
datafinaltest
```

Out[78]:

	odometer	odometer	year	year	Acura	Alfa Romeo	Audi	BMW	Buick	Cadillac
0	48000.0	2.304000e+09	2014.0	4056196.0	0	0	0	0	0	0
1	320000.0	1.024000e+11	2000.0	4000000.0	0	0	0	0	0	0
2	164000.0	2.689600e+10	2011.0	4044121.0	0	0	0	0	0	0
3	385672.0	1.487429e+11	1998.0	3992004.0	0	0	0	0	0	0
4	215652.0	4.650579e+10	2005.0	4020025.0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
30045	252000.0	6.350400e+10	2008.0	4032064.0	1	0	0	0	0	0
30046	290000.0	8.410000e+10	1997.0	3988009.0	0	0	0	0	0	0
30047	250000.0	6.250000e+10	1993.0	3972049.0	0	0	0	0	0	0
30048	267000.0	7.128900e+10	2002.0	4008004.0	0	0	0	0	0	0
30049	200000.0	4.000000e+10	1990.0	3960100.0	0	0	0	0	0	0

30050 rows × 93 columns

reading the best model from file.sav

In [79]:

```
import joblib
filename = 'finalized_model.sav'
loaded_model = joblib.load(filename)
```

predection

In [80]:

```
y_predicted =loaded_model.predict(datafinaltest)
```

calculate RMSE

In [81]:

```
import numpy as geek
object=[0,1,2,3,4,5,6,7,8,9,10,
       11,12,13,14,15,16,17,18,19,20,
       21,22,23,24,25,26,27,28,29,30,
       31,32,33,34,35,36,37,38,39,40,41,
       42,43,44,45,46,47,48,49]

result = geek.delete(y_predicted, object)
```

In [82]:

```
A=result-dataY_test['price']
temp=A**2
sumOfTemp=temp.sum()
n=result.size
import math
RMSE=math.sqrt(sumOfTemp/n)
print(RMSE)
```

```
2783.0964597905295
```

In [ ]:

```

```