# Analysis of Building Permit Applications in San Francisco

Javier Niño-Sears

Brown University

December 22nd, 2023

https://github.com/javierninosears/data1030project

Word Count: 1978

## INTRODUCTION

In the last thirty years, the housing crisis in San Francisco has exploded. With the tech industry infusing wealth into the region, the demand for housing has far outpaced the supply. This in turn has led to an increase in housing prices throughout the region, skyrocketing the cost of living.[1] But in conjunction with the acceleration in costs, the city of San Francisco has adopted some of the country's strictest zoning regulations.[2] This dangerous combination of amplified demand and bureaucratically-induced supply issues has left San Francisco, and the Bay Area, in real limbo.

I became curious to find any quantitative explanations for the effects of bureaucracy on development in the area. I found a dataset created and maintained by the City of San Francisco's Open Data Portal, documenting all of the city's permit applications dating back to 1901. The dataset has 1,226,960 records, but for ease of analysis I chose a subset of records with filing dates ranging from January 1, 2013 to December 31, 2016, leaving me with 198,900 records. The dataset's features include "Permit Type", "Number of Proposed Stories", "Estimated Cost", "Proposed Use", "Proposed Construction Type", and "Location" (geographic coordinates of proposed development), all of which could potentially be informative for a predictive model.

The model was engineered with the following research question in mind: **Can I predict how long it takes for a given building permit to be issued in the City of San Francisco?** This is a regression problem, because I am predicting a continuous variable—in this case, the number

---

[1] Cutler, Kim-Mai. "So You Want to Fix the Housing Crisis." *TechCrunch*, 4 Nov. 2014, techcrunch.com/2014/11/02/so-you-want-to-fix-the-housing-crisis/.

[2] *Bizjournals.Com*, www.bizjournals.com/sanfrancisco/news/2017/04/28/san-francisco-seattle-housing-production-pipelines.html. Accessed 20 Dec. 2023.

of days between filing a permit with the city and having it issued. An analysis of this question could be incredibly valuable for the construction industry, with the delay in issuing building permits considered the main reason for discrepancy in demand and supply in the real estate industry.[3] Additionally, a geographic analysis of the variance in permit issuance times across neighborhoods could unearth systemic inequities in development speeds in different neighborhoods, showcasing an unjust prioritization of certain neighborhoods and demographic groups over others.

There have been previous analyses on this dataset, the most notable being an analysis that formed the foundation of a 2022 SF Chronicle article by Dustin Gardiner and Susie Neilson.[4] Notably, the authors' analysis was not machine learning based; they just conducted exploratory data analysis on the dataset. Yet its relevance to my analysis is nonetheless important to discuss. Their conclusion supports the motivation for my analysis: permits take too long in San Francisco to be issued, a problem that the authors describe as "driving away many potential builders and shrinking the number of new homes created, even as the housing crisis has intensified with skyrocketing home prices and rents". A thorough analysis of this dataset could unearth new insights that could contribute to solving the housing crisis in San Francisco.

## EXPLORATORY DATA ANALYSIS

After performing some exploratory data analysis on the dataset, I generated a few key insights that helped guide my model creation process. First, I mapped the target variable to get an

[3] "City Building Permit Delays Costing Developers Time and Money." *Business in Vancouver*, 19 Dec. 2017, biv.com/article/2014/11/city-building-permit-delays-costing-developers-tim.

[4] *627 Days, Just for the Permit: This Data Shows the Staggering Timeline ...*, www.sfchronicle.com/sf/article/housing-permits-san-francisco-17652633.php. Accessed 21 Dec. 2023.

understanding of the types of values I was predicting. This analysis can be seen in Figure 1 below.

