Capstone Project Summary Report

Sanchaita Biswas

Capstone Project - 1: Predicting Property Prices in a Specific Location Using Machine Learning

1. Introduction to the Project:

The real estate market is influenced by numerous factors, making property price prediction a complex yet crucial task for various stakeholders. This capstone project uses machine learning techniques to predict property prices in a specific location. The project aims to provide valuable insights for real estate agents, buyers, and sellers by utilizing advanced data analysis and modeling.

2. Objectives of the Project:

1. Collect and clean real estate data from a specific location.

2. Handle ordinal and nominal columns separately to enhance prediction accuracy.

3. Implement scaling, PCA, and fillna() techniques to handle missing data effectively.

4. Perform exploratory data analysis (EDA) to identify key variables influencing property prices.

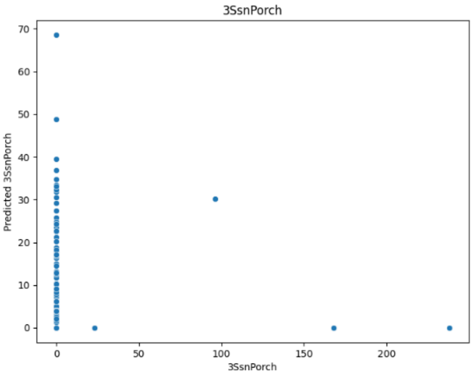
5. Determine appropriate encoding techniques for ordinal and nominal variables based on model requirements.

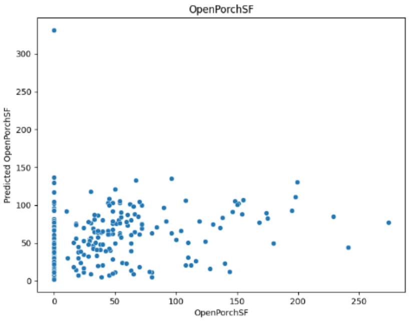
6. Develop a machine learning model capable of accurately predicting property prices.

7. Evaluate model performance and compare with alternative algorithms.

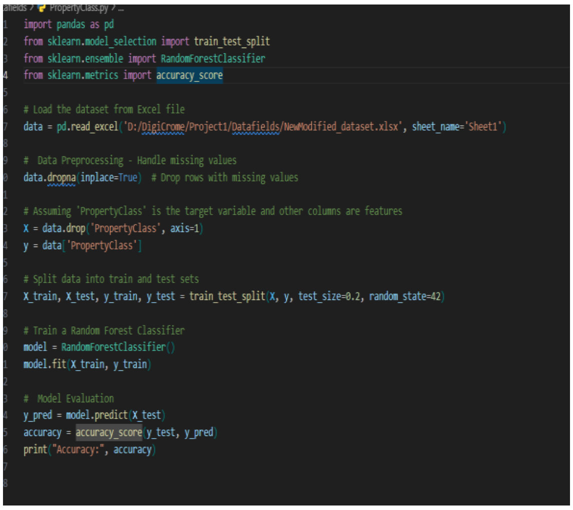
8. Present project findings and insights clearly and concisely.

