DSE511-Project 3: California Housing Prices Prediction

Albina Jetybayeva[1], Pragya Kandel[1], Isidora Fletcher[1], Amirehsan Ghasemi[1]
[1]The University of Tennessee, Knoxville, The Bredesen Center for Interdisciplinary Research and Graduate Education, Knoxville, TN, USA, 37996-3394

ABSTRACT In this project we targeted modeling and understanding the factors contributing to housing prices in California. The approach applied to this study was supervised regression modeling. Before modeling, we used the Exploratory Data Analysis (EDA) to observe the preliminary feature importances and made decisions on the data preprocessing, which defined the model performance. In our case, the observation of price distribution helped to notice the “capped” price values, the removal of which improved all the models performances significantly. After EDA we preprocessed the data by cleaning the missing values, features scaling, and categorical feature handling. Then we used Linear Regression (LR) as a baseline in addition with Lasso, Ridge and ElasticNet, Support Vector Regression (SVR), Decision Tree, and Random Forest Regression for modeling. These models helped to predict the housing prices based on the combination of the various variables. Among different models, the Random Forest performed the best. The hyperparameters of ‘max_features': 4 and 6, 'n_estimators': 30 contributed to better performance and lowered the root mean squared error to $43,000-$44,000. Additionally, we found that the most important features to define the prices are income and location/ocean proximity. The higher income and the closer proximity to the ocean implied purchases of more expensive houses. Furthermore, the other important features were population per household and bedroom per room. It was expected as larger households and larger number of bedrooms per room implies the bigger and more expensive properties and thus, higher prices..

INDEX TERMS Machine Learning, Decision Tree, Linear Regression (LR), Random Forest Regression, Support Vector Regression (SVR), House Price Prediction

I. INTRODUCTION

The new computing technologies have widened the scope of machine learning to a great extent. It’s ability to learn from previous computations and independently adapt to new data is making it popular across various disciplines. Various sectors such as business, bioinformatics, computer engineering, pharmaceuticals, medical, climate change, and statistics are using machine learning models to gather knowledge and predict future events [1]. One of the important sectors that machine learning can be used is on real estates to predict the prices of houses. Buying a new house is always a big decision. It gets affected by various factors such as location, size of house, quality of house, future trading price, school zone etc. but prioritizing these factors is tough [2]. What would be more important? Is it the location or quality of the house? Machine Learning (ML) can be used to ease the process of decision making by forecasting the house prices with maximum accuracy of the market trend and by building a model based on historical dataset [3]. What happened in the past and what was important, how it affected the price and what is going to happen? Prediction of house prices is not only limited to homeowners, but also equally important to real estate agents, appraisers, mortgage lenders, brokers, property developers as well as investors. The prediction of housing prices using ML is not a new concept. The selling price of houses of Pitt county, North Carolina was predicted by the use of parametric and semi parametric regression [4]. Bae and Park, 2015 used a machine learning algorithm for predicting the prices of houses in Fairfax county of Virginia. The ML approach to predict prices has also been used in various other countries such as Brazil. Afonso et al. 2019 [5], predicted the housing prices with deep learning and random forest ensemble in Brazil. Although several studies like these have been already done, there are still many interesting relations between prices and different features/parameters that can be investigated. Also, each case of different regions and conditions is unique, so more research is needed to reveal new effects of the specific conditions on housing pricing and how these can be modeled. In this project we are trying to understand the factors contributing to housing prices specifically in California. The chosen dataset on California was used for the machine learning basics introduction in the book by Aurélien Géron ’Hands-On Machine learning with Scikit-Learn and TensorFlow’ [7]. We hypothesize that the values of the houses will be directly related to the ocean proximity, income values, and inversely related to the housing median age and the population in the area (as it will be considered a more private area). We will also be investigating how the price is affected by the location (longitude, latitude and and ocean proximity), the number of bedrooms and the number of households. This study will be formulated as a supervised regression type problem, and the regression modeling such as Decision Tree, Linear Regression (LR), Random Forest Regression, and Support Vector Regression (SVR) will be used to predict the housing prices based on the combination of the various variables available in the dataset.