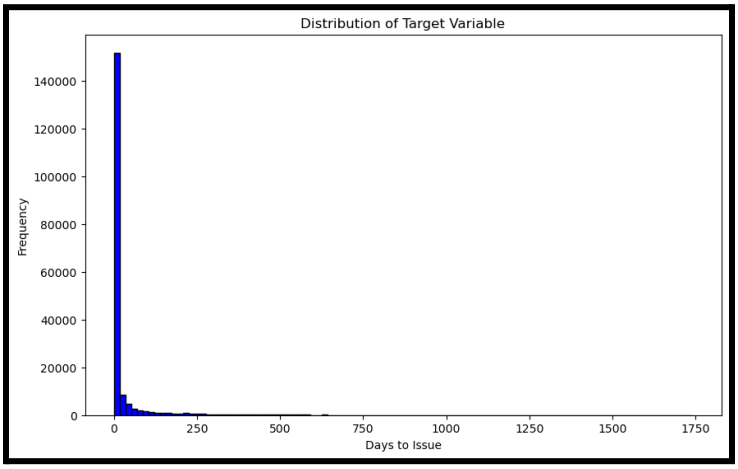


*Figure 1. Frequency Distribution of Target Variable*

The graph is extraordinarily right skewed. In fact, 115,488 of the 198,900 records in the dataset, or 58 percent, represent permits that were issued the same day of filing. This poses some interesting questions about why these permits were issued immediately, and why other permits take longer to be issued.

Another analysis I deemed relevant was to examine the number of permits in the dataset by neighborhood. The resulting data analysis is in Figure 2 below. As could be expected, the Financial District/South Beach neighborhood has the highest amount of building permits in the dataset. This tracks with the intuition that the business and commerce hub of the city would

reasonably have more commercial development activity.

