Name: Esha asif
Roll no:034
Section:BSAI 4A
Subject:Programming for AI
LAB TASK 1

Code:

```
#import pandas as pd
# import numpy as np
# import matplotlib.pyplot as plt
# import seaborn as sns
# from sklearn.model_selection import train_test_split, GridSearchCV
# from sklearn.preprocessing import LabelEncoder, StandardScaler
# from sklearn.impute import SimpleImputer
# from sklearn.ensemble import RandomForestRegressor
# from xgboost import XGBRegressor
# from sklearn.metrics import mean_squared_error



# train = pd.read_csv("train.csv")
# test = pd.read_csv("test.csv")
#df.head()
#print(df)
#df.tail()
#df.info()
#df.describe()
#print(df.count())
```

```
#df.nunique()
#print(df.isnull().sum())

print(train.columns)
print(test.columns)


for df in [train, test]:
  df.fillna(df.median(numeric_only=True), inplace=True)


categorical_cols = train.select_dtypes(include=['object']).columns
for col in categorical_cols:
   encoder = LabelEncoder()
encoder = LabelEncoder()
train[col] = encoder.fit_transform(train[col].fillna("Missing").astype(str))
test[col] = encoder.transform(test[col].fillna("Missing").astype(str))

X = train.drop("SalePrice", axis=1)
y = np.log(train["SalePrice"])
X_test = test

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder

categorical_cols = X_train.select_dtypes(include=['object']).columns

imputer = SimpleImputer(strategy='most_frequent')
X_train[categorical_cols] = imputer.fit_transform(X_train[categorical_cols])
X_test[categorical_cols] = imputer.transform(X_test[categorical_cols])

le = LabelEncoder()

for col in categorical_cols:
   X_train[col] = le.fit_transform(X_train[col].astype(str))

   X_test[col] = X_test[col].apply(lambda x: le.transform([x])[0] if x in le.classes_ else -1)
X_train.fillna(0, inplace=True)
X_test.fillna(0, inplace=True)
model = XGBRegressor(n_estimators=500, learning_rate=0.05, max_depth=4)
model.fit(X_train, y_train)
```

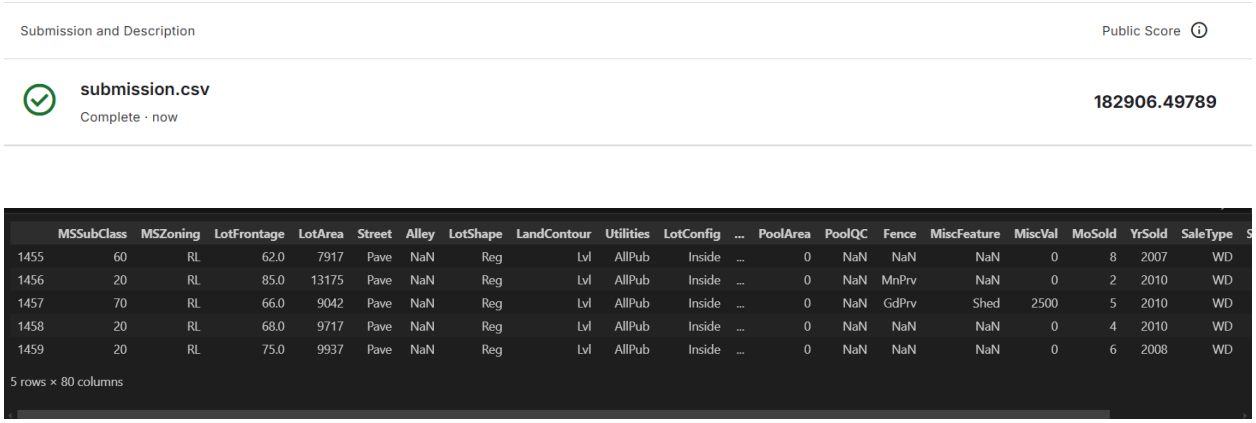
```
missing_cols = set(X_train.columns) - set(X_val.columns)
for col in missing_cols:
    X_val[col] = 0  # Add missing columns with default value

X_val = X_val[X_train.columns]
for col in X_val.select_dtypes(include=['object', 'category']).columns:
    le = LabelEncoder()
    X_val[col] = le.fit_transform(X_val[col].astype(str))
X_val = X_val.astype(np.float32)
y_pred = model.predict(X_val)


import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
test_data = pd.read_csv("C:\\Users\\eshaa\\OneDrive\\Desktop\\House prediction\\test.csv")
test_ID = test_data['Id']

X_test = test_data.drop(columns=['Id'])
print(X_test.dtypes)

label_encoder = LabelEncoder()
categorical_columns = X_test.select_dtypes(include=['object']).columns

for col in categorical_columns:
    X_test[col] = label_encoder.fit_transform(X_test[col].astype(str))

if X_test.isnull().sum().sum() > 0:
    X_test = X_test.fillna(0)
# Ensure all columns are numeric
print(X_test.dtypes)
X_test = X_test.astype(np.float32)
test_predictions = model.predict(X_test)

if len(test_ID) != len(test_predictions):
    raise ValueError(f"Mismatch between test_ID ({len(test_ID)}) and test_predictions
({len(test_predictions)}) length")

submission = pd.DataFrame({"Id": test_ID, "SalePrice": test_predictions})
submission.to_csv("submission.csv", index=False)

print("Submission file saved as submission.csv")
```

