Final Project

Module 6 Group Assignment

Neil Mascarenhas, Tejal Ambilwade, Sagar Mordiya

College of Professional Studies, Northeastern University

ALY 6040 Data Mining, Fall 2021

Prof: Justin Grosz

Dec. 05th, 2021

Abstract

The purpose of this paper is to go deeper into the Boston property assessment dataset. The Boston property assessment records all residential and non-residential properties in the city. The Boston Globe reported in May 2021 that the Boston housing market is highly competitive, resulting in soaring costs. Alongside, People are seeking more large homes as the pandemic persists. Because most property managers and realtors cannot show their properties to multiple persons, it becomes more difficult to find houses. This article aims to assist those individuals, realtors, and real estate agents, by providing an approximate price for the home they are looking for. We choose to utilize a few fundamental machine learning ideas to assist in determining the optimal selling price for the home, given specific preferences on the number of rooms, location, style, and other details regarding the bath and kitchen kinds. We will focus exclusively on residential property, as it is more in demand now. The overall purpose of this study is to build on earlier EDA (Exploratory Data Analysis) work by developing predictive models that address our business problems. Finally, optimizing the model's performance to achieve efficient results.

Analysis

We must first prepare the data to proceed with the analysis and modeling. It comprises determining whether the dataset has been cleansed, whether any values are missing, duplicate values, and any outliers, also known as extreme cases, by running sanity tests. The purpose of this paper was to develop a prediction model that can predict the estimated price of houses in Boston based on a variety of crucial variables. Answering the questions and making recommendations will be easier if we better understand these features. We have chosen '**TOTAL_VALUE**' as our dependent variable in this study, predicting its value using various models.

Moreover, we will use Linear Regression, Random Forest, LightGBM Regression, Ridge and LASSO regression to develop predictive models. After that, we will analyze the model's performance in terms of Accuracy, MSE (Mean Squared Errors), ROC AUC Score, Precision, Recall, F1, and the speed with which it can be executed for each model. These scores will determine which model is more suitable for our dataset to predict the property's value. Lastly, we intend to build additional optimized models using feature engineering techniques by performing feature selection methodologies. This way, we will find out the elements responsible for the changes in the Boston Housing market and which factors need to be considered for the total value of the house.

Cleaning and Exploratory Data Analysis

To check the data's cleanliness, we first look for duplicate values. We have 3 ID columns. However, each PID must be unique. This column will help us find duplicate records in the dataset. We detected 176 duplicate ids in this column with the same data. We dropped these records because they waste space, time, and resources. Next, we needed to know if the dataset had any missing values. We discovered missing values in **56 of 63** columns. Fixing this dataset would be difficult. The initial step was to delete columns with over **50%** missing values. We did this because using less than 50% of the data to fill in the missing values is statistically inefficient.

Further examination revealed that several of the columns should be blank. For example, when the bedrooms count was **1**, the first and second columns were empty. It was the same for bathroom and kitchen columns. Among the remaining columns, AC had the most missing data. We discovered that properties without air conditioning had these values blank. We could

understand lacking data. Therefore, we dropped the records. It benefited us. Now we have all missing data below **0.2 percent**. We removed those records since they had no impact on data analysis.

Action on values in *Mail_State* Column.

Our research focuses on Boston and its suburbs. This column featured state records like NY for New York and CA for California. We wanted Boston data because it is in Massachusetts. All records except MA were dropped because they did not cover business **(less than 729)**. All other values are gone. Therefore, all had MA. This column is useless for evaluating or anticipating anything. Hence, we removed it.

Action on values in *BLDG_SEQ and NUM_BLDGS*

We counted each column's value to check if it was skewed. Throughout the dataset, the **BLDG SEQ** column has only one data value. It is the same with the Mail State column. Our acts will be similar. Furthermore, The **NUM BLDGS field has 99.98%** one values. It signifies that the column is skewed. We decided to remove these columns because they add nothing to our analysis or prediction.

Converting the currency values to float

We saw currency values in columns. That is how they were recorded. It was a string object. We intended to show this as afloat. In addition, **TOTAL VALUE** is our dependent characteristic. We needed to remove the $ sign from the string value to make it invalid. We modified **$719,400.00**

to 719400.0. Finally, we eliminated the 3 ID columns as they were no longer needed to fill in other information.