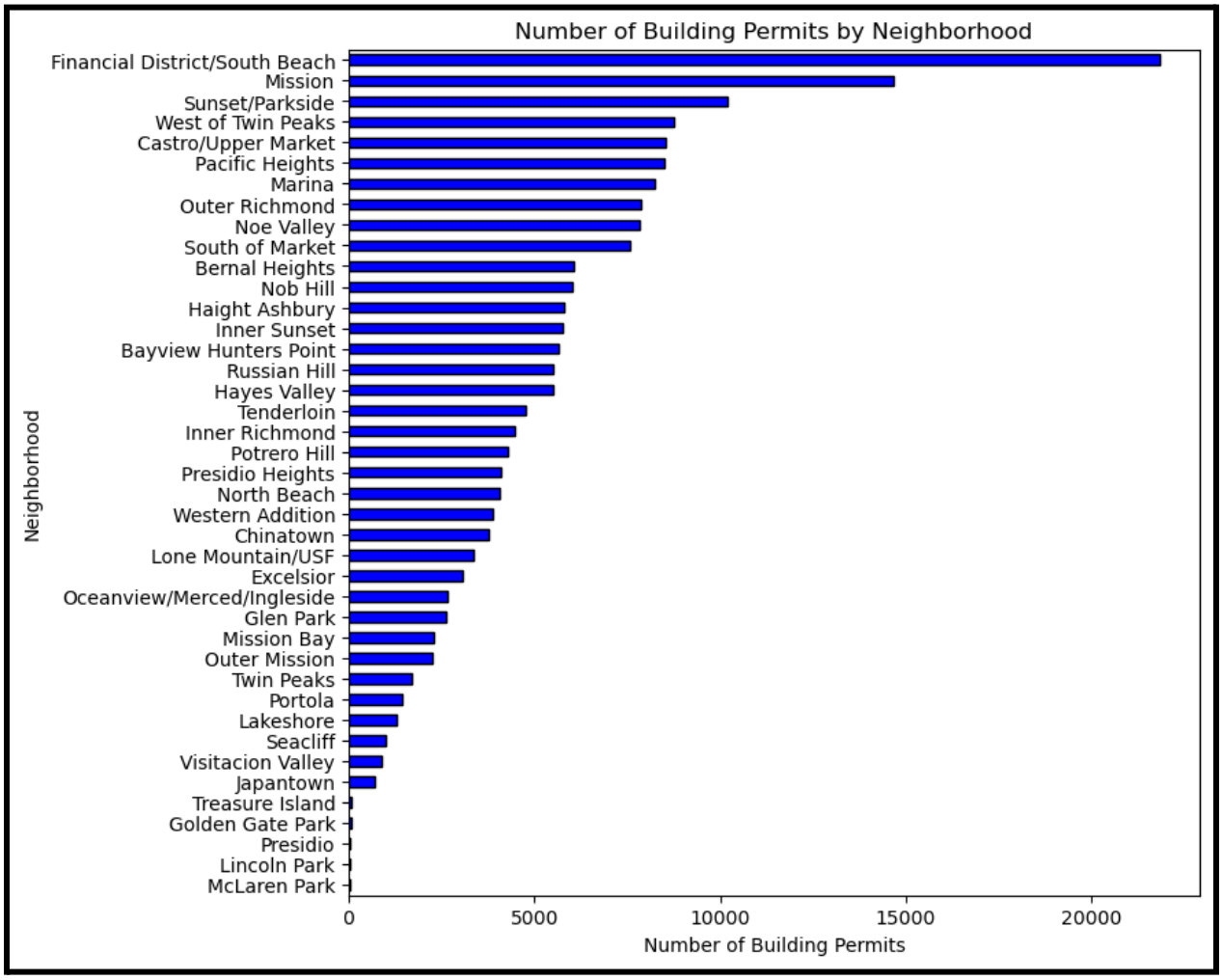


*Figure 2. Number of Building Permits by Neighborhood*

One last piece of exploratory data analysis I conducted was to map the estimated cost of building permits by neighborhood, to see where the most lucrative developments in the city were taking place. The results of this analysis can be seen in Figure 3 below. The neighborhood with the highest mean estimated cost, by an enormous margin, is Mission Bay. Chase Center was constructed in that neighborhood in the time period of this dataset, the most expensive building permit in the records (with an estimated cost of $537,958,646), serving to drive up the mean estimated cost in the Mission Bay neighborhood.

