Optimizing Class Labels For a Multi-Layer Perceptron Model For Housing Sale-Price Prediction

By Avi Hiriyanna (KSMC Data Scientist Candidate)

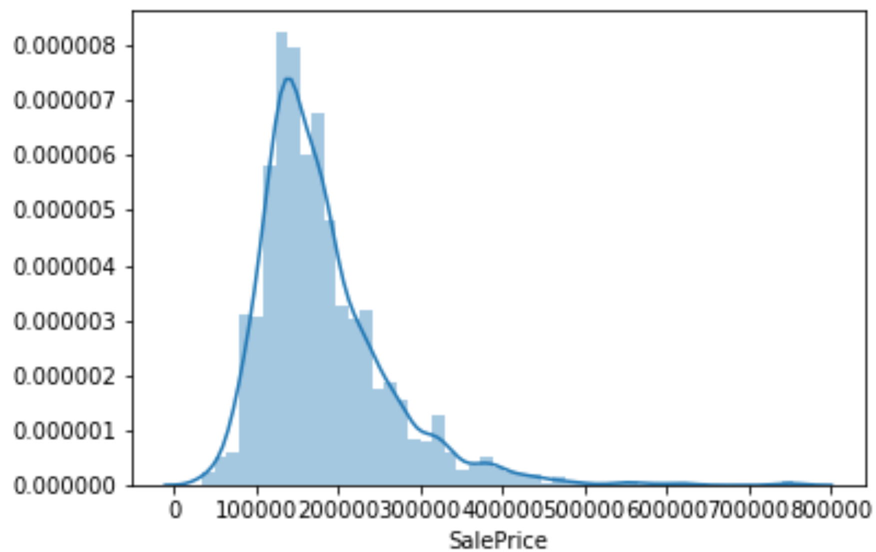
Overview

- Introduction
- Summary Statistics
- Model Development
  - Methodology
  - Results
- Conclusions

Introduction

- The Ames Housing Dataset was created as an alternative to the Boston Housing Dataset due to it's age, limited samples, and outdated feature values

- Data at First Glance
  - 2930 Observations
    - Training Set: 1460 Observations
    - Test Set: 1459 Observations
  - 78 Features (excluding ID and Sale Price)

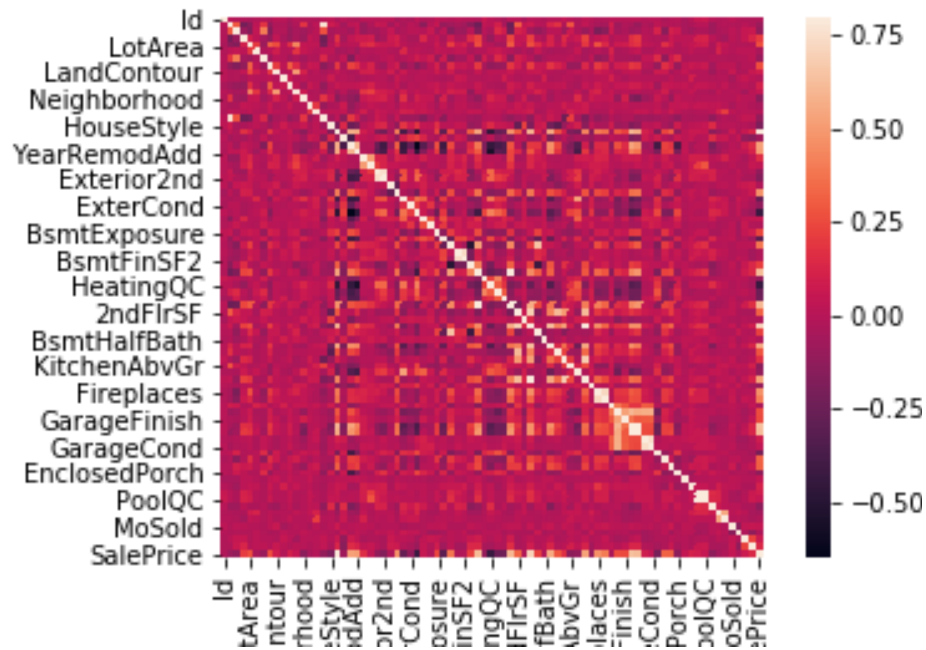


+ Data Descriptions: Sale Price

- Exploratory Statistics:
  - Mean: $180,921
  - Minimum: $214,000
  - Maximum: $755,000