While performing EDA, we realized that the dataset now has **19 features and 124,101** records with *city names, zip codes, apartment type, property value,* and *room count*. AC Type was a fascinating column. Figure 1 shows that there is just one value for **'Yes'** and that **62.78** percent of values are None. It worries us because it hinders our recommendations.

Like the Mail State column, this one caught our eye and made us ponder. It had county values. Strange since the data was limited for Boston properties. Not that we know what the correct settings are. We can see that most housing details are present for Boston, Dorchester, and South Boston after filtering the dataset for Massachusetts. **Boston (123942.09), Brookline (105786.62), and Roxbury Crossing (93781.11)**. We need to know this to plan for the outcome and make forecasts. As the values are slanted in different proportions, this may be possible and beneficial in prediction modeling.

Data Preparation

The model has categorical (discrete) and numerical (continuous) features. Preparing data for modeling ensures it is in the correct format. First, eliminate all columns with significant discrete values that hamper the modeling process. Other *BLDG_VALUE, TOTAL_VALUE, 'CITY,' YR_BUILT, EXT_COND, KITCHEN_TYPE, HEAT_TYPE, AC_TYPE are* the eight columns in this table. The label encoding will be used to select and include significant category variables. We

adopted this method because each feature had many values, had made a dummy variable approach, we would have ended up with many columns to work with.

Moreover, it would help us decode the variable names at the end. The next step is to bin (or bucket) the continuous variables in the dataset. We did this because binning improves prediction model accuracy. They were *BLDG_VALUE and TOTAL_VALUE* features that required binning. For the *BLDG_VALUE* feature, we created five bins with all the values distributed into the ***BLDG_VALUE*** feature for the ***BLDG_VALUE*** feature. Similarly, for ***TOTAL_VALUE***, we created ten bins.

We did this because we need to work with predictive variables that predict the property's value close to the real world. We tried many bin sizes before settling on this one. We dropped the initial columns because they were no longer needed. It would have caused multicollinearity. Next, we performed a VIF test to determine multicollinearity in the dataset. We found that there were many variables with VIF values above 10. We decided to drop those values as they will adversely affect the model performance and create issues with model prediction. Then we will split the data into train and test sets. This data will help us to evaluate the model's Accuracy. We utilized a **70:30 ratio** to feed the model (train) data and then test. Thus, *83725 records for training and 35883 for testing* were obtained.

Linear Regression

After building a Linear Regression model, we identified critical categorical variables linear. From **Figure 6,** we see that the P-value of each variable; **CITY, YR_BUILT,**

EXT_COND, KITCHEN_TYPE, and **BLDG_VALUE_bins**, are essential variables. ***KITCHEN_TYPE*** has the highest co-efficient value, followed by ***AC_TYPE*** and ***HEAT_TYPE***. According to this model, ***KITCHEN_TYPE*** is most important in predicting the estimated costs of the property in Boston. **Figure 7** graph shows the confusion matrix generated using Linear Regression. Using this confusion matrix, we calculated the model's Accuracy as **77.18% within 0.04299 seconds** and other performance parameters shown in **Figure 6.** We opt for feature selection to optimize the model. We utilized Sequential Feature Selector to select backward.

Furthermore, we chose this strategy because it is commonly preferred. Here it shows us that a property price is affected by their ***'KITCHEN_TYPE,' 'BLDG_VALUE_bins***. We used these results because they were better optimized.

Random Forest (Regressor)

Next, we performed Random Forest regressor modeling to uncover key categorical variables. **Figure 9** displays the feature importance for each variable. ***BLDG_VALUE_bins, CITY, YEAR_BUILT, and KITCHEN_TYPE*** were all essential features this model has come up in predicting. According to this model, the ***BLDG_VALUE_bins*** is the most crucial factor in deciding the property's price in Boston. The confusion matrix created by this model prediction is shown in **Figure 9**. We computed the model's Accuracy of 69.23 percent within 1.8609 seconds execution time and other performance aspects using this confusion matrix displayed in Figures **8 and 10**.