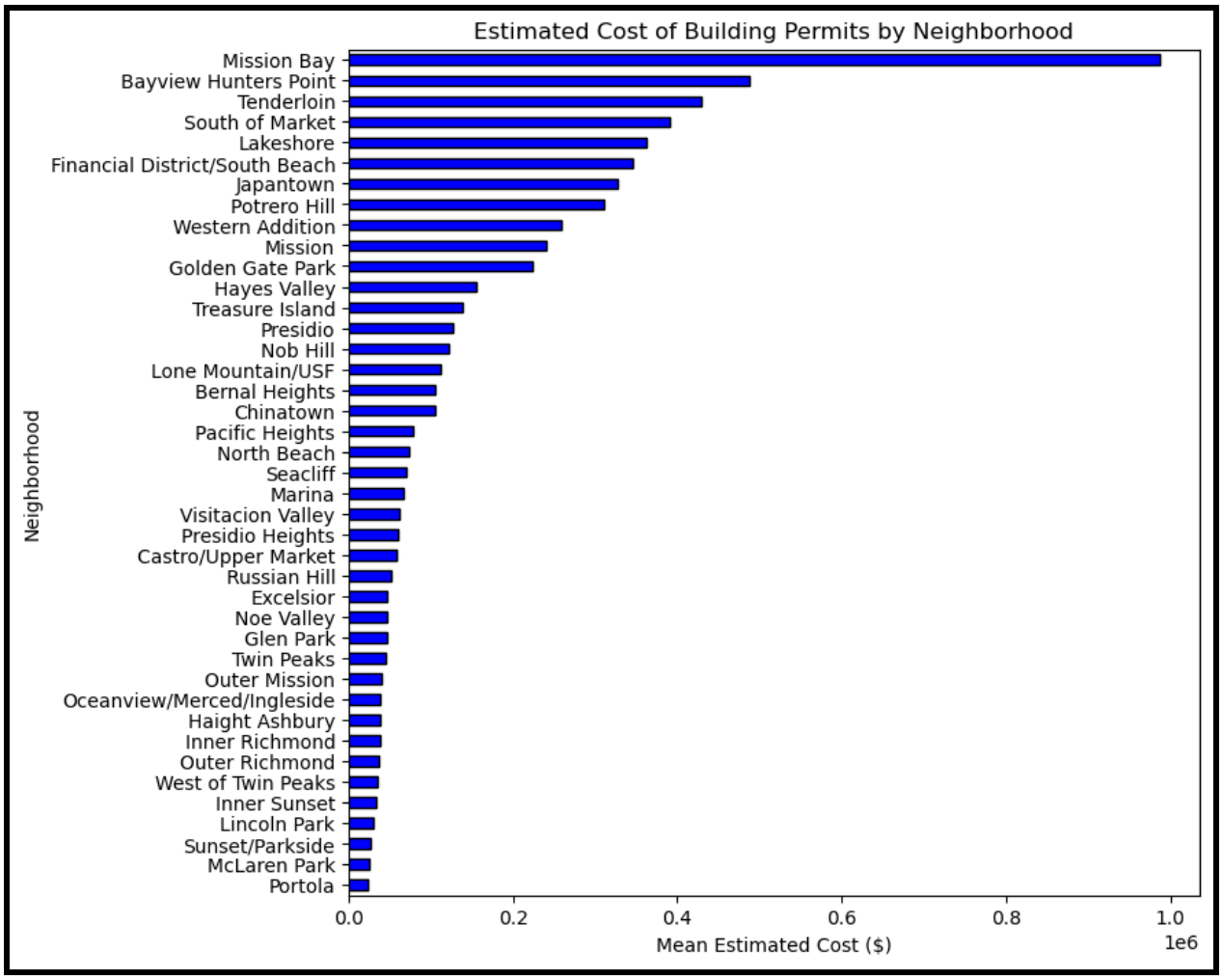


*Figure 3. Mean Estimated Cost of Building Permits by Neighborhood*

Before I began deciding how to construct my model, I had to manipulate the dataset to prepare it for analysis. I first did some feature engineering to create my target variable—by calculating the integer number of days between the "Filed Date" and "Issued Date" variables. I also converted all date columns to an integer representing the number of days since January 1, 2013 (the earliest date in the dataset). I then discovered that there were tremendous amounts of missing values in the data. Twenty-six percent of all cells in the dataset were empty. To address this problem, I removed every row that was missing a target variable, and removed every column

that was missing over fifty percent of values. The remaining missing values I felt comfortable imputing with SimpleImputer in the preprocessing stage.

**METHODS**

I split the data by using a custom function called MLpipe_KFold_RMSE that looped through five different random states. For each random state, I split the data with train_test_split (80/20 other/test split), created four splits with KFold, made a new pipeline with the given preprocessor information and the name of the desired ML algorithm. I then used the pipeline, a grid of hyperparameters to tune, and the KFold splits for cross-validation to run RandomizedSearchCV. To score the model, I used root mean squared error (RMSE). I deemed this scoring method to be ideal given that my problem was a regression one, and RMSE is one of the strongest evaluation metrics for regression models. After the function loops through all five random states, it returns a list of the five produced test scores and the models that they came from.

To decide which machine learning algorithms to use, I was limited by the computational power I had access to. The large size of the dataset meant that running SVR or KNN Regressor was unfeasible. I chose to run linear regression with L1, L2, and elastic net regularization, in addition to RandomForestRegressor and XGBoostRegressor for nonlinear algorithms. I felt like this selection of models gave me a wide variety of approaches to test to find the most predictive model for this dataset.

Before I ran MLpipe_KFold_RMSE, I had to create my preprocessing method. I sorted the features I wanted to use in the model into OneHotEncoder, MinMaxScaler, and StandardScaler categories—essentially, categorical features, continuous features bounded by

limits (such as the range of dates in the dataset) and continuous features with no limits. Figure 4 demonstrates which features were sorted under which transformer.

```
onehot_ftrs = ['Permit Type Definition', 'Existing Use',\
               'Proposed Use', 'Neighborhoods - Analysis Boundaries']
minmax_ftrs = ['Permit Type', 'Supervisor District', 'Zipcode']
std_ftrs = ['Number of Existing Stories', 'Number of Proposed Stories',\
            'Estimated Cost', 'Revised Cost', 'Existing Units', 'Proposed Units', 'Plansets']
```

*Figure 4. Features sorted into OneHotEncoder, MinMaxScaler, StandardScaler*

The last decision to make was which parameters to tune for the model and which values to test. Figure 5 showcases which hyperparameters were tuned and at what values.

```
param_grids = {
    'Lasso': {'lasso__alpha': np.logspace(-3, 3, 7)},
    'Ridge': {'ridge__alpha': np.logspace(-3, 3, 7)},
    'ElasticNet': {'elasticnet__alpha': np.logspace(-3, 3, 7),
                   'elasticnet__l1_ratio': [0.2, 0.5, 0.8]},
    'RandomForest': {'randomforestregressor__n_estimators': [10, 50, 100],
                     'randomforestregressor__max_depth': [10, 20]},
    'SVR': {'svr__C': np.logspace(-3, 3, 7),
            'svr__gamma': ['scale', 'auto']},
    'KNN': {'kneighborsregressor__n_neighbors': [1, 3, 5, 10]},
    'XGBoost': {'xgbregressor__n_estimators': [10, 50, 100],
                'xgbregressor__max_depth': [10, 20]}
}
```

*Figure 5. Hyperparameter grid and values*

The decision to tune these particular hyperparameters was based on some research with regards to the most effective hyperparameters to tune for each model, but also constituted trial and error, with a desire to sample a wide range of values and types of parameters to raise the predictive power of each model.

