Weekly Report for CSE 6940 (Graduate Research Methods in Computer Science)  
By: Nikitha Kondreddy Venkatramireddy  
Date: October 2, 2024

**House Price Prediction using Machine Learning in Python**

**Objective:**

The goal of this project is to predict house prices using machine learning techniques. This week's focus was on completing the data preprocessing pipeline, feature engineering, and initiating basic machine learning model development.

1. **Data Preprocessing:**

**1.1 Handling Missing Data:**

As discussed in the previous report, several columns contained missing values. This week, I applied imputation strategies to fill these missing values efficiently.

* **Numerical Features:**

For features like `LotFrontage`, missing values were replaced using the **median** value of the respective column. The reason for using the median is its robustness against outliers.

A black and white text on a black background

Description automatically generated

**Output:**

Before and after imputation, the count of missing values in `LotFrontage` can be visualized:

A black and white text on a black background

Description automatically generated

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

* **Categorical Features:**

For features like `GarageType`, the missing values were filled with the most frequent category (mode):

A black and white text on a black background

Description automatically generated

**Output:**

Frequency counts of `GarageType` before and after imputation can be compared:

A black and white screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

**1.2 Encoding Categorical Variables:**

Machine learning algorithms work with numerical data. I applied **One-Hot Encoding** to convert categorical columns into numerical format. This process expands categorical columns like `GarageType` into multiple binary columns (0 or 1).

A screen shot of a computer code

Description automatically generated

Output:

A screenshot of a computer

Description automatically generated

**Visualization:**

You can visualize the encoding impact on specific categorical features using a \*\*heatmap\*\* to show how many binary variables have been created.

A screen shot of a computer code

Description automatically generated

A graph of a number of words

Description automatically generated with medium confidence

**2. Feature Scaling:**

Scaling was applied to ensure that features are on the same scale, which is particularly crucial for distance-based models and models like **Linear Regression**.

**Standar**dScaler was used for scaling numerical columns:

A screen shot of a computer

Description automatically generated

**Output:**

A screenshot of a graph

Description automatically generated

**Visualization of Scaled Features:**

A computer screen with text

Description automatically generated

A graph of a normal distribution

Description automatically generated

**3.** **Building and Testing the Linear Regression Model**:

**3.1 Data Splitting:**

The dataset was split into training and testing sets (80% training, 20% testing) to allow proper evaluation of model performance.

A screenshot of a computer program

Description automatically generated

**Output:**

A close up of a sign

Description automatically generated

**3.2 Model Training:**

A Linear Regression model was built to serve as the baseline model.

A black screen with white text

Description automatically generated

**Output:**

**A close up of numbers

Description automatically generated**

**3.3 Model Evaluation:**

Predictions were made on the test set, and I evaluated the model using **Mean Squared Error (MSE)** and **R-squared**:

A screen shot of a computer program

Description automatically generated

**Output:**

A close-up of numbers

Description automatically generated

**Visualization of Predictions:**

A computer screen shot of a program code

Description automatically generated

This plot demonstrates how closely the predictions align with actual house prices.

**A graph showing a blue dotted line

Description automatically generated with medium confidence**

**Challenges Faced:**

**Handling Multicollinearity:**

Some of the features were highly correlated, which could affect model performance. I plan to explore dimensionality reduction techniques, such as Principal Component Analysis (PCA), to address this.

**Next Week's Plan:**

**Refining the Feature Selection Process**:

Experiment with removing features with high multicollinearity.

**Exploring Advanced Models**:

Test more complex models like **Random Forest** and **XGBoost**.

**Hyperparameter Tuning:**

Use grid search or random search to find the best hyperparameters for improving model performance.

**Evaluate Models:**

Focus on model comparison and use additional metrics such as **Root Mean Squared Error (RMSE)** and **Adjusted R-squared**.

**Github Link:**

* [**https://github.com/Nikitha130731/House-Price-Prediction-using-ML/tree/main**](https://github.com/Nikitha130731/House-Price-Prediction-using-ML/tree/main)

**References:**

* <https://www.kaggle.com/competitions/home-data-for-ml-course/data>
* <https://pandas.pydata.org/docs/user_guide/index.html>
* <https://www.w3schools.com/python/python_ml_getting_started.asp>
* <https://scikit-learn.org/stable/>