Title here (i.e., House Price Prediction)

```
# Edit all the Mardown cells below with the appropriate information
# Run all cells, containing your code
# Save this Jupyter with the outputs of your executed cells
# PS: Save again the notebook with this outcome.
# PSPS: Don't forget to include the dataset in your submission
```

Team:

- Anthony Lam
- FirstName LastName 2
- FirstName LastName 2

Course: CISD 43 – BIG DATA (Spring, 2024)

Problem Statement

- This project is about house price predictions.
- **Keywords:** House price prediction, real estate ,...,

Required packages

- Add instructions to install the required packages

```
## Your code begins here
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

Methodology

1. Explan your big data metodology
2. Introduce the topics you used in your project

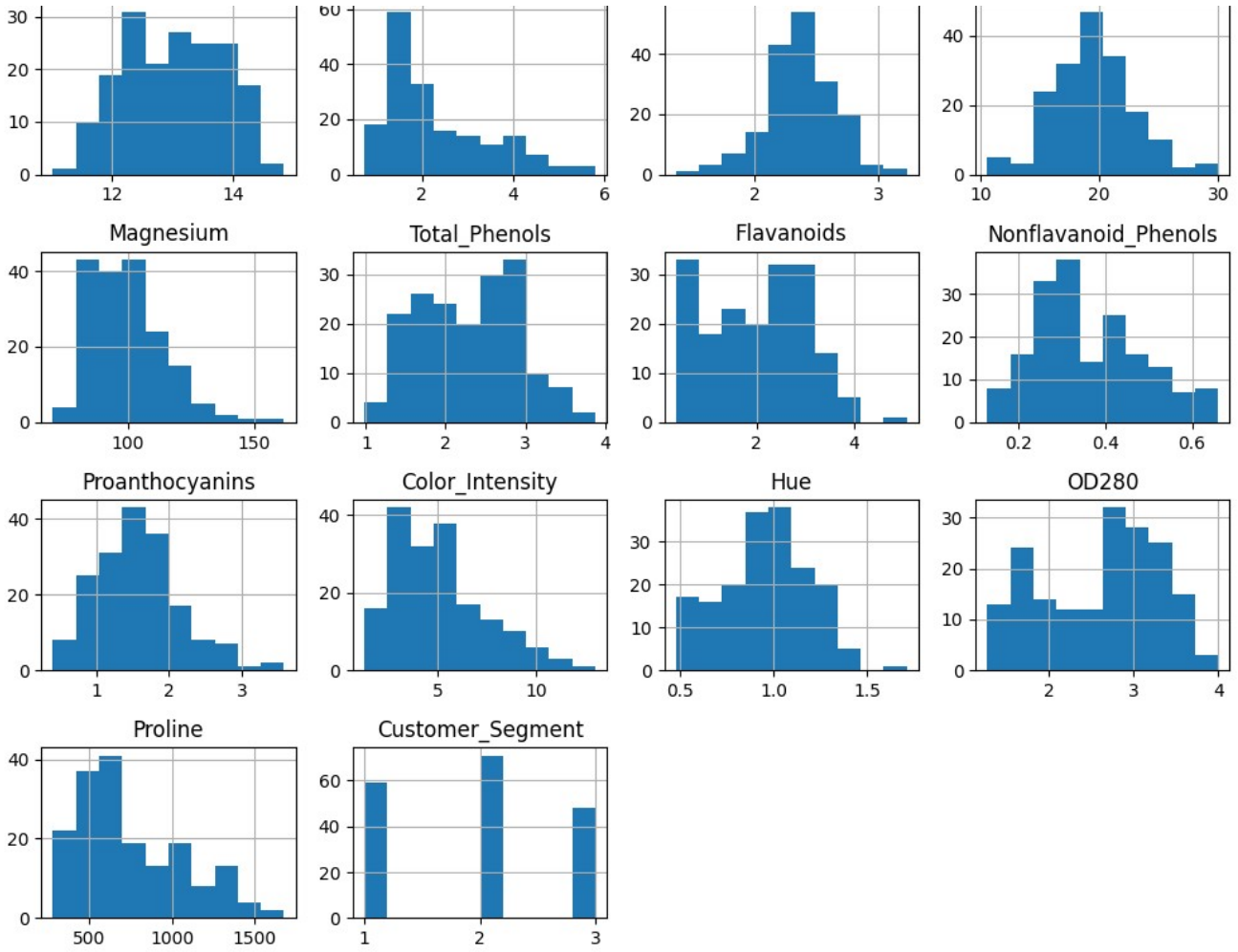
- Model 1
  - KNN
- Model 2
  - Linear Regression

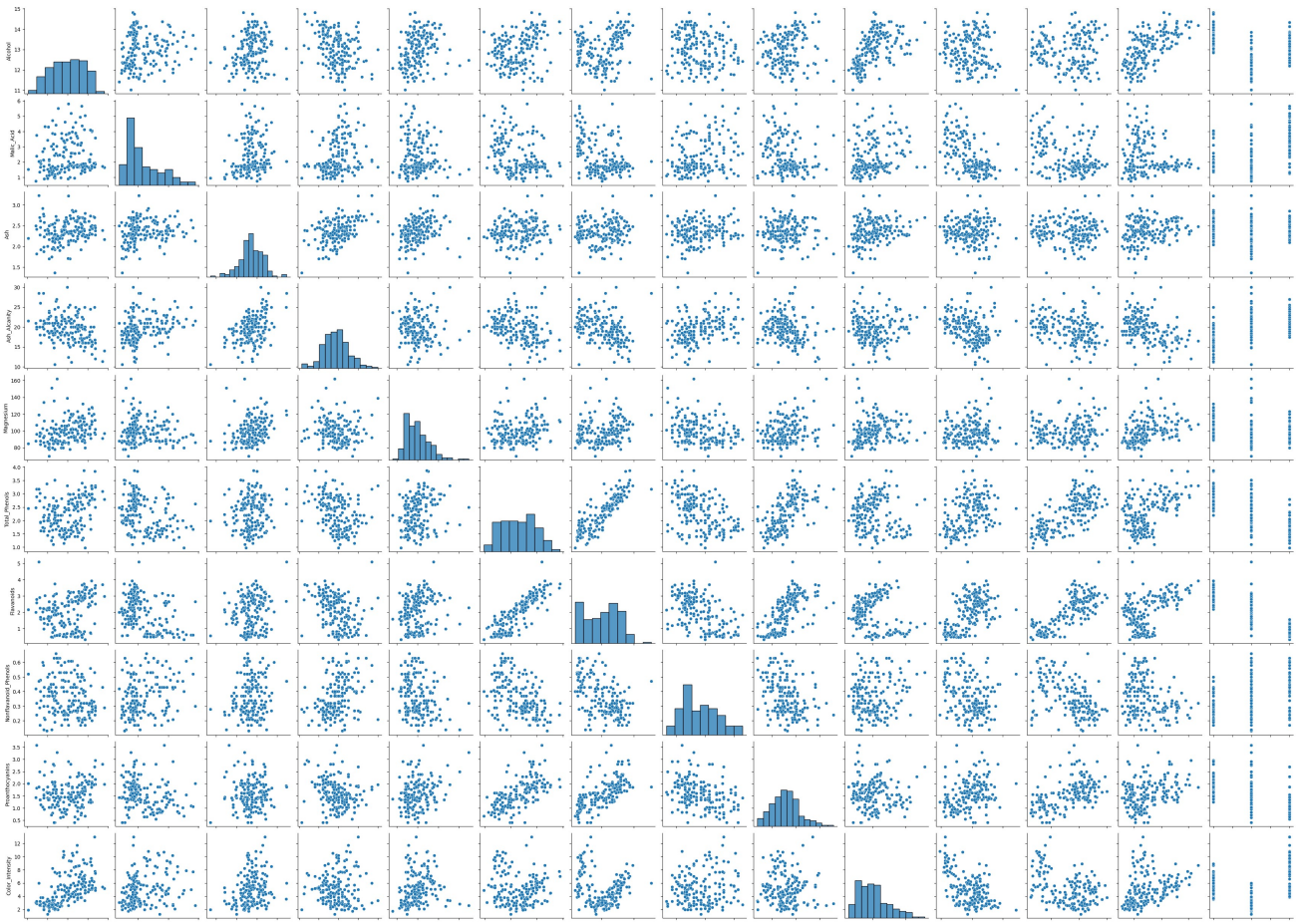
Your code starts here

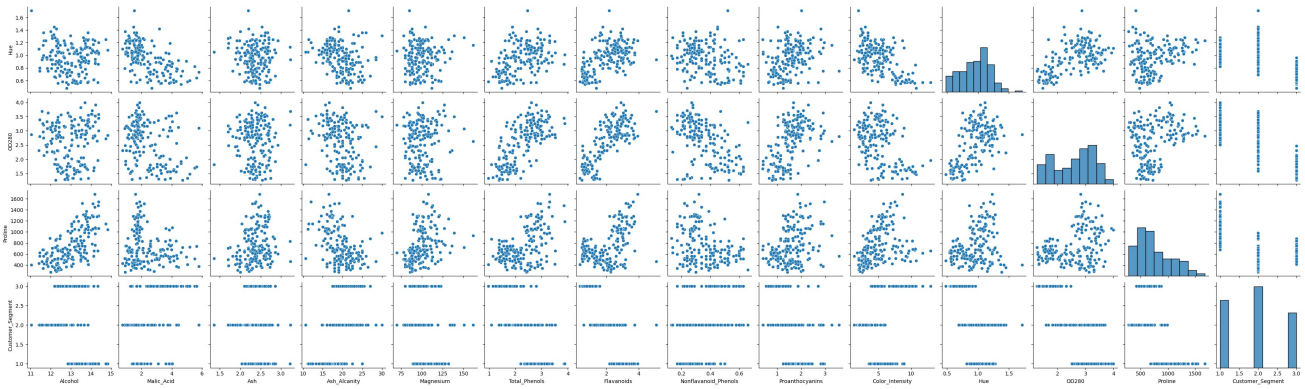
```
df = pd.read_csv('wine.csv')