## RESULTS

The baseline RMSE for the dataset was 91.0615. The performance of each tested model can be seen in Figure 6 below. The figure shows that not all of the tested models outperformed this baseline metric. Two of the three linear regression models tested had higher error than baseline and enormous standard deviations, implying immense underfitting of the model. From this analysis, I can completely discard those two linear models, as they do not at all explain the dataset and carry no predictive power whatsoever. However, the two non-linear models, RandomForestRegressor and XGBoostRegressor, as well as the linear regression with L2 regularization, both outperform baseline. All of these models also have very low standard deviations. XGBoost and RandomForest both perform exceptionally well on the dataset. In other instances of fitting the models, L2 has performed poorly, so I will stick to the two non-linear models. I opted to use an XGBoost model to perform further analysis on the dataset; although Random Forest is probably the best-performing model, I chose XGBoost given its pretty strong performance and quicker runtime for repeated iterations.

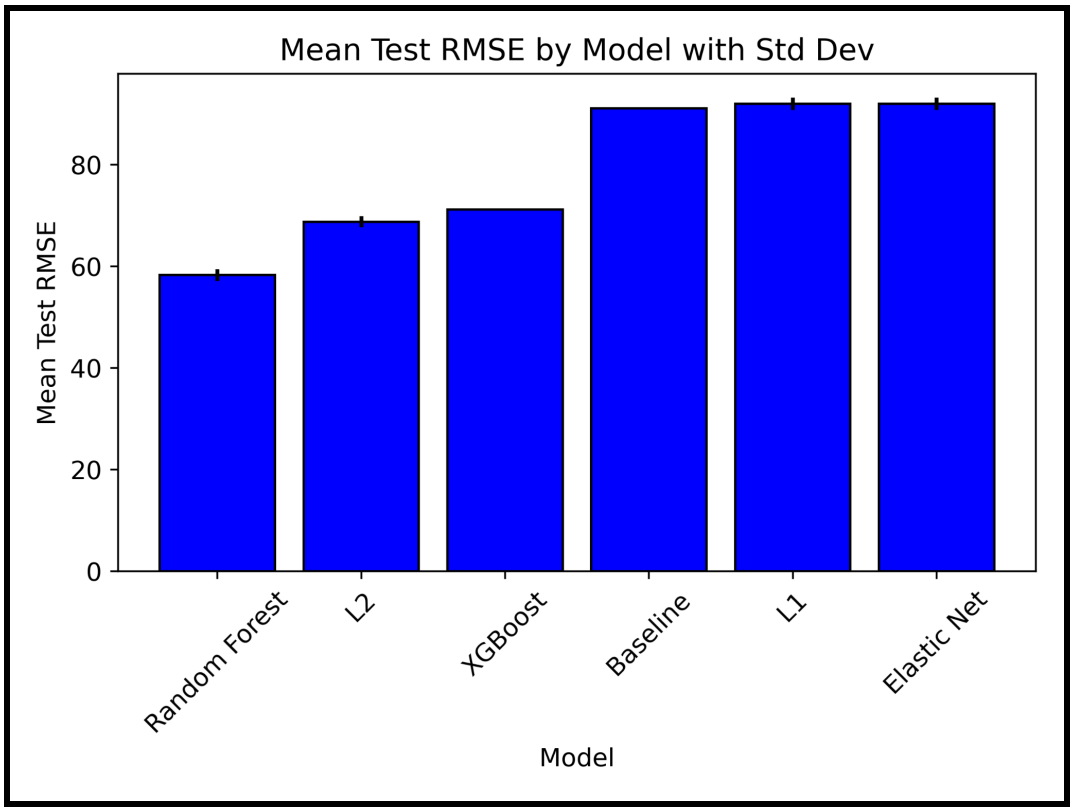


*Figure 6. Performance of each model on dataset*

To investigate global feature importance, the first metric I calculated was permutation feature importance. The results can be seen in Figure 7.

