Task#01_ML_House_Prices_Prediction_using_Regression_Techniques .ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

Gemini

[2]
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd

# Load the training data
train_df = pd.read_csv('/content/drive/MyDrive/house-prices-advanced-regression-techniques/train.csv')

# Display the shape of the dataframe
print("Shape of the dataframe:", train_df.shape)

# Display the first 10 rows of the dataframe
print("First 10 rows of the dataframe:")
print(train_df.head(1461))

# Optionally, to display a random sample of 10 rows
print("Random sample of 1461 rows:")
print(train_df.sample(1460))

# Optionally, display specific columns
print("First 1461 rows of specific columns:")
print(train_df[['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath', 'SalePrice']].head(1461))
```

/content/drive/MyDrive/house-prices-advanced-regression-techniques/train.csv (c

```
[1460 rows x 81 columns]
Random sample of 1461 rows:
        Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
676    677          70       RM         60.0     9600   Pave  Grvl      Reg
761    762          30       RM         60.0     6911   Pave   NaN      Reg
767    768          50       RL         75.0    12508   Pave   NaN      IR1
530    531          80       RL         85.0    10200   Pave   NaN      Reg
650    651          60       FV         65.0     8125   Pave   NaN      Reg
...    ...         ...      ...          ...      ...    ...   ...      ...
995    996          50       RL         51.0     4712   Pave   NaN      IR1
620    621          30       RL         45.0     8248   Pave  Grvl      Reg
898    899          20       RL        100.0    12919   Pave   NaN      IR1
571    572          20       RL         60.0     7332   Pave   NaN      Reg
1344  1345          60       RL         85.0    11103   Pave   NaN      IR1

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature MiscVal  \
676          Lvl    AllPub  ...        0    NaN    NaN         NaN       0
761          Lvl    AllPub  ...        0    NaN    NaN         NaN       0
767          Lvl    AllPub  ...        0    NaN    NaN        Shed    1300
530          Lvl    AllPub  ...        0    NaN    NaN         NaN       0
650          Lvl    AllPub  ...        0    NaN    NaN         NaN       0
...          ...       ...  ...      ...    ...    ...         ...     ...
995          Lvl    AllPub  ...        0    NaN  MnPrv         NaN       0
620          Lvl    AllPub  ...        0    NaN    NaN         NaN       0
898          Lvl    AllPub  ...        0    NaN    NaN         NaN       0
571          Lvl    AllPub  ...        0    NaN    NaN         NaN       0
1344         Lvl    AllPub  ...        0    NaN    NaN         NaN       0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
676       5   2006        WD         Normal     87000
761      10   2009        WD         Normal    100000
767       7   2008        WD         Normal    160000
530       8   2008        WD        Abnorml    175000
650       5   2008        WD         Normal    205950
...     ...    ...       ...            ...       ...
995       8   2006        WD        Abnorml    121600
620       9   2008        WD         Normal     67000
898       3   2010       New        Partial    611657
571      10   2006        WD        Abnorml    120000
1344      7   2007       New        Partial    155835

[1460 rows x 81 columns]
First 1461 rows of specific columns:
      GrLivArea  BedroomAbvGr  FullBath  HalfBath  SalePrice
0          1710             3         2         1     208500
1          1262             3         2         0     181500
2          1786             3         2         1     223500
3          1717             3         1         0     140000
4          2198             4         2         1     250000
...         ...           ...       ...       ...        ...
1455       1647             3         2         1     175000
1456       2073             3         2         0     210000
1457       2340             4         2         0     266500
1458       1078             2         1         0     142125
1459       1256             3         1         1     147500

[1460 rows x 5 columns]
```

[6]
```
# Select relevant columns
columns = ['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath', 'SalePrice']
df = train_df[columns]

# Check for missing values
```

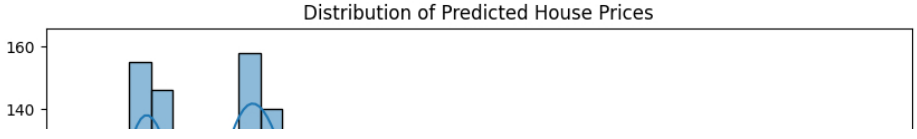
```
# Check for missing values
missing_values = df.isnull().sum()

