COMP309/AIML421 – Machine Learning Tools and Techniques
Assignment 2: Data Exploration, Manipulation and Modeling
greenthom – 300536064

Part 1: Business and Data Understanding

1a.

Using numpy and pandas variables (shape, selectdtypes()), identified that there were 1460 instances and 80(79) features. Moved 'MSSubClass', 'OverallQual', and 'OverallCond' from numerical features due to them being categorical (rating features). This produced, of the 80 (79) features identified, 46 categorical features and 34 numerical. This is on the exclusion of 'id' from numerical features, as it is an index column for the actual dataset.

1b.

Identified the top 5 numerical features correlated with SalePrice.

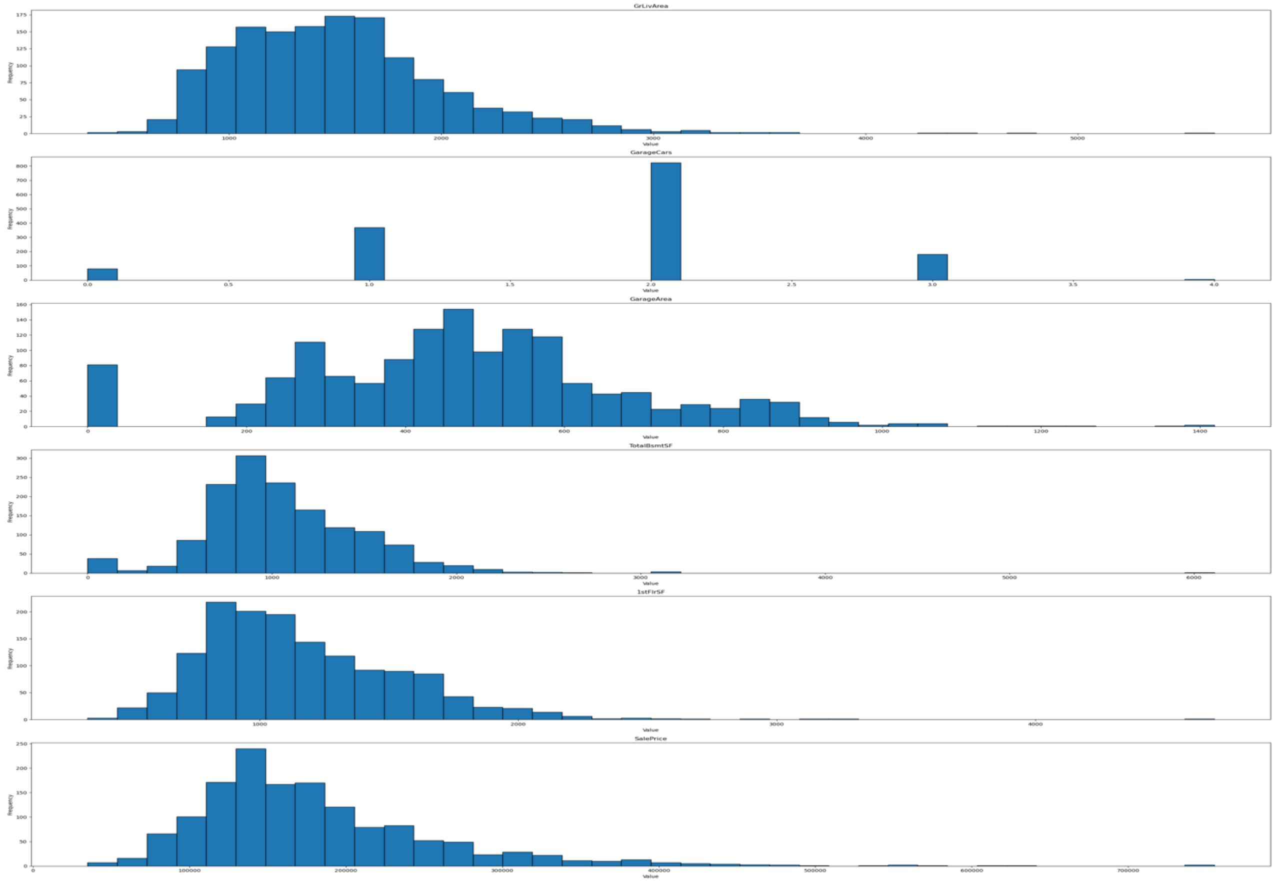
- GrLivArea -> Correlation Value: 0.7086244776126515
- GarageCars -> Correlation Value: 0.6404091972583519
- GarageArea -> Correlation Value: 0.6234314389183622
- TotalBsmtSF -> Correlation Value: 0.6135805515591943
- 1stFlrSF -> Correlation Value: 0.6058521846919153

1c.