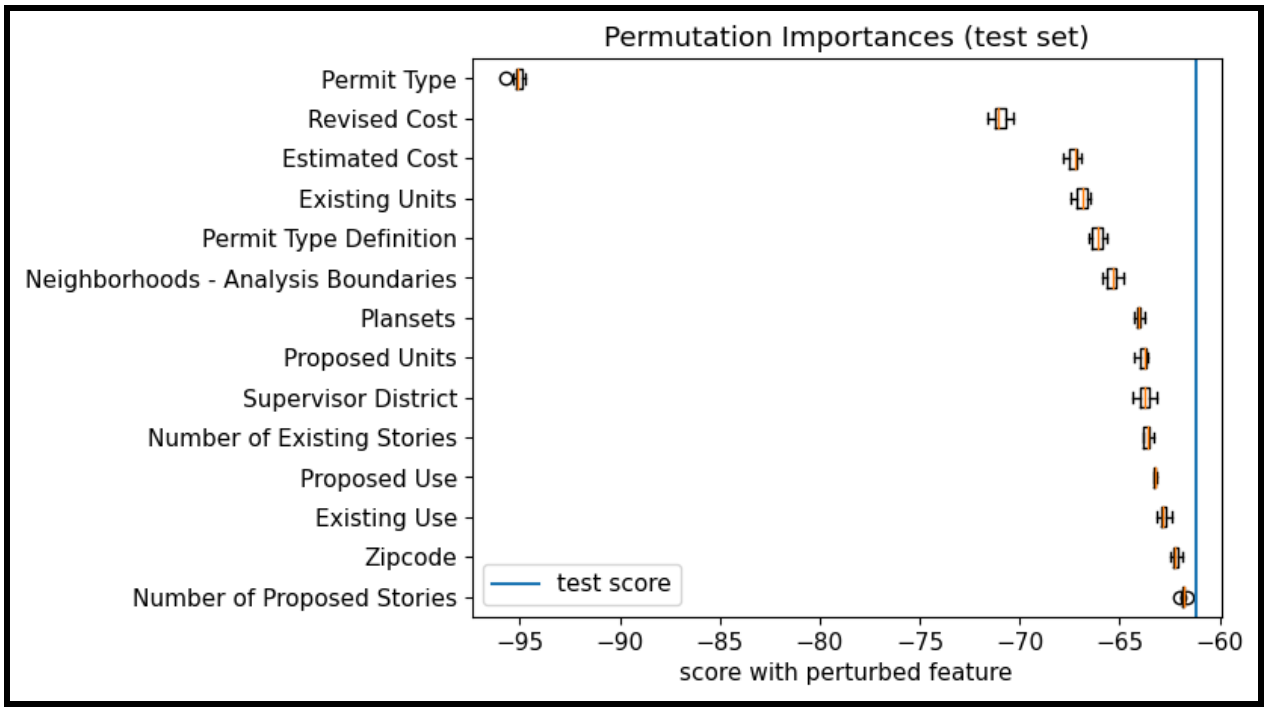


*Figure 7. Permutation feature importance (global)*

This graphic shows that by this metric, the overwhelmingly most important global feature is Permit Type, having a sizably larger score than any other feature.

Figure 8 shows global feature importance calculated using XGBoost's get_score feature. This metric seems to indicate that Estimated Cost and Revised Cost are the most globally important features, two features that also featured very highly on the previous measurement of global importance. Permit Type is the sixth most important global feature in XGBoost's calculations.