LightGBM Regression

Last, we ran Gradient Boosting Machine (GBM) modeling. We observed that the modeling was quite time-consuming and opted to run the **'light'** version. This LightGBM is significantly faster and delivers us the same results. We fitted the model where we uncovered key categorical variables. Figure 10 indicates the feature importance for each variable. ***CITY, KITCHEN TYPE,*** and ***BLDG VALUE bins*** were all crucial features this model has come up with in forecasting. According to this model, the ***CITY*** is the essential component in deciding the property's price in Boston. The confusion matrix obtained by this model prediction is displayed in Figure 8. We determined the model's Accuracy is 86.49 percent within 1.8609 seconds execution time and other performance factors using this confusion matrix given in **Figures 12 and 13.**

Model Regularization

How do we know that the results we got are stable and that the selected features will be consistent across all the other generalized models? This prevents the model from overfitting and underfitting or getting incorrect predictions. We decided to perform Ridge and Lasso Regression to ensure that the results were consistent or stable.

Ridge Regression

As we forward, we have tried to perform the ridge regression model on the data as we have the dependent variable in the continuous variable. The Ridge model is suitable for the highly correlated elements. The elements we are working on have more correlation, making this model perfect for the dataset. After performing the model, we found the model's Accuracy is **78.67 %**, which is more. We can say that the model performed well for the dependent variable. We have performed backward feature selection to get into deep analysis to know which elements have more

impact. By analyzing, we found that ***KITCHEN_TYPE*** and ***BLDG_VALUE_bins*** highly impact the house price as they are more important while considering the house prices.

LASSO Regression

Later, we tried to perform the LASSO model, which helps obtain the subset of predictors with minimum error and maximum Accuracy. It can help us know which features have the most impact. Moreover, after performing the LASSO on the dataset, we found that the LASSO model performed very well with the Accuracy of **78.59%.** It is second best after the logistic regression model we have performed and has the mean squared error of **40.96%.** It is relatively less than the other models we have performed. After looking at the model's performance, we can say that the model performed very well while performing the backward feature selection. We found that the LASSO and Ridge model has given the same impactful feature, ***KITCHEN_TYPE*** and ***BLDG_VALUE_bins***, but with slightly different Accuracy of both models.

Model Evaluation

Finally, we will evaluate the developed model and optimize it to increase performance scores to see which one is better and suggest to the business team. The Light Gradient Boosting Model has the best **Accuracy (86.49%)** compared to other models. Linear Regression (**77.18%**) and Random Forest (**69.23%**) performed with the lowest Accuracy. We chose this Light Gradient Boosting model over others despite the dataset's biases. We re-created the model using the selected feature using the stepwise selection method. Our results were that the model accuracy score increased slightly with regularization. For Ridge, we got (78.67%) whereas Lasso was (**78.59%**)

figures 14 and 15. It computed the feature importance for the given variable ***KITCHEN_TYPE*** and ***City***.

Conclusion

This paper sets the capstone and provides the finishing touch for the papers we worked on in the final project module. We have a solid understanding of the dataset and have prepared it through analysis. Based on the results of the predictive research, we have discovered that the housing prices are directly related to three factors: the value of the building, the location of the house, and the type of kitchen. When we look at the criteria, the building value and the location are the most important ones that can significantly impact the price. However, we cannot focus on that as a realistic prospect for the house prices. The type of kitchen offered in units is the most important factor that a property manager should consider. We saw that the kitchen type is driving the price since we categorized the kitchen values; we can say that having a well-furnished kitchen will significantly increase the property's price in Boston. If a property manager wants to make money and boost profit, he or she can change the kitchen type of houses priced lower than others to do so. The first thing that property management should look at is the type of kitchen available in the units. To generate income and raise the property's value, the manager can modify the type of kitchen available in the apartments that are less expensive than the others.