df.dropna(inplace=True)

df.drop_duplicates(inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Alcohol               178 non-null    float64
 1   Malic_Acid            178 non-null    float64
 2   Ash                   178 non-null    float64
 3   Ash_Alcanity          178 non-null    float64
 4   Magnesium             178 non-null    int64
 5   Total_Phenols         178 non-null    float64
 6   Flavanoids            178 non-null    float64
 7   Nonflavanoid_Phenols  178 non-null    float64
 8   Proanthocyanins       178 non-null    float64
 9   Color_Intensity       178 non-null    float64
 10  Hue                   178 non-null    float64
 11  OD280                 178 non-null    float64
 12  Proline               178 non-null    int64
 13  Customer_Segment      178 non-null    int64
dtypes: float64(11), int64(3)
memory usage: 19.6 KB
```

```
# Histograms of numerical features
df.hist(figsize=(10, 8))
plt.tight_layout()
plt.show()

# Pairplot to visualize relationships between numerical features
sns.pairplot(df)
plt.show()
```


Magnesium
Total_Phenols
Flavanoids
Nonflavanoid_Phenols
Proanthocyanins
Color_Intensity
Hue
OD280
Proline
Customer_Segment




Hue
OD280
Proline
Customer_Segment
Alcohol
Malic_Acid
Ash
Ash_Alcanity
Magnesium
Total_Phenols
Flavanoids
Nonflavanoid_Phenols
Proanthocyanins
Color_Intensity
Hue
OD280
Proline
Customer_Segment

Generate Close

1 of 1 Undo Changes Use code with caution