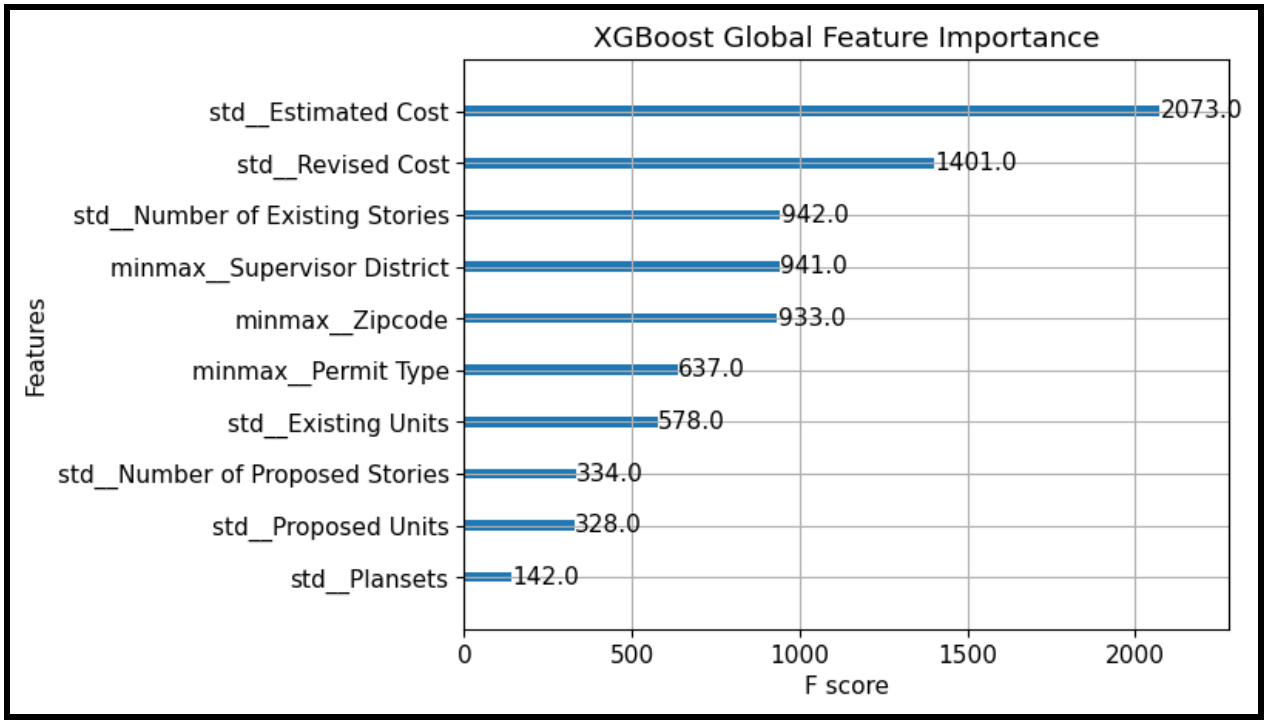


*Figure 8. Global feature importance as determined by XGBoost's get_score()*

A third way that I measured the global feature importance of my XGBoost model is through a SHAP summary plot. The results of this analysis can be seen in Figure 9. Once again, the most important feature is Permit Type. Revised Cost and Estimated Cost also are pretty predictive once more.

