HOUSE PRICES: ADVANCED REGRESSION TECHINQUES

I. Definiton:

Project overview: the challenge of this project is to produce a web-interface that will read user-defined inputs to predict the price of the user's house depending on the characteristics he/she has provided.
The project originates from a Kaggle competition (XXX) where the aim is to predict the house prices of the XXX's inhabitants. The dataset used to train the predictive model has been provided by Kaggle itself.

Problem statement: the problem is to successfully predict the price of a given house according to its characteristics. The attempt to solve such an issue will be deployed in the following manner:

- Analyze the dataset to better understand the data and its distribution.
- Deploy a feature selection algorithm in order to break down the number of independent variables used to predict a house's price.
- Build and train a XXX Neural Network to make the predictions
- Create a web-interface providing 10 drop-down menus for the user to provide inputs describing the characteristics of his/her house.
- Link the trained NN in Sagemaker through using a Gateway API in order for it to receive the user inputs and make predictions upon them. Put a Lambda function in place to ingest the user input and transform it in such a way that predictions can be made and sent back.

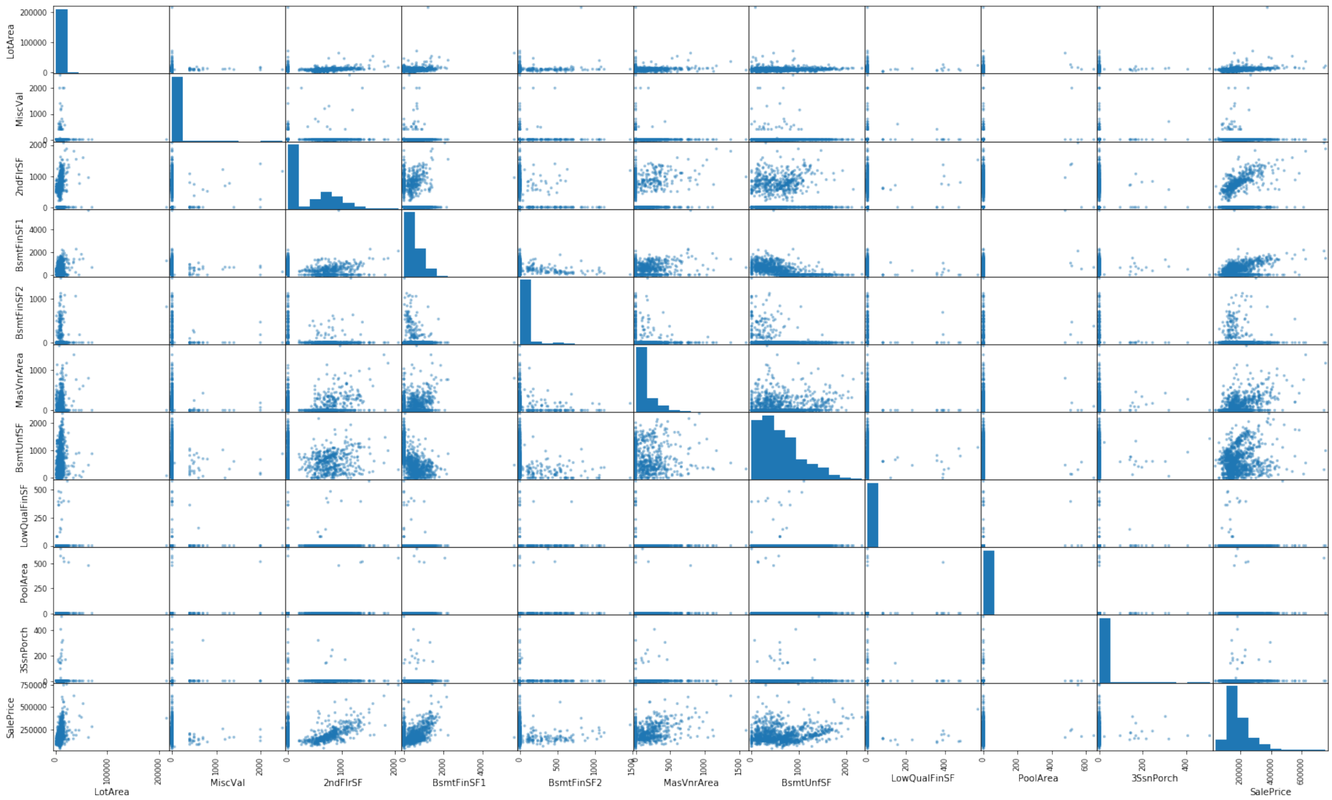
Metrics: in order to analyze how successful, the predictions are, I am planning on using the Root Mean Squared Error (RMSE). I believe that this metric will adequately assess the goodness of the predictive model given that it gives a realistic and measurable error rate. RMSE comes in the unit of measure in which the dependent variable is measured (in this case $).