Reference

40 Techniques Used by Data Scientists. (2020). Data Science Central. https://www.datasciencecentral.com/profiles/blogs/40-techniques-used-by-data-scientists

Bhattacharyya, S. (2020, September 28). *Ridge and Lasso Regression: L1 and L2 Regularization*. Medium. https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b

Brendel, C. (2021, December 14). *Quickly Compare Multiple Models - Towards Data Science*. Medium. https://towardsdatascience.com/quickly-test-multiple-models-a98477476f0

Brownlee, J. (2021, April 27). *How to Develop a Light Gradient Boosted Machine (LightGBM) Ensemble*. Machine Learning Mastery. https://machinelearningmastery.com/light-gradient-boosted-machine-lightgbm-ensemble/

ConvergenceWarning: lbfgs failed to converge (status=1): STOP: TOTAL NO. of ITERATIONS REACHED LIMIT. (2020, June 30). Stack Overflow. Retrieved December 5, 2021, from https://stackoverflow.com/questions/62658215/convergencewarning-lbfgs-failed-to-converge-status-1-stop-total-no-of-iter

Duca, A. L. (2021, October 24). *Data Preprocessing with Python Pandas — Part 5 Binning*. Medium. https://towardsdatascience.com/data-preprocessing-with-python-pandas-part-5-binning-c5bd5fd1b950

How can I determine the optimal binning system for a continuous variable in Python? (2020, December 8). Cross Validated. Retrieved December 5, 2021, from https://stats.stackexchange.com/questions/499941/how-can-i-determine-the-optimal-binning-system-for-a-continuous-variable-in-pyth

Malik, U. (2021, December 1). *Principal Component Analysis (PCA) in Python with Scikit-Learn*. Stack Abuse. Retrieved December 3, 2021, from https://stackabuse.com/implementing-pca-in-python-with-scikit-learn/

Miller, T. W. (2021). *Modeling Techniques In Predictive Analytics With Python And R: A Guide To Data Science* (1st ed.) [E-book]. Pearson Education.

N. (2021, October 29). *Key data science modeling techniques used in data evaluation and analysis*. Selerity. https://seleritysas.com/blog/2021/01/22/key-data-science-modeling-techniques-used-in-data-evaluation-and-analysis/

sklearn.feature_selection.SequentialFeatureSelector. (2010). Scikit-Learn. Retrieved December 4, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html

sklearn.linear_model.LogisticRegression. (n.d.). Scikit-Learn. Retrieved December 4, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

statsmodels Principal Component Analysis — statsmodels. (n.d.). StatsModel. Retrieved December 4, 2021, from

https://www.statsmodels.org/dev/examples/notebooks/generated/pca_fertility_factors.html

What is the difference between pandas.qcut and pandas.cut? (2015, May 13). Stack Overflow. Retrieved December 5, 2021, from https://stackoverflow.com/questions/30211923/what-is-the-difference-between-pandas-qcut-and-pandas-cut

Wijaya, C. Y. (2021, October 12). *5 Feature Selection Method from Scikit-Learn you should know*. Medium. Retrieved December 5, 2021, from https://towardsdatascience.com/5-feature-selection-method-from-scikit-learn-you-should-know-ed4d116e4172