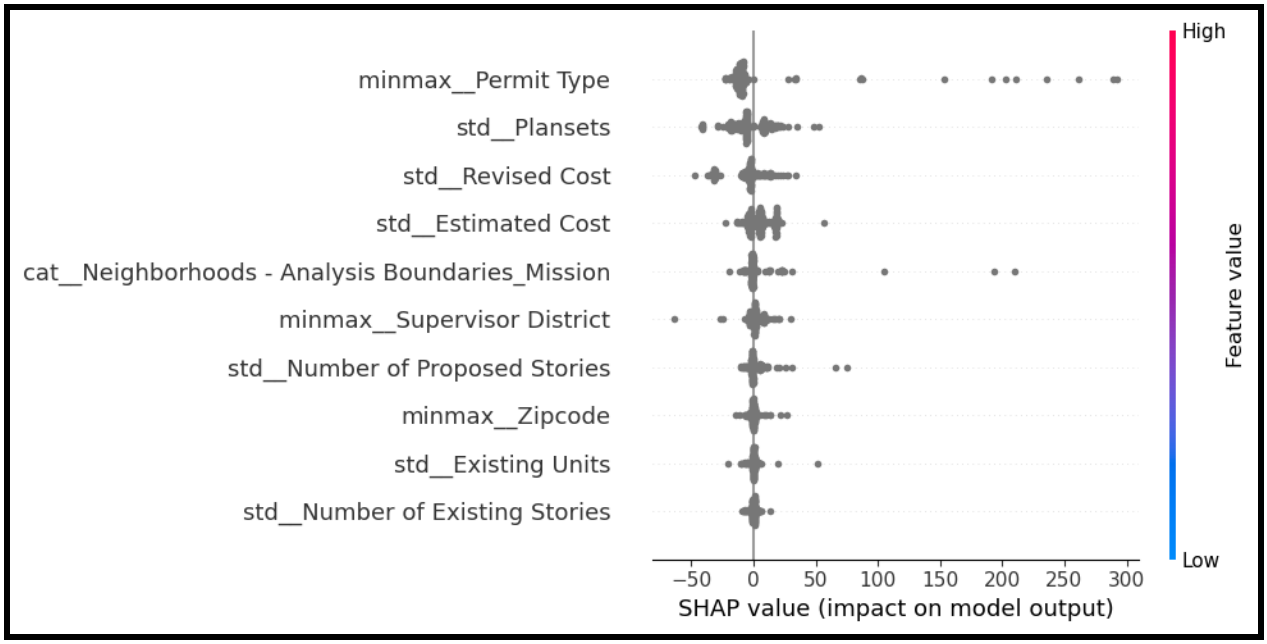


*Figure 9. Global feature importance as determined by SHAP values*

In addition, I sought to explain a sample datapoint to understand local feature importance. The resulting SHAP force plot, for index 55, can be seen in Figure 10.

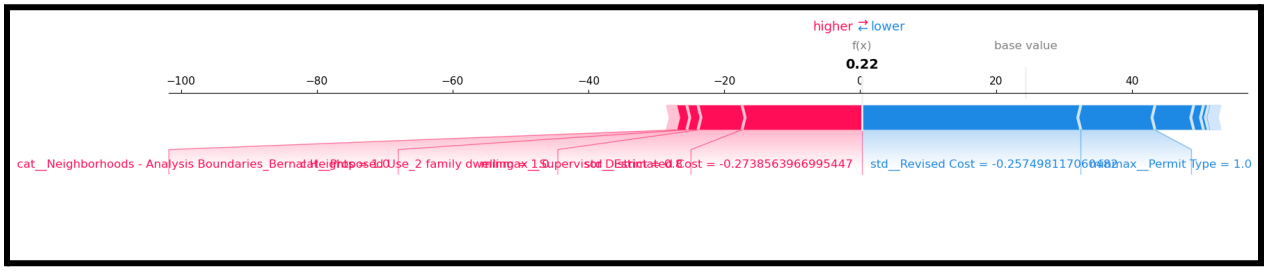


*Figure 10. Local feature importance for data point with index 55*

Supporting the analysis of global feature importance, Estimated Cost and Revised Cost are the two most important features for this particular data point.

It is a safe takeaway from these three analyses of global feature importance and the supporting analysis of local feature importance that the three most important features are Permit Type, Revised Cost, and Estimated Cost. It makes sense that all of these features would be predictive, as perhaps certain permit types or varying-cost developments would have different

speeds of being issued. It stands to reason that more expensive developments might take longer to be approved, given the increased importance of the development to the city. With regards to Permit Type, there are eight different permit types in the dataset, some of which correspond to minor changes like “wall or painted sign” or “sign - erect”, but others of which correspond to major changes like “new construction” or “demolitions”. For the latter category of permit types, it makes sense that permit issuance would take longer.

**OUTLOOK**

For future analysis, it could be interesting to apply geocoding techniques to more directly examine changes in predictions across the city. This could highlight systemic inequities in San Francisco’s permit allotment process, but could also help paint a map of development priorities. In addition, access to greater computing power could allow me to try algorithms like SVM or KNN, which could perhaps have a higher performance on the dataset, or to expand the dataset to include all 1.2 million records in the entire dataset, which could give a more holistic view of the data and allow for stronger predictions.

**REFERENCES**


Publications

Cutler, Kim-Mai. “So You Want to Fix the Housing Crisis.” *TechCrunch*, 4 Nov. 2014, techcrunch.com/2014/11/02/so-you-want-to-fix-the-housing-crisis/.

*Bizjournals.Com*, www.bizjournals.com/sanfrancisco/news/2017/04/28/san-francisco-seattle-housing-production-pipelines.html. Accessed 20 Dec. 2023.

“City Building Permit Delays Costing Developers Time and Money.” *Business in Vancouver*, 19 Dec. 2017, biv.com/article/2014/11/city-building-permit-delays-costing-developers-tim.

Data Sources

https://data.sfgov.org/Housing-and-Buildings/Building-Permits/i98e-djp9/data

Previous Work

*627 Days, Just for the Permit: This Data Shows the Staggering Timeline ...*, www.sfchronicle.com/sf/article/housing-permits-san-francisco-17652633.php. Accessed 21 Dec. 2023.