Benchmark:
Looking on Kaggle there are many submissions for this competition and I am seeing that people making their first submissions average an RMSE of 50K. This means that their submissions are on average +/- 50K $ off in price.

II. Analysis:

Data exploration:

- Scatter matrix

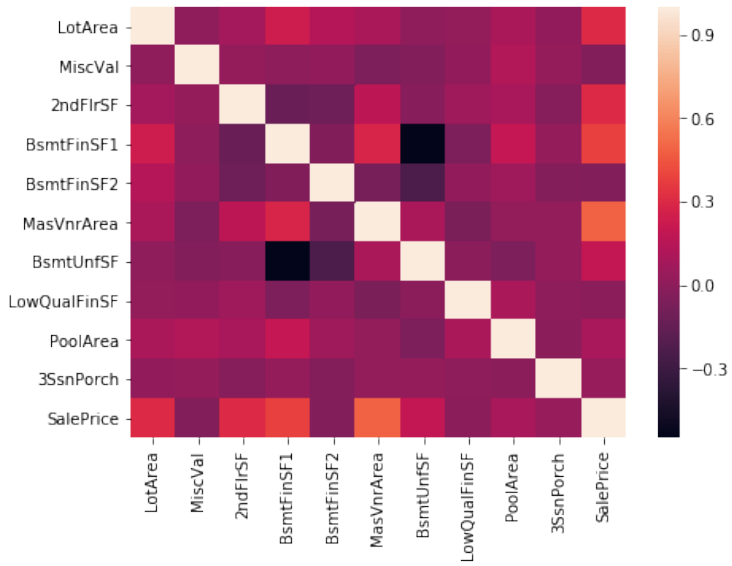


The scatter matrix above shows an exploration of the 10 variables with the strongest explicative power with respect to the dependent variable (SalePrice). The scatter matrix above shows each variable's distribution as well as a scatter plot exemplifying the correlation between each and every variable (amongst the selected ones).

It appears that the predominant distribution across the top 10 independent variables is the logarithmic one, where high frequencies of occurrences take place with low values of the respective variable. Said frequencies drop quite rapidly as measures of the respective variable increase.

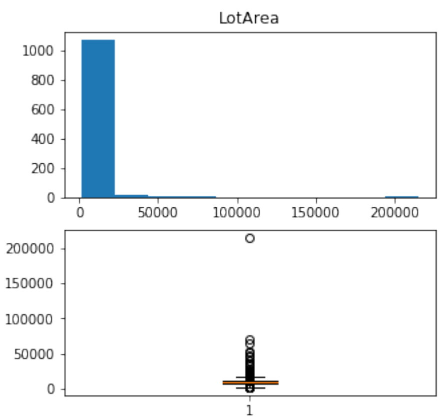
According to the scatter plots where the independent variables are plotted against the dependent one, the variables that show the strongest levels of positive correlation are: LotArea, 2ndFlrSF, BsmtFinSF1, MasVnrArea.