3. Flow Chart of Operations:

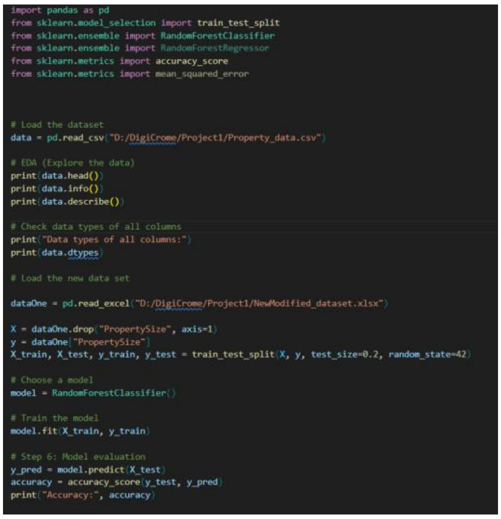




4. Python Codes:

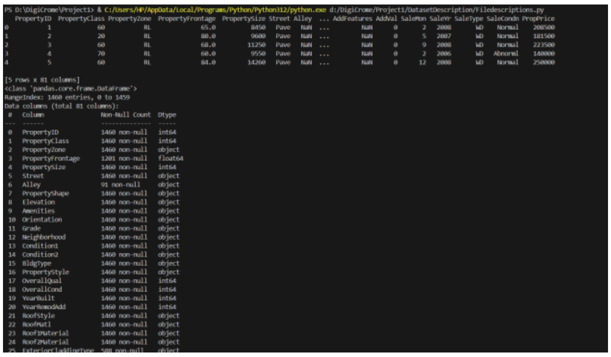


```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Load the dataset from Excel file
data = pd.read_excel('D:/DigiCrome/Project1/Datafields/NewModified_dataset.xlsx', sheet_name='Sheet1')

#  Data Preprocessing - Handle missing values
data.dropna(inplace=True)  # Drop rows with missing values

# Assuming 'PropertyClass' is the target variable and other columns are features
X = data.drop('PropertyClass', axis=1)
y = data['PropertyClass']

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Random Forest Classifier
model = RandomForestClassifier()
model.fit(X_train, y_train)

#  Model Evaluation
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

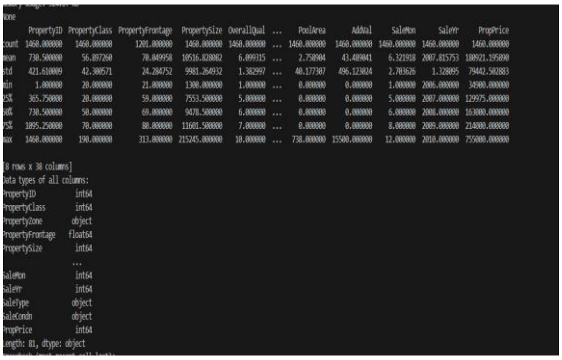


```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error



# Load the dataset
data = pd.read_csv("D:/DigiCrome/Project1/Property_data.csv")

# EDA (Explore the data)
print(data.head())
print(data.info())
print(data.describe())

# Check data types of all columns
print("Data types of all columns:")
print(data.dtypes)

# Load the new data set

dataOne = pd.read_excel("D:/DigiCrome/Project1/NewModified_dataset.xlsx")

X = dataOne.drop("PropertySize", axis=1)
y = dataOne["PropertySize"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Choose a model
model = RandomForestClassifier()

# Train the model
model.fit(X_train, y_train)

# Step 6: Model evaluation
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

5. Screenshot of the Outputs:



```
PS D:\DigiCrome\Project1> & C:/Users/HP/AppData/Local/Programs/Python/Python312/python.exe d:/DigiCrome/Project1/DatasetDescription/Filedescriptions.py
   PropertyID  PropertyClass PropertyZone  PropertyFrontage  PropertySize Street Alley  ... AddFeatures AddVal SaleMon SaleYr SaleType SaleCondn PropPrice
0           1             60           RL              65.0          8450   Pave   NaN  ...         NaN      0       2   2008       WD    Normal    208500
1           2             20           RL              80.0          9600   Pave   NaN  ...         NaN      0       5   2007       WD    Normal    181500
2           3             60           RL              68.0         11250   Pave   NaN  ...         NaN      0       9   2008       WD    Normal    223500
3           4             70           RL              60.0          9550   Pave   NaN  ...         NaN      0       2   2006       WD   Abnorml    140000
4           5             60           RL              84.0         14260   Pave   NaN  ...         NaN      0      12   2008       WD    Normal    250000

[5 rows x 81 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   PropertyID            1460 non-null   int64
 1   PropertyClass         1460 non-null   int64
 2   PropertyZone          1460 non-null   object
 3   PropertyFrontage      1201 non-null   float64
 4   PropertySize          1460 non-null   int64
 5   Street                1460 non-null   object
 6   Alley                 91 non-null     object
 7   PropertyShape         1460 non-null   object
 8   Elevation             1460 non-null   object
 9   Amenities             1460 non-null   object
 10  Orientation           1460 non-null   object
 11  Grade                 1460 non-null   object
 12  Neighborhood          1460 non-null   object
 13  Condition1            1460 non-null   object
 14  Condition2            1460 non-null   object
 15  BldgType              1460 non-null   object
 16  PropertyStyle         1460 non-null   object
 17  OverallQual           1460 non-null   int64
 18  OverallCond           1460 non-null   int64
 19  YearBuilt             1460 non-null   int64
 20  YearRemodAdd          1460 non-null   int64
 21  RoofStyle             1460 non-null   object
 22  RoofMatl              1460 non-null   object
 23  Roof1Material         1460 non-null   object
 24  Roof2Material         1460 non-null   object
 25  ExteriorCladdingType  588 non-null    object
```



```
None
       PropertyID  PropertyClass  PropertyFrontage   PropertySize  OverallQual  ...     PoolArea        AddVal      SaleMon       SaleYr      PropPrice
count  1460.000000    1460.000000       1201.000000    1460.000000  1460.000000  ...  1460.000000   1460.000000  1460.000000  1460.000000    1460.000000
mean    730.500000      56.897260         70.049958   10516.828082     6.099315  ...     2.758904     43.489041     6.321918  2007.815753  180921.195890
std     421.610009      42.300571         24.284752    9981.264932     1.382997  ...    40.177307    496.123024     2.703626     1.328095   79442.502883
min       1.000000      20.000000         21.000000    1300.000000     1.000000  ...     0.000000      0.000000     1.000000  2006.000000   34900.000000
25%     365.750000      20.000000         59.000000    7553.500000     5.000000  ...     0.000000      0.000000     5.000000  2007.000000  129975.000000
50%     730.500000      50.000000         69.000000    9478.500000     6.000000  ...     0.000000      0.000000     6.000000  2008.000000  163000.000000
75%    1095.250000      70.000000         80.000000   11601.500000     7.000000  ...     0.000000      0.000000     8.000000  2009.000000  214000.000000
max    1460.000000     190.000000        313.000000  215245.000000    10.000000  ...   738.000000  15500.000000    12.000000  2010.000000  755000.000000

[8 rows x 38 columns]
Data types of all columns:
PropertyID            int64
PropertyClass         int64
PropertyZone         object
PropertyFrontage    float64
PropertySize          int64
                     ...
SaleMon               int64
SaleYr                int64
SaleType             object
SaleCondn            object
PropPrice             int64
Length: 81, dtype: object
Traceback (most recent call last):
```