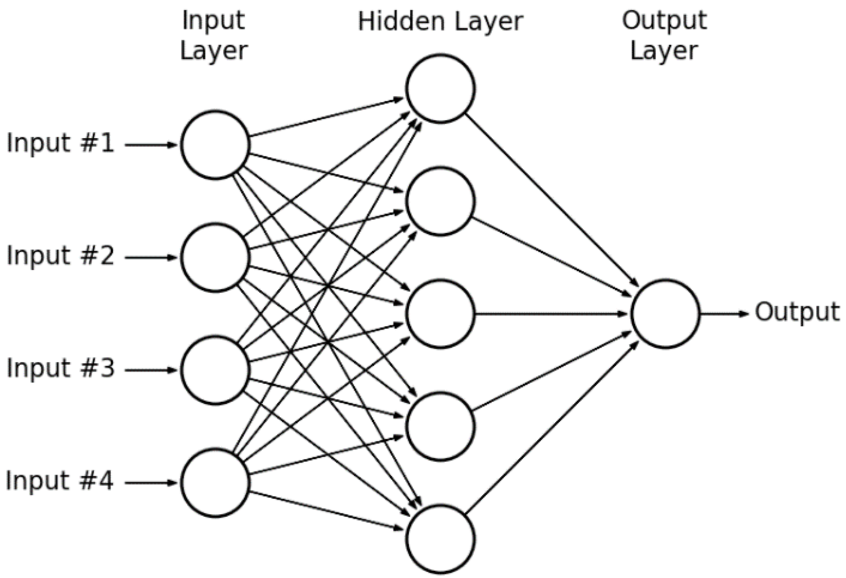
- Top 3 Correlated Features
  - Overall Quality (r=0.79)
  - Above ground living area square feet (r=0.71)
  - # of Car Garages (r=0.64)

- Top 3 Anti-Correlated Features
  - Type of Foundation (r=-0.43)
  - Heating Quality and Condition (r = -0.42)
  - Basement Finish Type 1 square feet (r=-0.30)



Model Development: How to Optimize Class Labels for MLP?

- What type of labels does a simple MLP best classify to?
  - Few Options for Labels
    - Use the original sale prices
    - Cluster the data using a centroid based method, and use designated cluster as the label
    - Label sale price depending on what quartile it falls into



+ Model Development: Pre-Processing

- The dataset has both qualitative and quantitative features
  - In order to consolidate the qualitative features, I enumerated the unique features for each column, and came it a numerical label (n=1,2,3…etc)
    - *Ex. labeledList, uniques = pd.factorize(currData)*
      - The factorize function essentially takes a unique list of samples for a particular feature, and labels them sequentially, thus giving it a numerical class label.
      - Transforming qualitative features to numbers makes it infinitely easier to deal with.

- Data Normalization
  - I utilized the StandardScaler function to normalize the features of the dataset, to optimize performance by making the data zero-mean.

Model Development: Feature Selection

- No Feature Selection Was Used in this Exercise due to the lack of time.
- Ideally it would have been beneficial to run PCA, mRMR, or another filter type method in order to minimize the feature-space to the most relevant and minimally redundant features.
- We could then ideally test a model with the entire feature set or the reduced feature set and see what works best for us.

+ Model Development: Class Labeling

- Method 1: Utilize the Prices As Is
  - Keep dollar value labels at a multitude of prices
  - May be non-ideal due to the large array of prices presented
  - Would need to utilize more complex methods (Recurrent Neural Networks, etc.) to get better prediction of ACTUAL prices
- Method 2: Utilize K-Means in order to generate Labels
  - Minimizes the number of labels than using just housing prices
  - May be non-ideal due to the stochastic nature of K-Means; making it hard to have matching labels with a given test set
- Method 3: Break the Prices down into Quartiles and classify the houses via Quartile
  - Minimizes the number of labels than using just housing prices
  - Acts as an appropriate proxy label to actual housing price
  - Generated by a simple conditional statements