II. DATA

For our investigation we used the dataset for California Housing Prices, which contains information on houses from the 1990 California census [7]. This data was helpful to learn the regression techniques. Moreover, this data was chosen as it had an understandable list of variables and the optimal size for ML practice. The raw dataset consisted of 20641 rows and 10 columns. There were 10 columns of self-explanatory features that are shown in Table 1 below:

Table 1. Dataset: Features

Features	Description
Longitude	A measure of how far west a house is; a higher value is farther west
Latitude	A measure of how far north a house is; a higher value is farther north
housing_median_age	Median age of a house within a block; a lower number is a newer building
total_rooms	Total number of rooms within a block
total_bedrooms	Total number of bedrooms within a block
population	Total number of people residing within a block
households	Total number of households, a group of people residing within a home unit, for a block
median_income	Median income for households within a block of houses (measured in tens of thousands of US Dollars)
median_house_value	Median house value for households within a block (measured in US Dollars)
ocean_proximity	Location of the house w.r.t ocean/sea

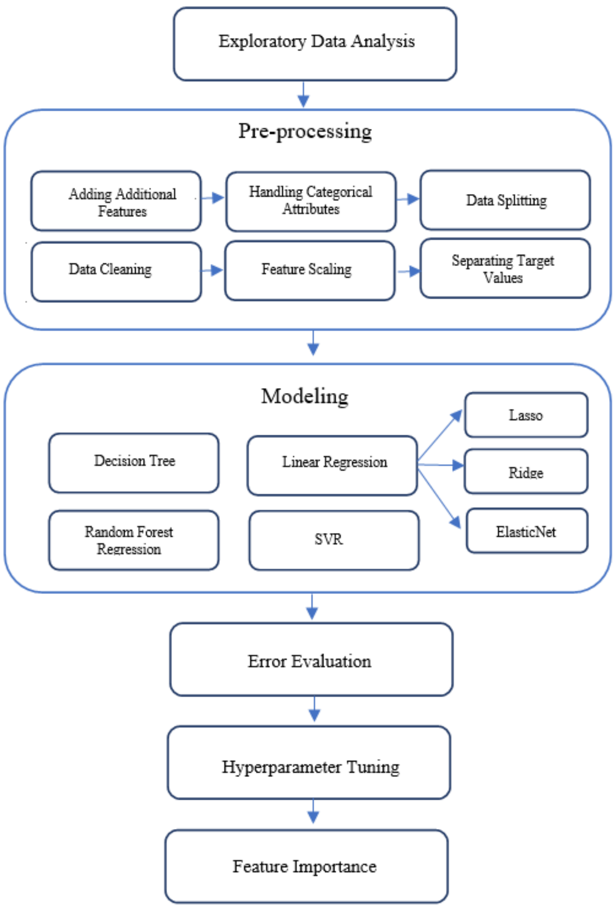

Figure 1. Workflow diagram

III. METHODS

In this project, we solved a machine learning problem. It was a supervised problem, as we used the house prices as labels. As the prices were continuous values, we applied regression for analysis. After loading the raw data, we followed the schematics shown in the following Figure 1. The process started by doing exploratory data analysis (EDA). The EDA consisted of observing how the features were related initially. The highlights of the EDA part will be discussed in the next section. Generally, based on the EDA, it was clear that cleaning the missing values, features scaling and categorical feature handling were required on the data.
First, the categorical attribute “ocean proximity” was handled by assigning the specific number in the gradient order (the lower the number the further away is the house from the ocean). This gradient was chosen for the better and easier interpretation of models’ results (feature importances). Cleaning the missing values was done using the scikit-learn “SimpleImputer” [8] and inserting the median values. Finally, feature scaling was performed on the data, as they had very different ranges, and ML algorithms generally do not perform well when the input numerical attributes have very different scales followed by standardization that results in distribution having unit variance. The advantage of standardization is that it is much less affected by outliers. As scaling the target values is generally not required, these were not scaled.