```
X = df[['Total_Phenols']]
y = df['Flavanoids']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


knn_regressor = KNeighborsRegressor(n_neighbors=5)
knn_regressor.fit(X_train, y_train)
```

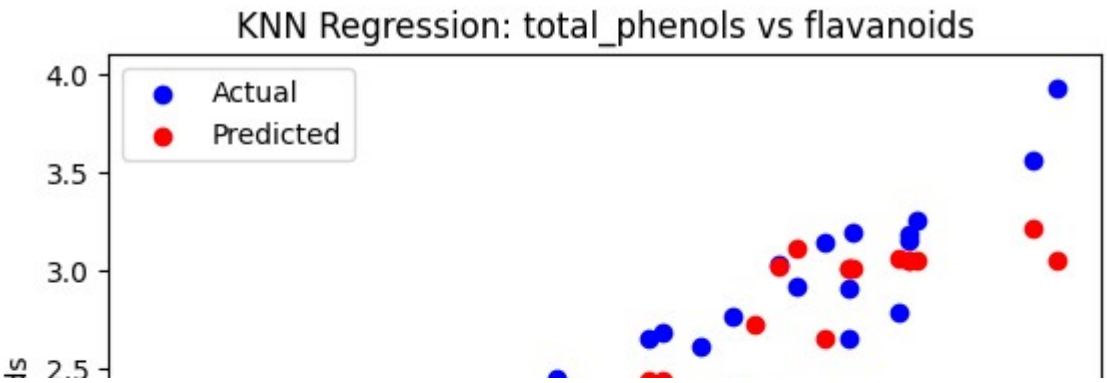
▾ KNeighborsRegressor

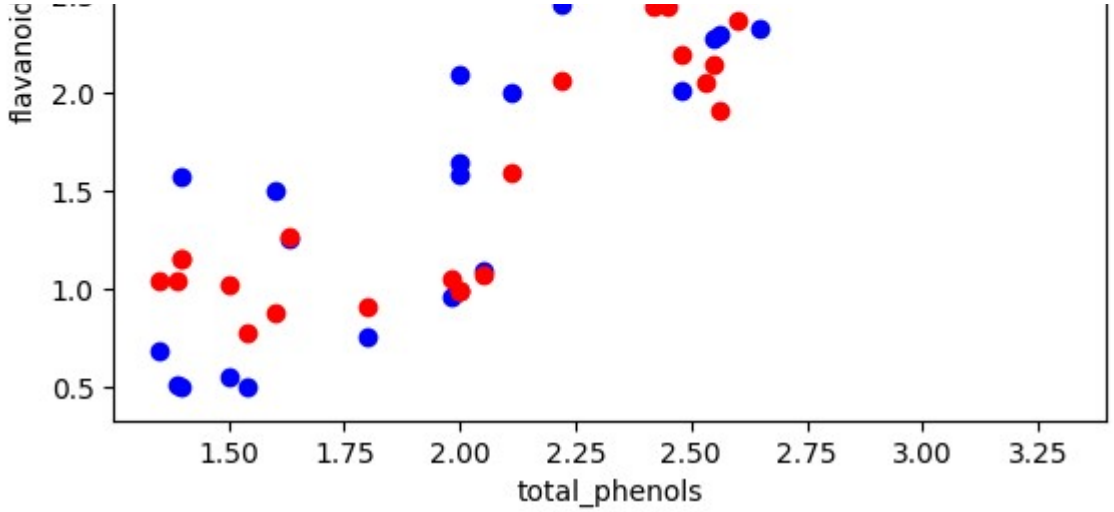
```
KNeighborsRegressor()
```

```
y_pred = knn_regressor.predict(X_test)


mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
```

```
Mean Squared Error: 0.17833688888888888
R-squared: 0.8106158639155603
```

```
# Visualization of Predictions
plt.scatter(X_test, y_test, color='blue', label='Actual')
plt.scatter(X_test, y_pred, color='red', label='Predicted')
plt.xlabel('total_phenols')
plt.ylabel('flavanoids')
plt.title('KNN Regression: total_phenols vs flavanoids')
plt.legend()
plt.show()
```





Generate Close

1 of 1 Undo Changes Use code with caution

Generated code may be subject to a license | Fouziya15/Linear-regression-to-predict-housing-prices | Michall495002/Personal-Chall