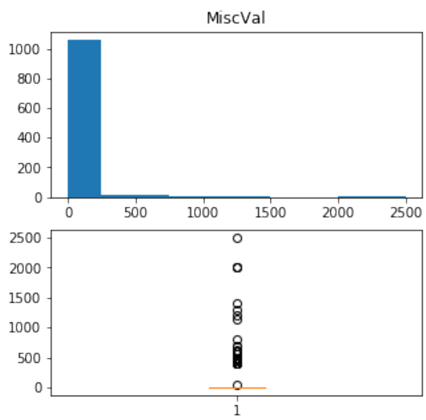
- Correlation Heatmap:

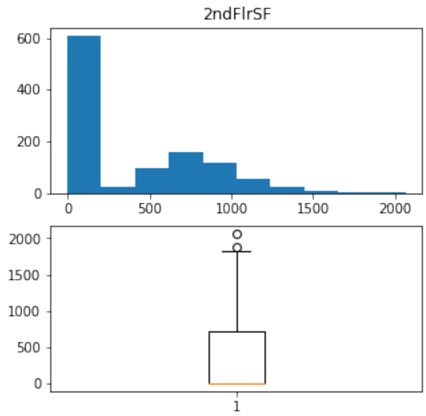


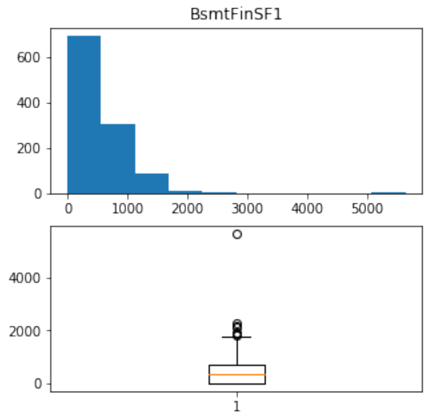
The correlation heatmap above reinforces the concepts depicted by the scatter matrix presented beforehand, with variables like: LotArea, 2ndFlrSF, BsmtFinSF1, MasVnrArea having the highest levels of correlation (above 0.3) with the dependent variable (SalePrice).

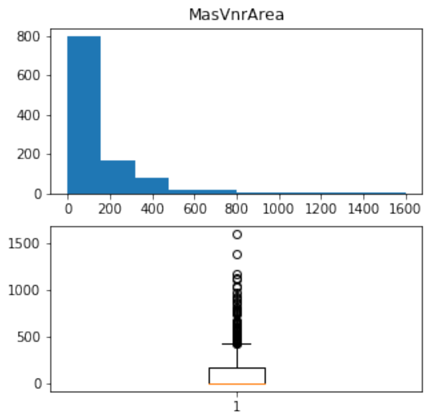
- Variable exploration:

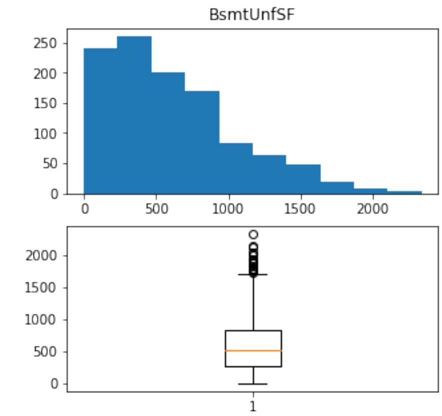


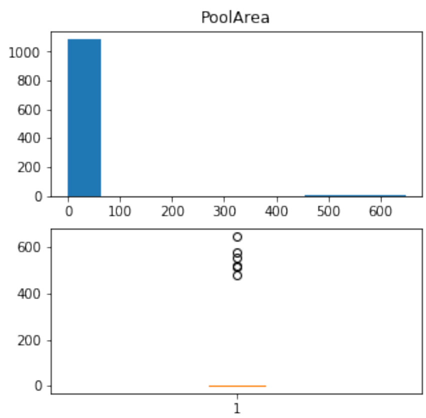


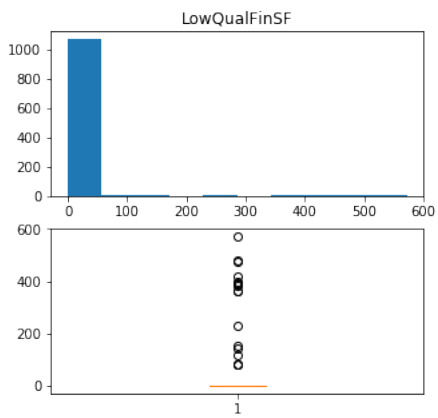

2ndFlrSF

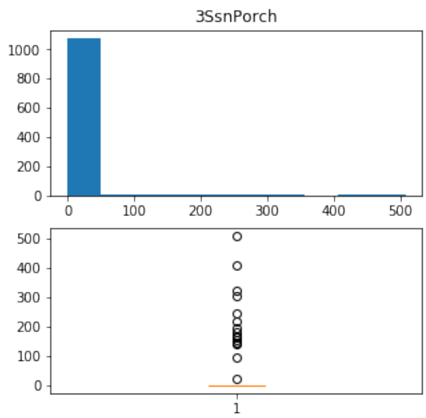

BsmtFinSF1

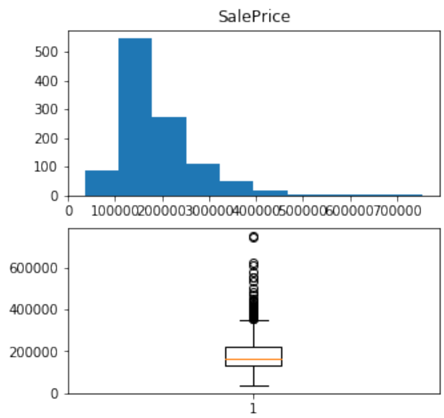

MasVnrArea


BsmtUnfSF


PoolArea


LowQualFinSF





III. Methodology:

<u>Data processing:</u> To find out which variables provide the highest level of explicative power in terms of variability with respect to the dependent variable, a feature selection algorithm has been applied. For this task I decided to use Univariate Selection.

The above plots depicting the distribution of every variable as well as a boxplot accounting for percentiles (25 & 75) as well as mean and outliers, have lead me to take into consideration the following hypothesis: the logarithmic distribution justifies a strong number of data points at low levels of measurement for each variable, and therefore the less frequent, high values for each and every one of those variables may seem outliers at first, however I believe that by removing them important data points storing valuable information would be removed from the dataset thus reducing the variability in the data which may produce better results in the training set, however, affecting negatively the performance of the model's generalization power (against the test set). Keeping this in mind, this is why I decided to keep these points who look like outliers when analyzing the box plots.

<u>Implementation:</u>

The implemented solution has managed the data from its source in the following way:

- Feature Selection: use Univariate Selection algorithm to pick the top 10 variables with the strongest predictive power.
- Anomaly inspection: these top 10 variables have been thoroughly inspected for any outliers or abnormalities
- Training/Predictions: an XXX Neural Network has been built, and trained according to which hyperparameters delivered the best results.
- Web-interface generation: create a simple web page and through the use of drop-down menus store user-defined inputs that will display the characteristics of a user's house.

- Lambda function to ingest user input: put in place a Lambda function that will process the user inputs and call the NN endpoint in order to make predictions. The lambda function will also deliver said predictions.
- API Gateway: API used to connect the web-interface with the model and lambda in order to deliver the predictions made.