6. Report on EDA:

Exploratory Data Analysis (EDA) is crucial in understanding the underlying patterns and relationships within the dataset, providing valuable insights into the factors influencing property prices. This section presents the key findings from our EDA process, accompanied by graphs and visualizations.

1. Distribution of Property Prices:

- **Histogram**: The distribution of property prices reveals whether they are normally distributed or skewed.

- **Boxplot:** Identifying outliers and understanding the spread of property prices across different quartiles.

2. Correlation Analysis:

- **Heatmap:** Visualizing the correlation matrix to identify features strongly correlated with property prices.

- **Scatterplots:** Exploring relationships between property prices and other numerical variables, such as property size, number of bedrooms, and overall quality.

3. Categorical Variables Analysis:

- **Bar Charts:** Examining the distribution of categorical variables like property class, zoning classification, and neighborhood about property prices.
- **Boxplots:** Comparing property prices across different categories of categorical variables to identify any significant differences.

4. Time Trends Analysis:

- **Line Plot:** Analyzing trends in property prices over time (years old) to understand market dynamics and seasonality effects.

5. Feature Engineering Insights:

- **Derived Features:** Exploring newly created features based on domain knowledge or statistical techniques and their relationship with property prices.

Key Insights:

1. **Property Size:** Larger properties tend to have higher prices, indicating a positive correlation between property size and price.

2. **Neighbourhood**: Certain neighborhoods exhibit higher property prices compared to others, highlighting the importance of location in determining property values.

3. **Overall Quality**: Properties with higher overall quality ratings command higher prices, emphasizing the significance of property condition.

4. **Time Trends**: Property prices may exhibit temporal trends, with fluctuations influenced by economic factors and market conditions.

5. **Categorical Variables**: Certain property classes, zoning classifications, and amenities may significantly impact property prices, warranting further investigation.

Recommendations:

1. Focus on properties in neighbourhoods with historically high price premiums.

2. Prioritize properties with larger sizes and superior quality ratings for potentially higher returns.

3. Consider the impact of time trends on property prices when making investment decisions.

4. Investigate the influence of specific categorical variables on property prices to tailor marketing strategies accordingly.

In conclusion, the EDA process has provided valuable insights into the determinants of property prices, laying the groundwork for developing an accurate predictive model. Further analysis and feature selection based on these insights will enhance the model's performance and predictive capability.

7. Learning Outcomes:

- Proficiency in data collection, cleaning, and preprocessing techniques.

- Understanding of feature engineering methods to enhance model performance.

- Familiarity with various machine learning algorithms and their application in property price prediction.

- Skills in model evaluation and comparison to determine the best-performing approach.

- Experience in presenting project findings and insights effectively.

8. Conclusion:

This capstone project endeavours to develop a robust machine-learning model for predicting property prices in a specific location. By following a structured methodology encompassing data collection, cleaning, analysis, modeling, and evaluation, the project aims to deliver accurate predictions and valuable insights for real estate stakeholders.

9. Citations:

- Dean De Cock. (n.d.). Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project. Retrieved from http://jse.amstat.org/v19n3/decock.pdf

- Scikit-learn: Machine Learning in Python. (n.d.). Retrieved from https://scikit-learn.org/stable/