Here are the skewness and kurtosis for the top 5 numerical features in correlation with "SalePrice":

- GrLivArea
  - Skewness: 1.365155954883434
  - Kurtosis: 4.874265760253215
- GarageCars
  - Skewness: -0.34219689543081294
  - Kurtosis: 0.2161348715110729
- GarageArea
  - Skewness: 0.1797959420587727
  - Kurtosis: 0.9098227950882034
- TotalBsmtSF
  - Skewness: 1.522688086978629
  - Kurtosis: 13.201041630635025
- 1stFlrSF
  - Skewness: 1.3753417421837937
  - Kurtosis: 5.722076026587489
- SalePrice
  - Skewness: 1.880940746034036
  - Kurtosis: 6.509812011089439

Here is the distribution of the top 5 numerical features and "SalePrice" as a histogram.



Bin value that was chosen allowed noise in the dataset to not be taken into consideration when plotting data distributions. The bin value I chose was ‘int(math.sqrt(n_instances))’. This is a unsupervised discretization technique, which provides a good starting point for binning continuous data where ‘no prior knowledge’ of the data distribution is known. It provides a balance between too many bins, which might overfit and show noise, and too few bins (which might underfit and miss important details in the data.

Based on the outputs for the top 5 numeric we can see that ‘GrLivArea’, ‘TotalBsmtSF’, and ‘1stFlrSF’ showed right-skewed histograms. The high kurtosis values for ‘TotalBsmtSF’, and ‘1stFlrSF’ show there are significant outliers indicating extreme values occur more frequently than would be expected in a normal distribution. ‘GarageCars’ was another difference with it being a negative skew. Features like this show normal distributions with low skewness and kurtosis which means that for its central values, it shows an even spread. SalePrice also shows a high kurtosis value similar to ‘TotalBsmtSF’ and ‘1stFlrSF’, where the outliers impact the distribution.

1d.

Identify missing values in the data. Here are the percentage missing features.

PoolQC (99.520548), MiscFeature (96.301370), Alley (93.767123), Fence (80.753425), MasVnrType (59.726027), FireplaceQu (47.260274), LotFrontage (17.739726), GarageYrBlt (5.547945), GarageCond (5.547945), GarageType (5.547945), GarageFinish (5.547945), GarageQual (5.547945), BsmtExposure (2.602740), BsmtFinType2 (2.602740), BsmtCond (2.534247), BsmtQual (2.534247), BsmtFinType1 (2.534247), MasVnrArea (0.547945), Electrical (0.068493)

Identified the total number of missing features is 19. However, we can assume with the higher values 90+ (‘PoolQC’, ‘MiscFeature’, ‘Alley’, ‘Fence’) that they are extreme outliers due to the number of missing values identified.

2a.

What factors affect house price and how.

Data Mining Goal: Identify key features that most significantly impact house prices.

In which ways do the factors affect the house price?

Data Mining Goal: Observe and rank the key features that most significantly impact house prices.