In order to avoid the data leakage and bias, we splitted the data (80% train and 20% test sets) before cleaning and feature scaling. We used obtained medians, means and standard deviations from the train set in processing the test set. The next step was modeling. The models chosen for this part were Decision Tree, LR (along with Lasso, Ridge and Elastic Net), Random Forest Regression and SVR. We used the tools available with scikit-learn to fit the data. Then, we evaluated the models’ performance by using it on test data for prediction.. After that, we did error evaluation, by using root mean squared error (RMSE), mean absolute error (MAE), model score [9], wall time (train and predict), and cross validation score. These tools helped us to determine how successful the models were and to adjust accordingly. It is important to mention that among all of these model performance evaluation metrics, the main one was chosen to be RMSE. We optimized our models using hyperparameters tuning “GridSearchCV” from scikit-learn (Table A1). Finally, we observed and analyzed the importance of each feature using either coefficients of the models (LR, Lasso, Ridge, Elastic Net) or “.feature_importances_” for Random Forest and Decision Tree. The modeling was also tested on two different datasets: “capped” and “uncapped”. The capped dataset contained all prices from the original file, while the uncapped dataset removed the values of prices >$490,000. The details on this decision will be provided in the EDA sections.

IV. RESULTS AND DISCUSSIONS

IV.I Exploratory Data Analysis (EDA)

First, analyzing the histograms of features, the important observation was done on the houses prices (Figure 2). The large last bin with $500,000 houses was noticed, which is unusual for the datasets of that type . From the code main source description it was found that the median house values were capped, meaning that all the prices which were more than $500,000 were set to be exactly $500,000. That is why there are many houses with $500,000 median_house_value. The capped house value might be a problem for a precise modeling since it was chosen to be the target attribute (labels). The ML algorithms may learn that prices never go beyond that limit or it might be difficult to predict in that price range in general. Thus, it was decided to remove those capped values from the training set (and also from the test set, since the system should not be evaluated poorly if it predicts values beyond $500,000) and compare the models performance on both datasets with and without capped values.
Next, as there was geographical information available (latitude and longitude), mapping of the houses was done with the radius of each circle representing the population and the color showing the houses prices gradient (Figure 3). It was clear that the houses closer to the ocean are more expensive as well as those located in the regions of major cities such as Los Angeles, San Francisco and San Diego, which is expected.

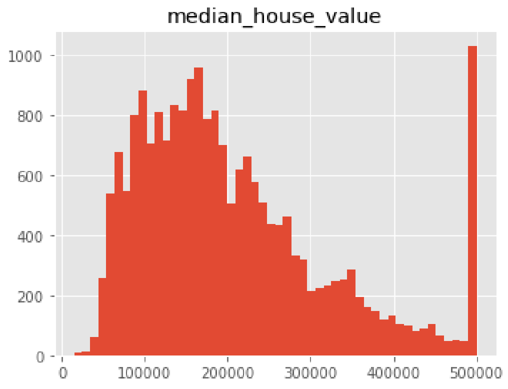

Figure 2. Median house value (prices) distribution.