Appendix

Figure 1: *AC_Type* Columns

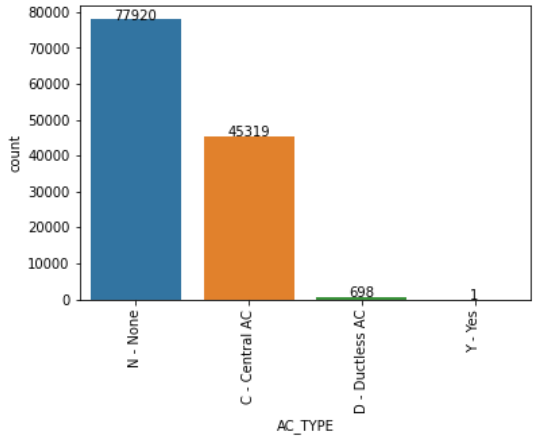

Figure 2: City Column

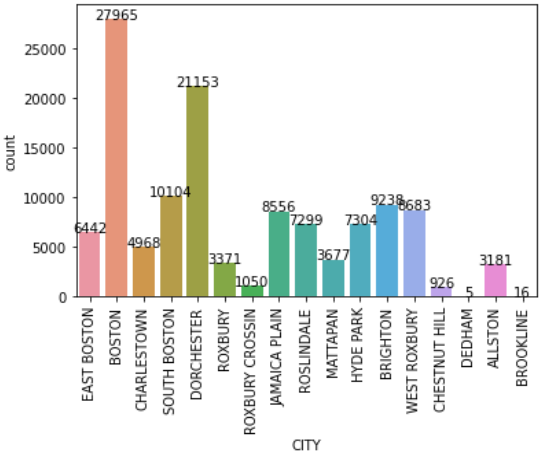

Figure 3: Building type column, value counts

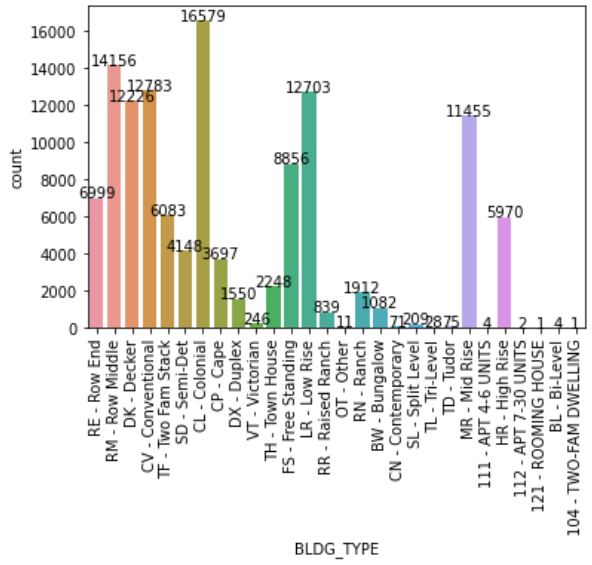

Figure 5: Logistic Regression filtered output

	pval	coef
CITY	0.00	0.01
YR_BUILT	0.00	0.00
EXT_COND	0.00	0.00
KITCHEN_TYPE	0.00	-0.07
HEAT_TYPE	0.00	0.02
AC_TYPE	0.00	0.07
BLDG_VALUE_bins	0.00	1.04

Figure 4: Logistic Regression output

OLS Regression Results

Dep. Variable:	TOTAL_VALUE_bins	R-squared (uncentered):	0.964
Model:	OLS	Adj. R-squared (uncentered):	0.964
Method:	Least Squares	F-statistic:	2.751e+05
Date:	Sun, 05 Dec 2021	Prob (F-statistic):	0.00
Time:	23:35:42	Log-Likelihood:	-41295.
No. Observations:	71756	AIC:	8.260e+04
Df Residuals:	71749	BIC:	8.267e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
CITY	0.0097	0.000	29.954	0.000	0.009	0.010
YR_BUILT	0.0015	3.34e-05	45.541	0.000	0.001	0.002
EXT_COND	0.0037	0.001	2.854	0.004	0.001	0.006
KITCHEN_TYPE	-0.0713	0.001	-117.299	0.000	-0.073	-0.070
HEAT_TYPE	0.0216	0.001	30.460	0.000	0.020	0.023
AC_TYPE	0.0670	0.002	35.374	0.000	0.063	0.071
BLDG_VALUE_bins	1.0431	0.002	569.580	0.000	1.040	1.047

Omnibus:	268.274	Durbin-Watson:	1.996
Prob(Omnibus):	0.000	Jarque-Bera (JB):	277.699
Skew:	-0.135	Prob(JB):	4.99e-61
Kurtosis:	3.143	Cond. No.	172.

Figure 7: Confusion Matrix Linear Regression.

```
== Logestic Regression ==
Confusion Matrix :
 [[ 6521  4233     6     0     0]
 [ 1304 21565  2914     0     0]
 [    3  2718  5116   166     0]
 [    0    38   833  1401    15]
 [    0     1    52   274   678]]
Test accuracy =  0.7375099293448723
Mean Squared Error =  0.2688657552573268
```

Figure 8: Confusion matrix for Random Forest.

```
array([[ 9597,  1163,     0,     0,     0],
       [ 7020, 18498,   265,     0,     0],
       [    6,  3969,  3972,    56,     0],
       [    0,    65,  1004,  1218,     0],
       [    0,     2,    95,   908,     0]],
```

Figure 9: Selected features Random Forest

```
Variable: BLDG_VALUE_bins      Importance: 1.0
Variable: CITY                 Importance: 0.0
Variable: YR_BUILT             Importance: 0.0
Variable: EXT_COND             Importance: 0.0
Variable: KITCHEN_TYPE         Importance: 0.0
Variable: HEAT_TYPE            Importance: 0.0
Variable: AC_TYPE              Importance: 0.0
```

Figure 10: Confusion matrix for Random Forest.

