**PSL Project1:**

In order to predict the housing prices in Ames, I built two predictive models, utilizing XGBoost and Elastic Net respectively.

**Technical Details**:

**Model 1:** XGBoost

As for the XGBoost model, I used all variables except 'PID'. After dropping 'PID', X_train has 81 variables in total, in which 35 variables are numerical and 46 are categorical.

As for numerical variables, the only pre-processing step I did is filling the missing values in the 'Garage_Yr_Blt' column with 0. As for Categorical variables, I used OneHotEncoder for one-hot encoding to transfer categorical values into numerical values. I set the parameter handle_unknown='ignore' to ignore new levels present in the X_test that were not seen during the training process. After these pre-processing steps, X_train has a shape of (2051, 346).

Then I utilized XGBoost for regression modeling. Specifically, I employed the XGBRegressor package in python and set the parameters as: max_depth=6, eta=0.05, n_estimators=5000, subsample=0.6.

**Model 2:** Elastic Net

As for the Elastic Net model, I removed following variables:
'PID', 'Street', 'Utilities', 'Condition_2', 'Roof_Matl', 'Heating', 'Pool_QC', 'Misc_Feature', 'Low_Qual_Fin_SF', 'Pool_Area', 'Longitude', 'Latitude'

After dropping, X_train has 70 variables in total, in which 31 variables are numerical and 39 are categorical.

As for numerical variables, I filled the missing values in the 'Garage_Yr_Blt' column with 0. I winsorized the following numerical variables by calculating the 95th percentile (referred to as M) based on the training data. Any value in both the training and test datasets that exceeds M is then replaced with M. The winsorized variables includes:
"Lot_Frontage", "Lot_Area", "Mas_Vnr_Area", "BsmtFin_SF_2", "Bsmt_Unf_SF", "Total_Bsmt_SF", "Second_Flr_SF", 'First_Flr_SF', "Gr_Liv_Area", "Garage_Area", "Wood_Deck_SF", "Open_Porch_SF", "Enclosed_Porch", "Three_season_porch", "Screen_Porch", "Misc_Val"

As for Categorical variables, I used OneHotEncoder for one-hot encoding to transfer categorical values into numerical values. I set the parameter handle_unknown='ignore' to ignore new levels present in the X_test that were not seen during the training process. After these pre-processing steps, X_train has a shape of (2051, 309).

Then I utilized Elastic Net for regression modeling. Specifically, I employed the ElasticNet package in python and set the parameters as: alpha=0.001, l1_ratio=0.1, max_iter=10000.

**Performance Metrics**:

The computer system used is: MacBook Pro 13.3" Laptop - Apple M2 chip - 24GB Memory - 1TB SSD (Latest Model) - Silver.

The performance and the execution time (including pre-processing and evaluation) on the 10 splits are as follows:

**Model 1: XGBoost**

| root_mean_squared_error | execution time |
|---|---|
| 0.11779035873404123 | 4.96525502204895 |
| 0.12066295714352665 | 5.520492076873779 |
| 0.115153310738678 | 4.629983186721802 |
| 0.11728299814668453 | 4.441195964813232 |
| 0.11356525326153337 | 4.116149187088013 |
| 0.1270769902491383 | 5.2610907554626465 |
| 0.13399286588561624 | 4.702088117599487 |
| 0.12740473127481133 | 4.386313438415527 |
| 0.1331163714093709 | 4.922964811325073 |
| 0.11957428474335385 | 4.3986499309539795 |

**Model 2: Elastic Net**

| | |
|---|---|
| 0.12062059672817009 | 0.8061611652374268 |
| 0.1157528153658965 | 0.4147939682006836 |
| 0.11104198238335992 | 0.951470136642456 |
| 0.11516257752812245 | 0.43228912353515625 |
| 0.10857993553821996 | 0.5212709903717041 |
| 0.1341710879790037 | 0.8441569805145264 |
| 0.1323482635513255 | 0.38021397590637207 |
| 0.12470938493460917 | 0.5832960605621338 |
| 0.12984569028446144 | 0.31334805488586426 |
| 0.12284628665471527 | 0.3178989887237549 |