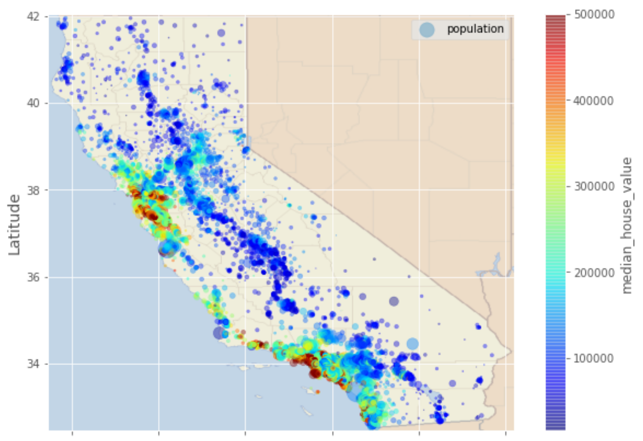

Figure 3. Housing prices map

It could be also useful to try out various attribute combinations. For example, it might be helpful to add the number of rooms per household. Similarly, the total number of bedrooms by itself was not very helpful, so it may be more informative to compare it to the number of rooms. In addition, the population per household also seemed like an interesting attribute combination to look at.
By analyzing the standard correlation coefficient (also called Pearson's r) between every feature and price an important observation was done. A strong positive correlation (0.68) was demonstrated for the prices and income. This is expected, as larger income allows people to buy more expensive houses.
As it was observed above, income and ocean proximity are probably the most important parameters defining the house prices. So to visualize their relations, seaborn pairplot of median income vs median house value with hue as Ocean Proximity was done (Figure 4). It can be seen that "inland" houses mostly lie in the region of lower income and house prices, as expected. Most of the "near bay" and "near ocean" houses are placed closer to the higher income-price region, while "<1 hour drive to ocean" is more dispersed in the middle. These clearly show the main trends, although there are obviously exceptions and other important parameters such as bedrooms per rooms and rooms per household, as well as other attributes not present in the dataset, which need careful consideration for more accurate prediction.

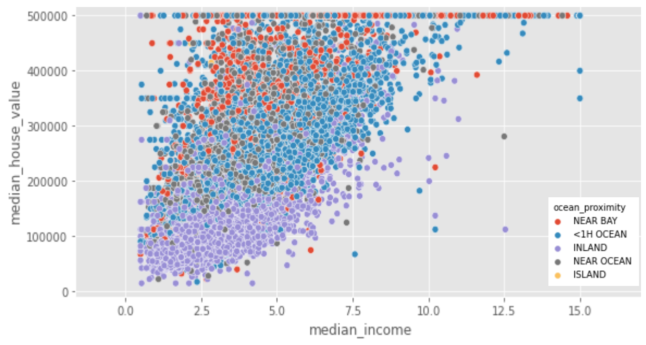

Figure 4. Scatter plot of income vs prices with color coded ocean proximity.

IV.II Modeling

First, the dataset was modeled using the simplest LR as a baseline. LR fits a linear model with coefficients that minimize the residual sum of squares between the observed and predicted by the linear approximation targets in the dataset. Analyzing the results of LR, it is clear that the typical RMSE of approximately $69,000 for capped data was found (Table 2). Considering that most of the prices range between $120,000 and $265,000, this error is quite large. This is an example of a model underfitting the training data. When this happens it might be a result of the features not providing enough information to make good predictions, or that the model is not powerful enough. The main ways to fix underfitting are to select a more powerful model, to feed the training algorithm with better features, or to reduce the constraints on the model. The LR model is not regularized, so this rules out the last option.
However, the other related regression models were tested instead like Lasso, Ridge and ElasticNet [10]. Ridge solves a regression model where the loss function is the linear least squares function and regularization is given by the L2-norm. Lasso, or Least Absolute Shrinkage and Selection Operator, is quite similar

conceptually to Ridge. It also adds a penalty for non-zero coefficients, but unlike Ridge, which penalizes sum of squared coefficients (the so-called L2 penalty), Lasso penalizes the sum of their absolute values (L1 penalty). As a result, for high values of λ, many coefficients are exactly zeroed under Lasso, which is never the case in Ridge. Finally, ElasticNet first emerged as a result of critique on Lasso, whose variable selection can be too dependent on data and thus unstable. The solution was to combine the penalties of Ridge regression and Lasso to get the best of both.

When analyzing these models' results, it can be seen that even after hyperparameters tuning the results were not improving a lot for Lasso and Ridge compared to LR. ElasticNet after hyperparameters tuning showed that the best model would be Lasso's tuned model (Table A1), which was just previously tested and discussed. Thus, testing of the ElasticNet was done with just changing the alpha parameter and the results indicated even worse performance than the baseline (Table 2).

The largest improvement in all the above mentioned models was observed for running the same code on the uncapped dataset (Table 3). The RMSE was reduced by almost $10,000, which is quite significant. The effect can be explained by the fact that by removing the capped prices values, the model was trained better and thus, performed better, as these capped values significantly affected the patterns and their distribution. Moreover, they probably prevented the good generalization of modeling.