missing_values
```

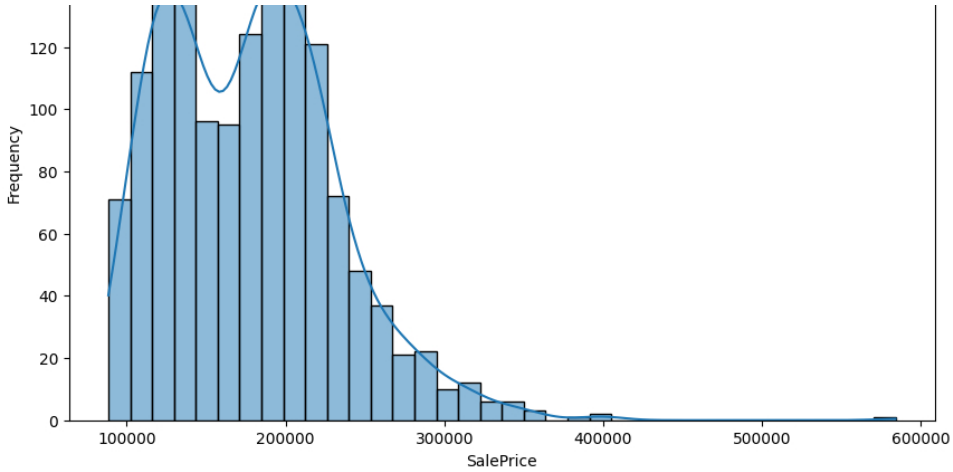
```
GrLivArea       0
BedroomAbvGr    0
FullBath        0
HalfBath        0
SalePrice       0
dtype: int64
```

[7]
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error

# Split the data into training and validation sets
X = df[['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath']]
y = df['SalePrice']
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict on the validation set
y_pred = model.predict(X_val)

# Evaluate the model
mae = mean_absolute_error(y_val, y_pred)

mae
```

```
36018.563138363446
```

[9]
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Load the training data
train_df = pd.read_csv('/content/drive/MyDrive/house-prices-advanced-regression-techniques/train.csv')

# Select relevant columns for training
columns = ['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath', 'SalePrice']
df = train_df[columns]

# Split the data into training and validation sets
X = df[['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath']]
y = df['SalePrice']
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Load the test data
test_df = pd.read_csv('/content/drive/MyDrive/house-prices-advanced-regression-techniques/test.csv')

# Select relevant columns for prediction
X_test = test_df[['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath']]

# Make predictions
test_pred = model.predict(X_test)

# Prepare the submission file
submission = pd.DataFrame({
    'Id': test_df['Id'],
    'SalePrice': test_pred
})

# Save the submission file
submission.to_csv('/content/drive/MyDrive/house-prices-advanced-regression-techniques/sample_submission.csv', index=False)

# Plot the distribution of predicted house prices
plt.figure(figsize=(10, 6))
sns.histplot(test_pred, kde=True)
plt.title('Distribution of Predicted House Prices')
plt.xlabel('SalePrice')
plt.ylabel('Frequency')
plt.show()

# You cannot plot predicted vs. actual prices because the test set does not contain actual prices.
# If you want to evaluate your model's performance, you should use the validation set (X_val, y_val) instead.
```






```python
import pandas as pd
from sklearn.linear_model import LinearRegression

# Load the test data
test_df = pd.read_csv('/content/drive/MyDrive/house-prices-advanced-regression-techniques/test.csv')

# Select relevant columns for prediction
X_test = test_df[['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath']]

# Assuming you have trained your model already
model = LinearRegression()
model.fit(X_train, y_train)  # Ensure X_train and y_train are defined as shown earlier

# Make predictions
test_pred = model.predict(X_test)

# Prepare the submission file
submission = pd.DataFrame({
    'Id': test_df['Id'],
    'SalePrice': test_pred
})

# Save the submission file
submission.to_csv('/content/drive/MyDrive/house-prices-advanced-regression-techniques/sample_submission.csv', index=False)

# Display the saved submission file
saved_submission = pd.read_csv('/content/drive/MyDrive/house-prices-advanced-regression-techniques/sample_submission.csv')
print(saved_submission.head())
```

```
<function read_csv at 0x78776ea94820>
     Id      SalePrice
0  1461  121423.030985
1  1462  143380.870622
2  1463  204748.668874
3  1464  202205.354725
4  1465  191336.364775
```

```python
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Load the training data
train_df = pd.read_csv('/content/drive/MyDrive/house-prices-advanced-regression-techniques/train.csv')

# Select relevant columns
columns = ['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath', 'SalePrice']
df = train_df[columns]

# Split the data into training and validation sets
X = df[['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath']]
y = df['SalePrice']
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Load the test data
test_df = pd.read_csv('/content/drive/MyDrive/house-prices-advanced-regression-techniques/test.csv')

# Select relevant columns for prediction
X_test = test_df[['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath']]

# Make predictions
test_pred = model.predict(X_test)

# Prepare the submission file
submission = pd.DataFrame({
    'Id': test_df['Id'],
    'SalePrice': test_pred
})
```

```
})

# Save the submission file
submission_file_path = '/content/drive/MyDrive/house-prices-advanced-regression-techniques/sample_submission.csv'
submission.to_csv(submission_file_path, index=False)

# Display the saved submission file
saved_submission = pd.read_csv('/content/drive/MyDrive/house-prices-advanced-regression-techniques/sample_submission.csv')

# Print the shape to confirm the number of rows and columns
print("Shape of the submission file:", saved_submission.shape)

# Display the first few rows to verify
print(saved_submission.head(1444))
```

```
Shape of the submission file: (1459, 2)
        Id      SalePrice
0     1461  121423.030985
1     1462  143380.870622
2     1463  204748.668874
3     1464  202205.354725
4     1465  191336.364775
...    ...           ...
1439  2900  157511.070484
1440  2901  198129.385699
1441  2902  203849.470385
1442  2903  233810.494934
1443  2904  221399.121890

[1444 rows x 2 columns]
```