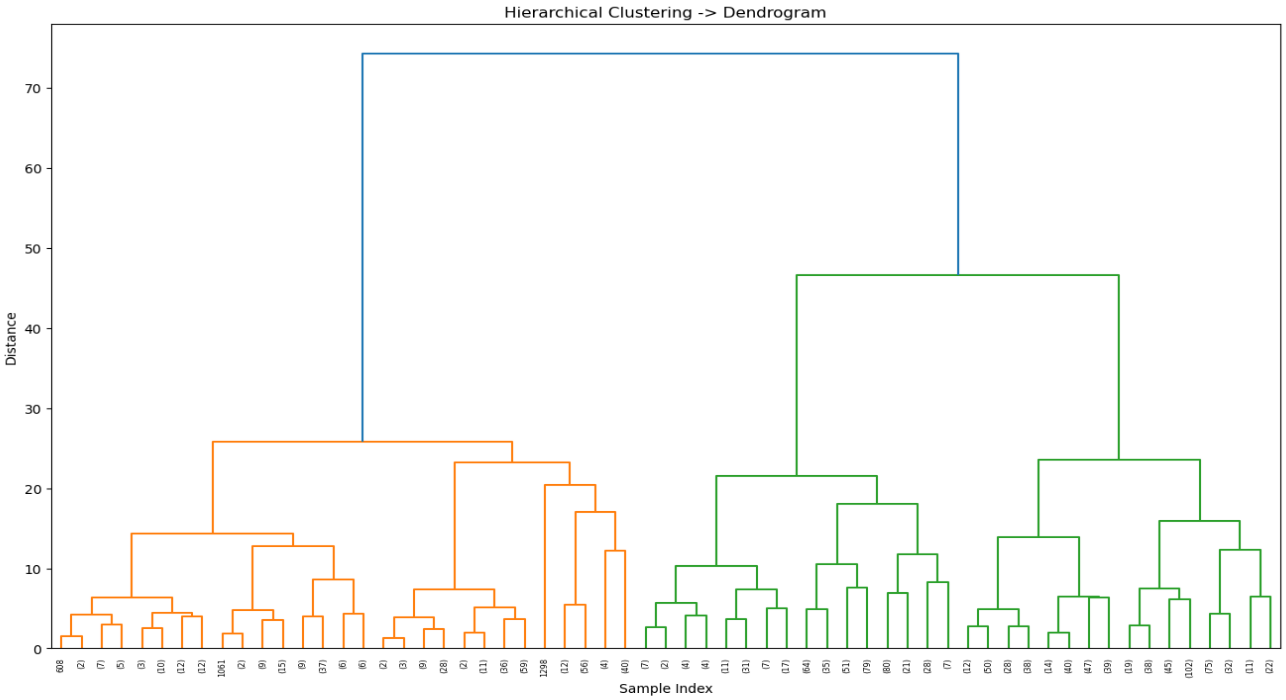
2b.