Among all above discussed models, Lasso showed the best result on the uncapped dataset. Generally, L1 regularization was found to be better for sparse data, which is our case, that is why the performance is probably the best.

Support vector machines are really powerful tools for classification purposes in machine learning. As the housing prices is a regression problem, Support Vector Regression (SVR) was performed on the dataset in order to predict the continuous values. In the first stage, SVR model was trained using a training dataset and was applied on a test set. While using it on a test set, the root mean square error was of about $98,318 dollars and the MAE of $76,481 dollars. The error is very high compared not only with the range of the house prices, but also with predictions from other models.

The comparison of SVR with linear, RBF and polynomial kernels was done on the data set to evaluate the performance. Upon comparison, the mean square error for linear was low by almost $6000 compared to the RBF. Following the usual SVR prediction, the hyperparameter tuning was performed in order to increase the effectiveness of the model. While performing hyperparameter tuning, in addition to different types of kernel we tried with different values of c, epsilon, and gamma.. The best parameters fro tuning would increase the performance of the SVR model in this data set. Like the comparison above the hyperparameter tuning also predicted that the linear kernel performed better than other kernels. The use of support vector regression came with challenges. The first one being the time. Among all the models, SVR took the longest to run compared to other models. The second one was the error, even after hyperparameter tuning the errors were highest amongst all the models. The third one was calculation of the feature importance. Even when the kernel was set up to be linear, we weren't able to get feature importance. With the strength of the SVR algorithm being low and with an unacceptable amount of run time, we understood that it was the worst model among all models we used in this study.

The Decision Tree algorithm is one of the most used ML approaches for classification and regression problems. It builds regression or classification models by breaking down a dataset into smaller subsets. In this study, the Decision Tree Regressor model using sklearn was applied to the training set. The model used the default parameters to predict the target in the test set and the results were compared with the real values. Generally, as in the previous models, the Decision Tree performed better on the uncapped dataset as shown in Table 2 and 3. For the uncapped dataset RMSE and R^2 score, which evaluates the performance of a regression model, were $62459 and 0.58, respectively.

Also, the model score for both the training and testing set was considered. We attained a 100 % score on the training and for the testing, it was almost 63%. As we mentioned in the previous paragraph, for the first fitting, we didn't do any hyperparameter tuning and the model was overfitting.

To improve the score of the model hyperparameter tuning was utilized. Some important parameters such as minimum sample split, max depth, minimum sample leaf, and maximum leaf nodes were considered. After an exhaustive search over the defined parameters the best estimators for the model were obtained. The results for the best hyperparameters are shown in Table A1. Then the tuned model was fitted to the training set and the wall time was recorded. To speed the computation time n_jobs = -1 was considered. The wall time for the uncapped training set is shown in Table 3. To predict the target, the tuned model with the best estimator was applied to the test set and the results for metrics were recorded. For the uncapped dataset after tuning the hyperparameters, the RMSE and R^2 were reported $54244 and 0.68 respectively which prove that model has improved.

To visualize the results the following scatter plot and error plot plots can be considered (Figure 5).

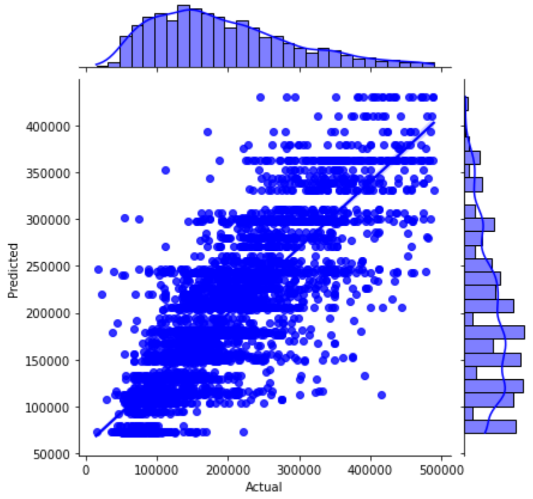

Figure 5. Scatter plot and histograms: Predicted values against actual values

As Figure 5 illustrated after tuning the parameters the predicted target shows linear association with the actual target. Also, as histograms demonstrate we got almost the same shape for predicted and actual values.