```
linear_regressor = LinearRegression()
linear_regressor.fit(X_train, y_train)
```

▾ LinearRegression i ?

LinearRegression()

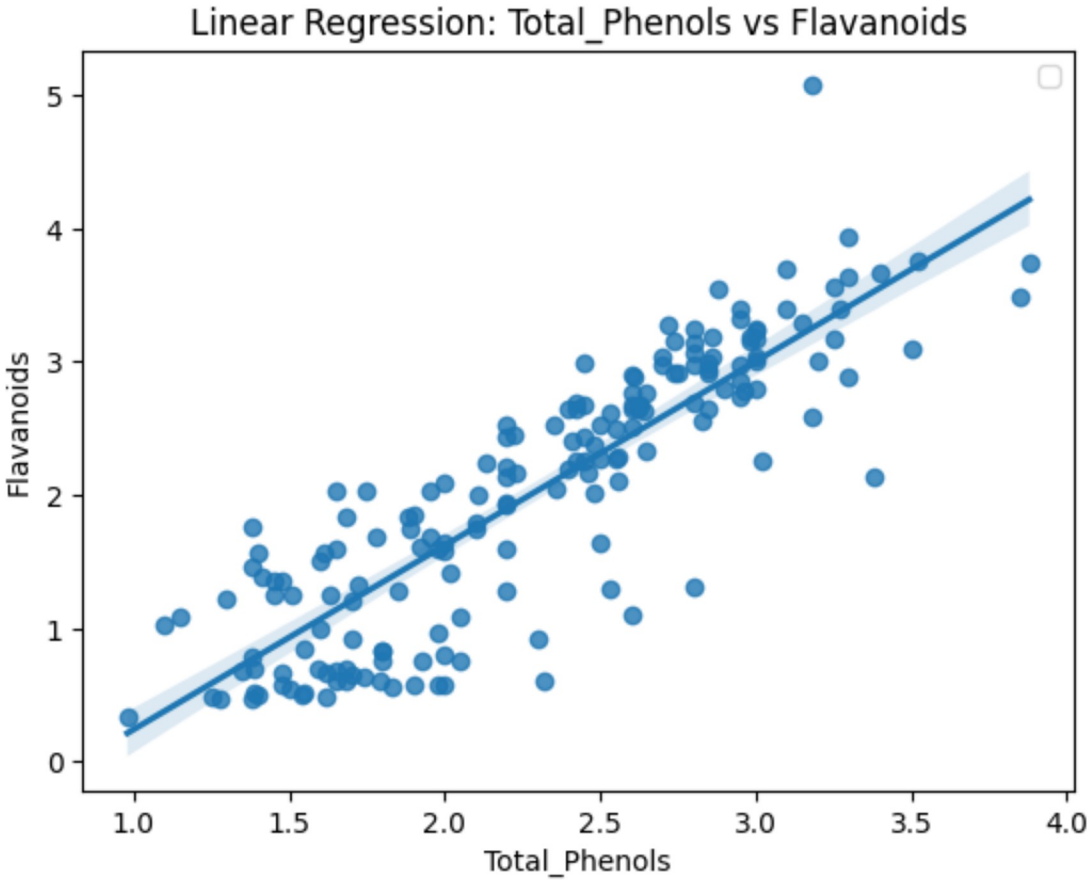
```
y_pred_linear = linear_regressor.predict(X_test)


mse_linear = mean_squared_error(y_test, y_pred_linear)
r2_linear = r2_score(y_test, y_pred_linear)

print(f"Linear Regression Mean Squared Error: {mse_linear}")
print(f"Linear Regression R-squared: {r2_linear}")
```

```
Linear Regression Mean Squared Error: 0.14295664791546014
Linear Regression R-squared: 0.8481877673672726
```

```
# Visualization of Predictions
sns.regplot(x='Total_Phenols', y='Flavanoids', data=df)
plt.xlabel('Total_Phenols')
plt.ylabel('Flavanoids')
plt.title('Linear Regression: Total_Phenols vs Flavanoids')
plt.legend()
plt.show()
```

```
<ipython-input-30-2dda86489ab2>:6: UserWarning: No artists with labels found to put i
  plt.legend()
```

```
plt.legend()
```



Conclusions

```
# Linear Regression seems to be performing better in terms of MSE and R^2
```

References

- Academic (if any)
- Online (if any)

```
Start coding or generate with AI.
```

Credits

- If you use and/or adapt your code from existing projects, you must provide links and acknowldge the authors.

> *This code is based on .... (if any)*

Start coding or generate with AI.

End of Project