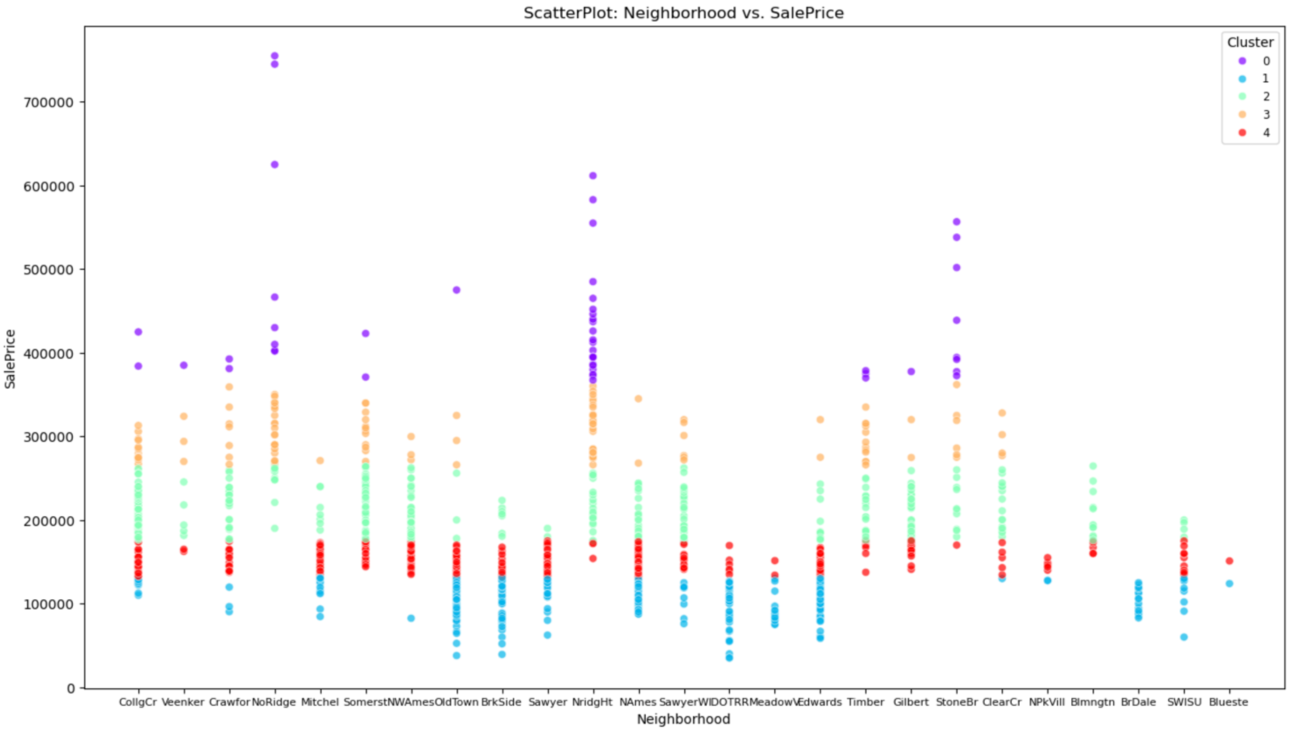
Regression is ideal for these goals as it would help model the relationship between SalePrice (target var) and the categorical and numerical features. By fitting a regression model, we can quantify how changes in each feature impact the house price. Using regression models like Linear Regression will provide coefficients for each feature, indicating its influence and the impact that it has on the effect of SalePrice. Also implementing dimensionality reduction complements this analysis by simplifying the feature space and revealing key patterns. Using reduction methods like PCA (Principal Component Analysis) and t-SNE can reduce the number of features while preserving the essential information. PCA, for example, transforms the original features into a smaller set of principal components that captures the most variance. This allows visualizing complex data as well as highlighting the most important features affecting SalePrice by ranking their contribution to these principal components.

Integrating both regression and dimensionality reduction will allow a comprehensive understanding of the main features affecting house prices.

3.

From the resulting dendrogram, we can identify that the houses from each neighborhood fall into two main clusters. This shows there is a correlation in the dataset that forms these two clusters. These two clusters have significant differences as seen at higher levels of the dendrogram indicating that houses within each of the two clusters have different feature values as well as SalePrice. Although the dendrogram does not explicitly show the SalePrice in correlation with the neighborhood, we can assume that it does vary, and houses with the same features are in the same neighborhood. Observing the lower areas of the dendrogram we can identify that the houses in that area have similar features within their clusters. To identify this further, I used Agglomerative Clustering with 5 clusters to group data based on their SalePrice in correlation with the neighborhood. I then used a scatterplot to visualize the relationship between neighborhoods and SalePrice. From the output, we can identify the 0 clusters with neighborhoods like NridgHt, NoRidge, and StoneBr identified as high-quality (SalePrice around 400000+) neighborhoods with fewer houses in comparison to some other clusters. Clustering has identified these neighborhoods, as having high-value features, which is reflected by the prices. Cluster 1 is significantly lower (80000-120000) showing they are more affordable or have older properties (OldTown!) compared to those in Cluster 0. This shows a distinct grouping of neighborhoods with lower feature values. Cluster 2 and 3 range from 180000-350000 showing distinct grouping of neighborhoods with similar features. Cluster 4 ranges from 120-180000 indicating lower-mid range values. From observing this scatterplot, we can make the statement that house prices do vary by neighborhood shown by the clear distribution of higher, mid, and lower clusters.





Part 2: Data Preparation and Machine Learning

1.