Random Forest is a type of supervised learning algorithm used to solve regression and classification problems. Therefore, this estimator was chosen to solve this regression problem. Random Forest Regressor is an estimator that generates a number of decision trees using various sub-samples of the data. Then, it averages the results obtained from the sub-samples in order to improve the predictive accuracy and control over-fitting.

First, the dataset was modeled using Random Forest Regressor with its default parameters. The obtained RMSE values were around $52500 for the capped dataset and $46300 for the uncapped dataset. After these values were obtained, hyperparameter tuning was applied using GridSearchCV. Once the best model was generated, the new values for RMSE were obtained. The new values were around $49300 for the capped dataset and $43700 for the uncapped dataset. Other results extracted from the different metric techniques of model evaluation can be found in Tables 2 and 3. It is important to notice that the errors obtained for the capped values were larger than the errors obtained for the uncapped values as it is seen in the previously mentioned tables. This was expected and discussed in the EDA section of the report. Hyperparameter tuning helped improve the errors for this model. The errors for the model are quite high considering the range of prices, but this model still was the most accurate out of the four implemented models, as it can be seen in Tables 2 and 3.

Table 2. Model vs Error Evaluation Method. Capped Values (tuned hyperparameters)

Model	RMSE	MAE	cross_val_score	score	wall time [train]	wall time [test]
LR	68949	49591	68471	0.637	17.9 ms	9.94 ms
Lasso	68949	49590	68471	0.637	48.8 ms	4.99 ms
Ridge	68949	49590	68471	0.637	18 ms	8.98 ms
Elastic Net	77862	58734	78592	0.537	25.9 ms	7.98 ms
SVR	79738	56104	N/A	0.460	14min 19s	2.07s
Decision Tree	61041	42239	61301	0.715	3 min	3.89 ms
Random Forest Regression	49310	32349	49495	0.812	1.8 s	53.3 ms

Table 3 Model vs Error Evaluation Method. Uncapped Values (tuned hyperparameters)

Model	RMSE	MAE	cross_val_score	score	wall time [train]	wall time [test]
LR	60044	43727	59668	0.614	27.9 ms	7.98 ms
Lasso	59913	43705	59650	0.616	45.9 ms	4.94 ms
Ridge	60003	43721	59668	0.614	14.9 ms	5.98 ms
Elastic Net	67663	52149	68184	0.510	25.9 ms	7.98 ms
SVR	68516	50235	70561	0.461	12min 3s	1.89s
Decision Tree	54244	38900	54812	0.685	2 min 29s	3.6 ms
Random Forest Regression	43777	29718	45104	0.795	1.24 s	36.9 ms

IV.III Feature importance

A. Linear Regression

To evaluate the features importance of the LR model, the coefficients of the model were extracted. Regression coefficients are estimates of the unknown population parameters and describe the relationship between a predictor variable and the response. The sign of each coefficient indicates the direction of the relationship between a predictor variable and the response variable. A positive sign indicates that as the predictor variable increases, the response variable also increases. A negative sign indicates that as the predictor variable increases, the response variable decreases. The coefficient value represents the mean change in the response given a one unit change in the predictor. So the larger the coefficient can be interpreted as more weight and significance for this feature. A few valuable observations on features importance can be highlighted based on Figures 6 and 7:

1) It can be seen that among all the features the income has the highest positive significance, followed by the households and bedrooms per room numbers. These are expected, as higher salaries correlate with buying more expensive houses, while larger households and larger number of bedrooms per room imply the larger properties and thus, higher prices as well.

2) Interestingly, the strong negative coefficients are noticed for the longitude, latitude and the price. By analyzing and visualizing that, it can be seen that the more north and the east the direction on the map the cheaper the houses are. By looking at the map of California from EDA (Figure 3), it can be seen that this direction represents the exact moving more towards the continental part of the state and away from the coastal line. So this finding highlights that.