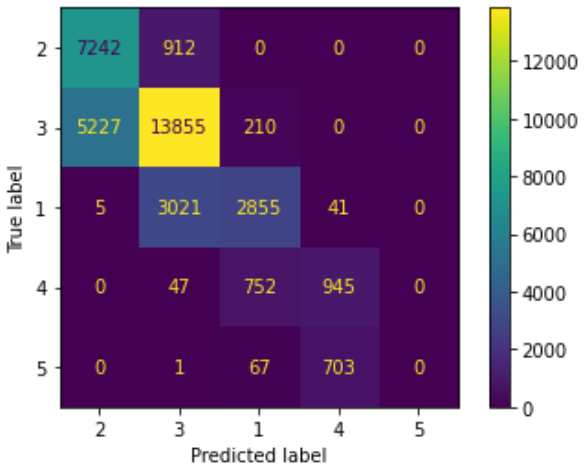

Figure 11: Selected features LightGBM model

```
Variable: CITY                 Importance: 864
Variable: BLDG_TYPE            Importance: 495
Variable: BLDG_VALUE_bins      Importance: 348
Variable: KITCHEN_TYPE         Importance: 341
Variable: EXT_FINISHED         Importance: 261
Variable: BTHRM_STYLE1         Importance: 185
Variable: KITCHEN              Importance: 176
Variable: EXT_COND             Importance: 162
Variable: HEAT_TYPE            Importance: 87
Variable: AC_TYPE              Importance: 81
```

Figure 12: Confusion matrix for LightGBM

```
== LightGBM Regression ==
Confusion Matrix :
 [[ 6179  1971     4     0     0]
 [  463 18056   773     0     0]
 [    0  1377  4213   332     0]
 [    0     4   188  1498    54]
 [    0     0     2    69   700]]
Mean Squared Error =  0.14678259900231308
```

Figure 13: Confusion matrix for LightGBM

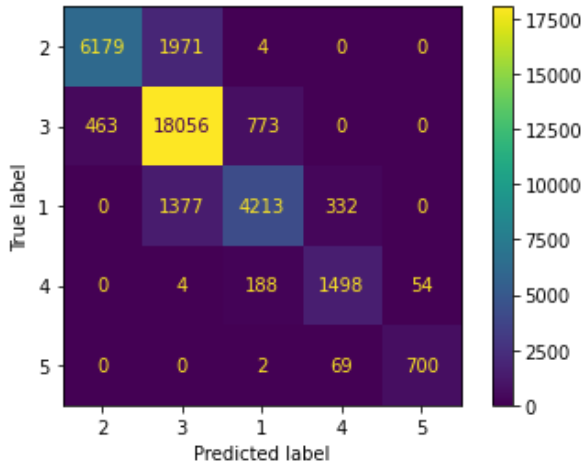

Figure 14: Confusion matrix for Linear Regression

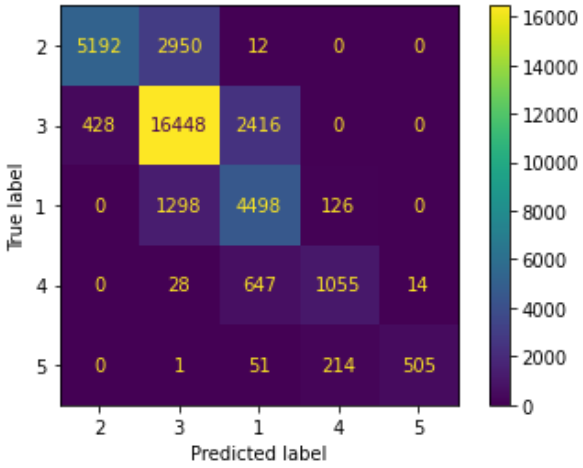

Figure 15: Model Comparison Table

	Model	Accuracy
2	LightGBM	0.86
3	Ridge Regression	0.79
4	Lasso Regression	0.79
0	Linear Regression	0.77
1	Random Forest	0.69

Figure 16: Model Comparison Bar plot