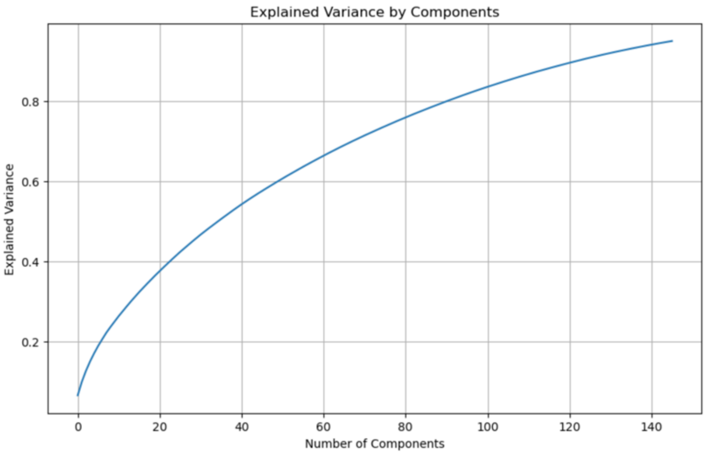
Handled missing numerical and categorical values in the dataset. For missing numerical, I filled the missing values with the median value of each column for both training and testing sets. Use median because they aren't affected by the outliers ensuring that the replacement data is more reflective of the values present in the dataset and not affected by outliers. For missing categorical, I replaced them with 'NA' in both training and testing sets. This was due to NaN throwing issues when performing dimensionality reduction techniques with PCA and ICA.

Then sort the numerical and categorical values in the dataset. There are ordinal and nominal categorical data in the dataset, as well as three categorical data columns ('MSSubClass', 'OverallQual', 'OverallCond') in the numerical values. I created a list of all the identified ordinal data, where I then separated these categorical data values from the remainder of the categorical data and numerical datasets using the sort_categorical_numerical method. Stored all the ordinal, nominal, and numerical data values in a list which I then encoded. For ordinal encoding, I encoded them using OrdinalEncoder (assigns numerical values to ordered categories). For nominal encoding, I used one-hot encoding (uses pd.get_dummies to create binary cols for each category). For numerical encoding, I used a MinMaxScaler (normalizes the numerical data). After encoding I then aligned the training and testing datasets to ensure they have the same columns.

After encoding then normalized to ensure training and testing datasets are standardized. Used StandardScaler to ensure features have a mean of 0 and a standard deviation of 1. Then the processed training a testing datasets can be found in the output_data directory.

2.

Used PCA and ICA dimensionality reduction techniques to identify irrelevant/redundant features. PCA is a feature extraction technique that projects the data onto new axes that better represent the input data points. It does this by computing a covariance matrix of the input data, which helps define a transformation mapping the original data onto a new set of dimensions. The process involves finding the eigenvectors and eigenvalues of this matrix. Eigenvectors represent directions that remain unchanged under the transformation, and PCA minimizes the sum of squared errors across all dimensions, unlike regression, which does so in only one dimension. For my implementation, I chose n_components=0.95, meaning the PCA retained enough components to explain at least 0.95 of the variance in the data. The SalePrice column was removed from the input data before applying PCA, as it represents the target variable and not a feature. Including it would have distorted the PCA analysis. The reduced datasets retained 146 principal components, explaining 0.95 of the variance.



ICA focuses on transforming data into statistically independent components by analyzing and reducing the dimensionality of the feature space, rather than orthogonal ones(PCA). I applied ICA to the same data, setting components=146 to match the PCA results. ICA managed to cut the feature space from 252 features to 146 relevant features. This alignment allowed for a direct comparison between the components identified by PCA and ICA, ensuring that the training and testing datasets retained the same shapes. Also removed the SalePrice column from input training and testing data before applying ICA to avoid distorting the analysis.

Training and test sets respectively computed from using PCA and ICA were then converted to data-frames to then be used in machine learning questions below.

3a.