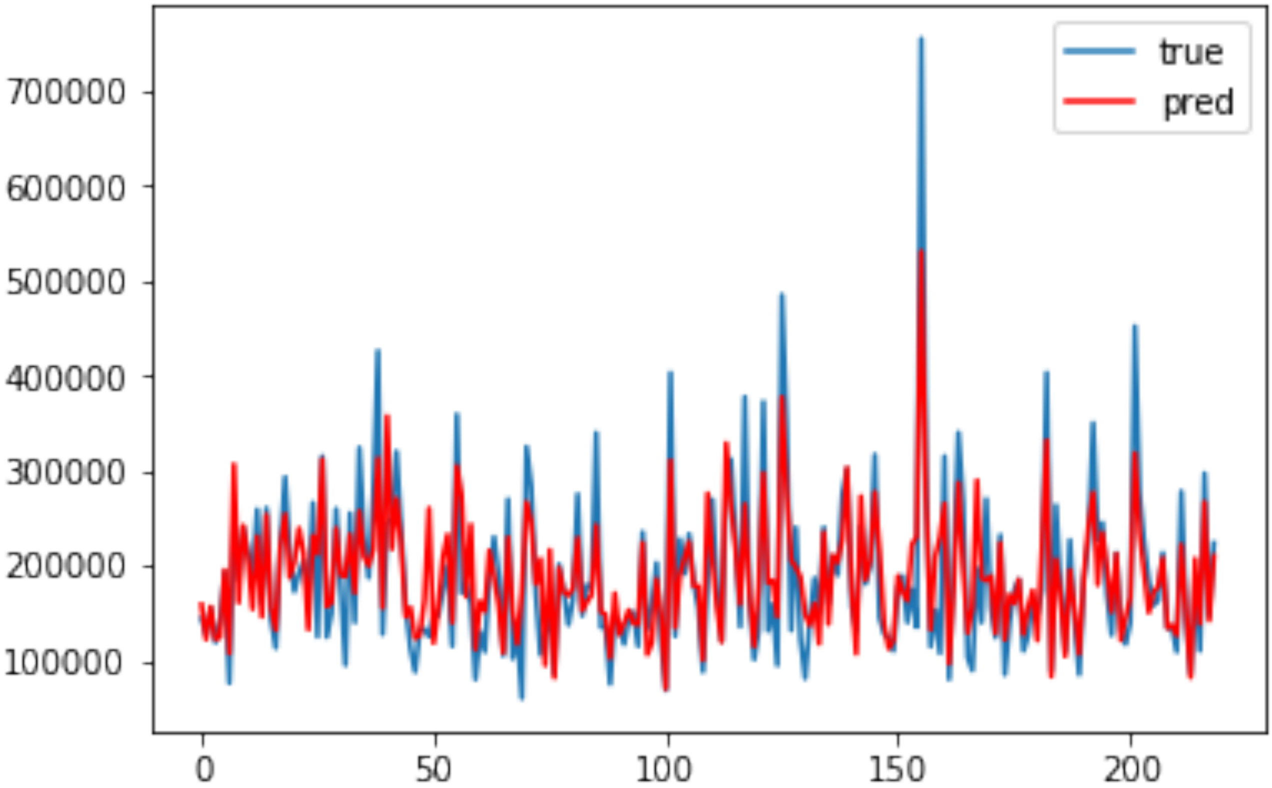
Refinement: the process of refinement put into practice has been the manual adjustments made to the Neural Network's architecture by tweaking both its structure (layers, neurons ...etc.) as well as its hyperparameters in order to optimize for loss during the training phase.
After various training phases, an optimal solution has been reached. The Neural Network with the following characteristics has been put in place:

- 2 layers
- 128 neurons
- Loss function – XXX
- Number of epochs – XXX
- Loss rate – 0.01

I have not put in place the hyperparameter optimization technique provided by Sagemaker given that I built this NN from scratch and wanted to understand how each hyperparameter as well as component of its architecture would affect the training phase. This is why I did not implement the automated command the picks the best hyperparameters for you, but I went for a "manual" and more mechanical version of it (trial and error).