HOW AND WHY:

This is for a kaggle competition House pricing its for prediction the house sale price for each house based on the dataset.It starts by loading the data and checking for missing values and then filling them by using me sales price is log-transformed for better modeling.

After that it splits the data into training and validation set and handles the missing values in categorical columns and encodes them similarly in both the training and test datasets.Then it trains the xg boost model to predict house prices.And in the end it preprocesses the test data makes predictions and creates a submission file with the predicted prices, saving it as submission.csv as you can see it in the last few lines.

Output:



Submission and Description | Public Score

submission.csv
Complete · now | 182906.49789

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	S
1455	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	NaN	0	8	2007	WD	
1456	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	MnPrv	NaN	0	2	2010	WD	
1457	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	GdPrv	Shed	2500	5	2010	WD	
1458	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	NaN	0	4	2010	WD	
1459	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	NaN	0	6	2008	WD	

5 rows × 80 columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 80 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   MSSubClass     1460 non-null   int64
 1   MSZoning       1460 non-null   object
 2   LotFrontage    1201 non-null   float64
 3   LotArea        1460 non-null   int64
 4   Street         1460 non-null   object
 5   Alley          91 non-null     object
 6   LotShape       1460 non-null   object
 7   LandContour    1460 non-null   object
 8   Utilities      1460 non-null   object
 9   LotConfig      1460 non-null   object
 10  LandSlope      1460 non-null   object
 11  Neighborhood   1460 non-null   object
 12  Condition1     1460 non-null   object
 13  Condition2     1460 non-null   object
 14  BldgType       1460 non-null   object
 15  HouseStyle     1460 non-null   object
 16  OverallQual    1460 non-null   int64
 17  OverallCond    1460 non-null   int64
 18  YearBuilt      1460 non-null   int64
 19  YearRemodAdd   1460 non-null   int64
...
 78  SaleCondition  1460 non-null   object
 79  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(34), object(43)
memory usage: 912.6+ KB
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Python

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	...	WoodDeckSF	OpenPorchSF	EnclosedPorch	3S
count	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1452.000000	1460.000000	1460.000000	...	1460.000000	1460.000000	1460.000000	146
mean	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.685262	443.639726	46.549315	...	94.244521	46.660274	21.954110	
std	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407	181.066207	456.098091	161.319273	...	125.338794	66.256028	61.119149	2
min	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	
25%	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	
50%	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	0.000000	383.500000	0.000000	...	0.000000	25.000000	0.000000	
75%	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	166.000000	712.250000	0.000000	...	168.000000	68.000000	0.000000	
max	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	1474.000000	...	857.000000	547.000000	552.000000	50

8 rows × 37 columns

```
print(df.count())
```

```
MSSubClass       1460
MSZoning         1460
LotFrontage      1201
LotArea          1460
Street           1460
                 ...
MoSold           1460
YrSold           1460
SaleType         1460
SaleCondition    1460
SalePrice        1460
Length: 80, dtype: int64
```