Linear Regression Model -> on PCA and ICA Reduced Data

```
LR - PCA - MSE: Training = 0.09980793820137084 , Testing = 0.13515937127788202
LR - PCA - R^2: Training = 0.9001920617986292 , Testing = 0.866449429003867
LR - ICA - MSE: Training = 0.09980793820137081 , Testing = 0.13515937127788194
LR - ICA - R^2: Training = 0.9001920617986292 , Testing = 0.8664494290038671
```

Ridge Regression Model -> on PCA and ICA Reduced Data

```
RR - PCA - MSE: Training = 0.09980797125421242 , Testing = 0.135125240213231
RR - PCA - R^2: Training = 0.9001920287457876 , Testing = 0.8664831538068885
RR - ICA - MSE: Training = 0.09980815345404428 , Testing = 0.13513978677236282
RR - ICA - R^2: Training = 0.9001918465459557 , Testing = 0.8664687803952654
```

The MSE values that we identified are similar across PCA and ICA-reduced datasets for both regression models. This indicates that the predictive performance in terms of error is consistent across different dimensionality reduction techniques. The R^2 scores (proportion of variance explained by the model) are similar between PCA and ICA-reduced datasets for both regression models. This suggests that both dimensionality reduction techniques maintain similar explanatory power for the models. The differences between MSE and R^2 show that the inclusion of ridge penalty (0.5) doesn’t change the model performance, although the ridge regression model does slightly improve the testing R^2 score, indicating good generalization on unseen data. PCA aims to capture the maximum variance in the data while ICA focuses on statistical independence. Based on the results we can make the statement that either approach is effective for this dataset in terms of using the linear and ridge regression models. Linear regression, due to it being a simple regression method, did not perform well on this dataset due to the varying feature influence, which might’ve not enabled it to define a ‘clear line’, causing it to underfit. The ridge regression model did perform slightly better (in terms of R^2) on the testing set which may be due to ridge regression’s ability to encourage smaller coefficients, reducing **overreliance** on any single feature, which is beneficial in a dataset with varying feature influence.

3b.

Random Forest Model -> on PCA and ICA Reduced Data

```
RF - PCA - MSE: Training = 0.028708301448299205 , Testing = 0.1527318323169427
RF - PCA - R^2: Training = 0.9712916985517008 , Testing = 0.8490861327456378
RF - ICA - MSE: Training = 0.06216143419153855 , Testing = 0.494285592136381
RF - ICA - R^2: Training = 0.9378385658084615 , Testing = 0.5115978829965314
```

Using the Random Forest model, we can see significant improvement in predictive performance in comparison to the linear and ridge regression models, especially with PCA. The random forest model achieved a notably lower MSE and a higher R^2 on both training and testing sets for PCA. However, its performance on ICA reduced data, while still better than the linear models, shows a larger gap between training and testing results, suggesting that Random Forest is better from the variance obtained, PCA, than statistical independence, ICA. Random forest is very good at identifying complex patterns however it does tend to overfit when handling a large dataset, which can be seen when observing the MSE outputs for PCA and ICA. Can then make the statement that due to the data's complexity, linear and ridge regression models would be preferable to make better/identifiable results with higher accuracy just by observing the shifts in training and testing MSE and R^2 accuracy across the regression models. Unless more preprocessing steps can be implemented, removing irrelevant data and removing outliers, can be implemented to reduce overfitting across regression models, I would recommend using Ridge regression just based on the MSE and R^2 values that have been identified, if using dimensionality reduction.

References

- ChatGPT (Help with plotting and debugging for question 1,3 for part 1. Help with debugging encoding, missing value methods, and PCA reduction implementation for part 2)
- https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html (scikit- FastICA library -> help implementing ICA dimensionality reduction)
- https://www.datacamp.com/tutorial/one-hot-encoding-python-tutorial (one-hot encoding -> help implementing one-hot encoding for part 2)
- https://ecs.wgtn.ac.nz/foswiki/pub/Courses/COMP309_2024T2/LectureSchedule/Comp309_Week5-Tutorial.ipynb (comp309 tutorial -> help with analyzing datasets for the part 1)
- https://ecs.wgtn.ac.nz/foswiki/pub/Courses/COMP309_2024T2/LectureSchedule/week6a-data-preprocessing.pdf (comp309 lecture 10 -> help encoding for part 2)
- https://ecs.wgtn.ac.nz/foswiki/pub/Courses/COMP309_2024T2/LectureSchedule/week3a-regression-dimensionality-reduction1.pdf (comp309 lecture 4 -> understanding regression for part 2 and answering the question regarding PCA for part1.2 and part2.2 and 3)