IV. Results:

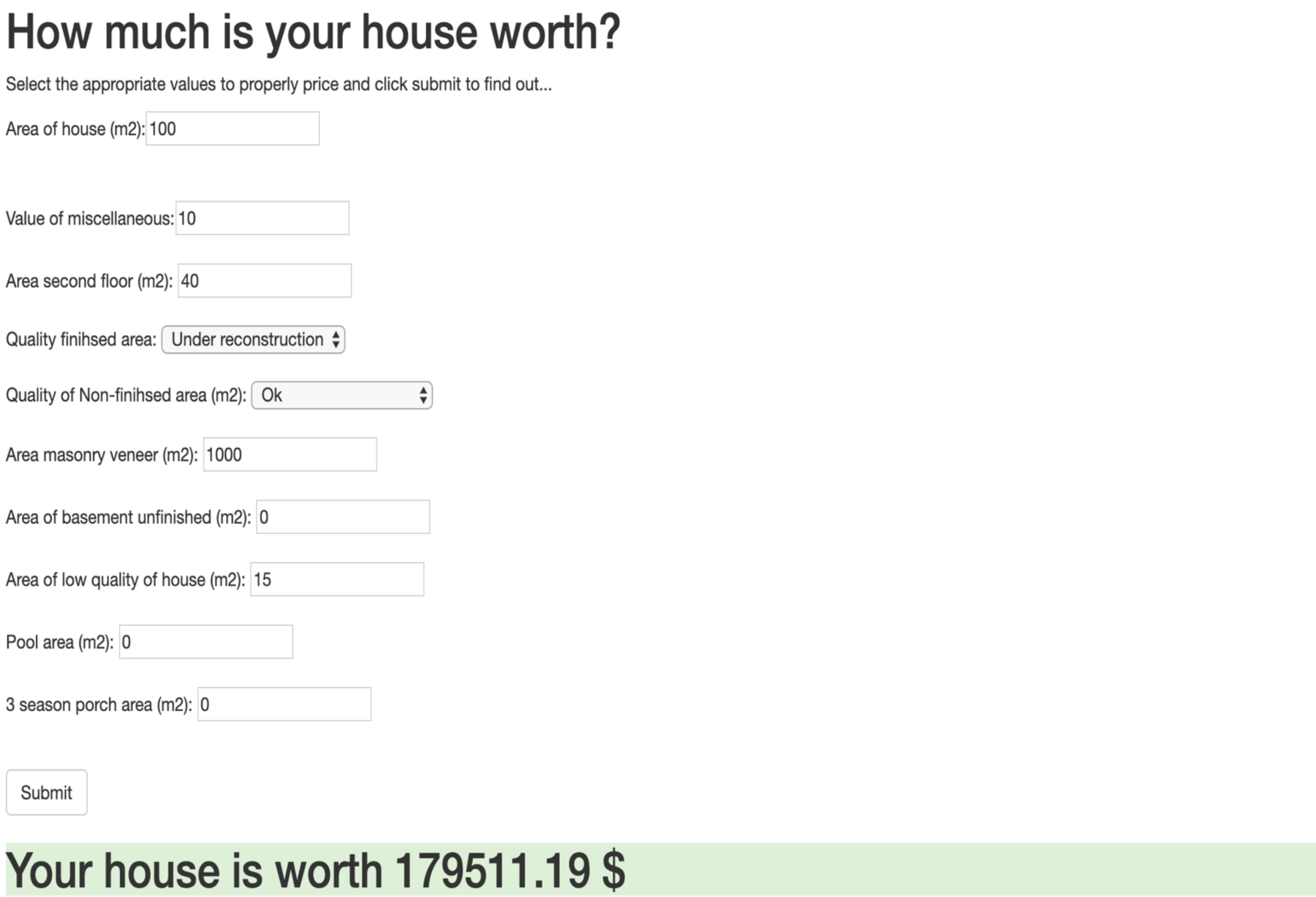
Model evaluation and validation:



- `MSE: 2,086,653,985.72`
- `RMSE: 45,679.91`

Looking at the graph above it is clear how the deployed model is performing quite well with houses that tend to float around a price of 200K, however, it looks like it doesn’t pick up houses with especially high prices (>= 400K).
An improvement would be to look in the determining factors that make the price of th ese houses so high and find ways to make it easier for the model to pick up such a signi ficant trait.
Clustering is an option.
PCA is also an option considering.

V. Conclusion:



This visualization is to show how a first version of the end to end solution works. The idea of this project was to give an end user a quick and easy way to find out its house's worth.

Reflection:

The most interesting aspect of the project in m phase opinion was to make the Lambda function work. It was amazing to see how the inputs went from the frontend to the Lambda that was linked with the Neural Network endpoint that made the prediction and sent it back to the front end.

Improvement:

This final submission fits my expectations because the skeleton sitting behind this infrastructure is solid and works really well.
The next steps I would apply to this project are the following:

1. In-depth analysis of the model's failures to find ways to improve it (clustering, PCA are just initial ideas)
2. Improve the front-end, in order to give the user a better UI/UX experience due to the look and feel of the app.

It would be interesting to see how people would interact with such an app and maybe integrate it with some other functionality like:

- Exemplifying the characteristics of houses with similar house prices
- Adding a location functionality – geo location (latitude and longitude) features are always interesting and very visually appealing in an app.