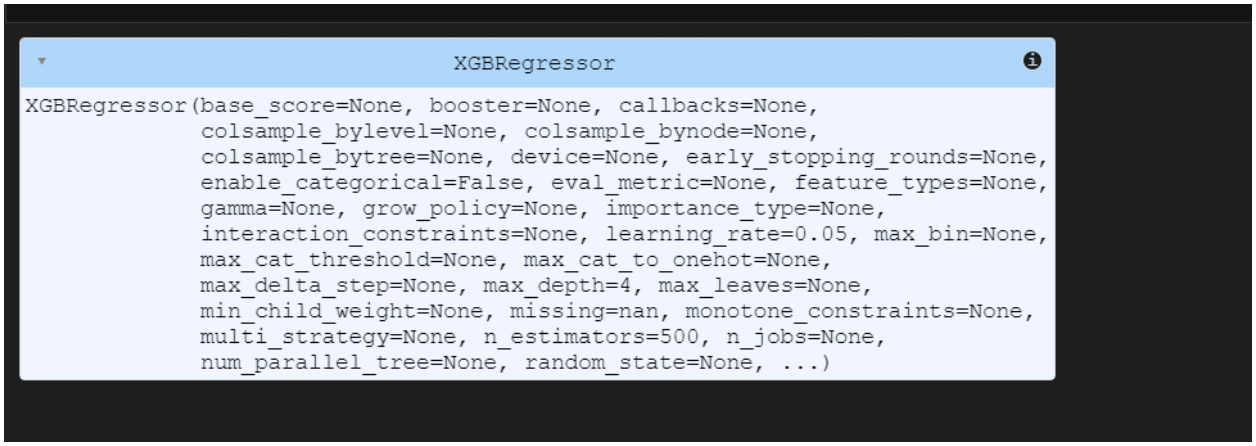
```
df.nunique()
```

```
MSSubClass         15
MSZoning            5
LotFrontage       110
LotArea          1073
Street              2
                  ...
MoSold             12
YrSold              5
SaleType            9
SaleCondition       6
SalePrice         663
Length: 80, dtype: int64
```

```
]
```

```
MSSubClass          0
MSZoning            0
LotFrontage       259
LotArea             0
Street              0
                  ...
MoSold              0
YrSold              0
SaleType            0
SaleCondition       0
SalePrice           0
Length: 80, dtype: int64
```

```
...  Index(['MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street', 'Alley',
       'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
       'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
       'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyle',
       'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea',
       'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating', 'HeatingQC',
       'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd',
       'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType', 'GarageYrBlt',
       'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual', 'GarageCond',
       'PavedDrive', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'PoolQC', 'Fence', 'MiscFeature', 'MiscVal',
       'MoSold', 'YrSold', 'SaleType', 'SaleCondition', 'SalePrice'],
      dtype='object')
    Index(['MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street', 'Alley',
       'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
       'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
       'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyle',
       'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea',
       'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating', 'HeatingQC',
    ...
       'PavedDrive', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'PoolQC', 'Fence', 'MiscFeature', 'MiscVal',
       'MoSold', 'YrSold', 'SaleType', 'SaleCondition'],
      dtype='object')
    Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```


XGBRegressor
XGBRegressor(base_score=None, booster=None, callbacks=None,
colsample_bylevel=None, colsample_bynode=None,
colsample_bytree=None, device=None, early_stopping_rounds=None,
enable_categorical=False, eval_metric=None, feature_types=None,
gamma=None, grow_policy=None, importance_type=None,
interaction_constraints=None, learning_rate=0.05, max_bin=None,
max_cat_threshold=None, max_cat_to_onehot=None,
max_delta_step=None, max_depth=4, max_leaves=None,
min_child_weight=None, missing=nan, monotone_constraints=None,
multi_strategy=None, n_estimators=500, n_jobs=None,
num_parallel_tree=None, random_state=None, ...)

```
MSSubClass         int64
MSZoning          object
LotFrontage      float64
LotArea            int64
Street            object
                  ...
MiscVal            int64
MoSold             int64
YrSold             int64
SaleType          object
SaleCondition     object
Length: 79, dtype: object
MSSubClass         int64
MSZoning           int32
LotFrontage      float64
LotArea            int64
Street             int32
                  ...
MiscVal            int64
MoSold             int64
YrSold             int64
SaleType           int32
SaleCondition      int32
Length: 79, dtype: object
Submission file saved as submission.csv
```