3) Another strong negative relationship was observed for the population, which can be explained by the fact that the lower the population in the area the higher the price, as it can be considered a more private area.

4) Interestingly, the small positive correlation was found for the house age and price, which contradicts with the expected result, since the newer houses were predicted to be more expensive. Most probably, those houses which are older, are placed in the favorable and popular locations, and location, as we observed previously, plays an important role in defining the house price and this effect overpasses the age factor.

5) Although the location is important, it might be difficult to interpret the ocean proximity with this model as the coefficient is small. Still its positive value indicates that with closer proximity to the ocean the housing prices go up.

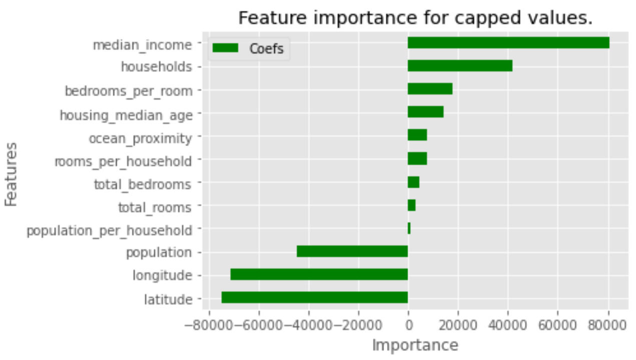

Figure 6. Linear Regression feature importances (capped dataset)

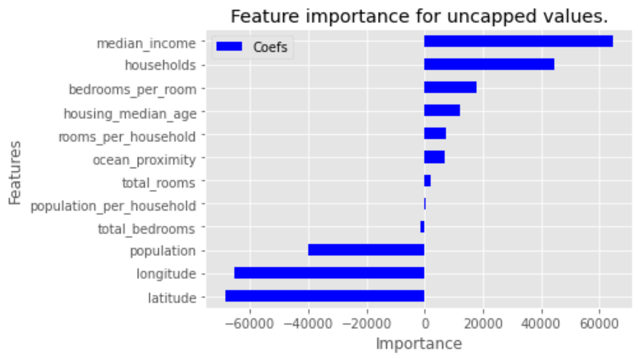

Figure 7. Linear Regression feature importances (uncapped dataset)

B. Lasso / Ridge / ElasticNet

The feature importances for Lasso and Ridge (Figure A1, Figure A2) were observed to be very similar to LR's (Figure 7), where the most important features were income, households, location (longitude and latitude) and population. For ElasticNet, the features coefficients were different (Figure 8). The largest difference was that the ocean proximity became the second strongest feature, which was initially expected. However, as this model performed worse than the baseline LR and Lasso in terms of having largest errors, it would not be recommended to use this less reliable model.

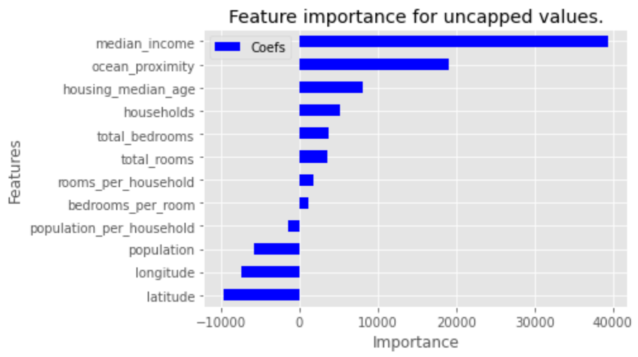

Figure 8. ElasticNet feature importances (uncapped dataset)

C. Decision Tree

In Decision Tree the feature importances were were found by the application of the attribute "feature_importances" to the model. For uncapped and capped datasets the values have shown similar behaviour, where the most important features were median_income, ocean_proximity, households, population per household (Figure 9 and 10). So these results are somewhat similar to LR, Lasso and Ridge. However, ocean proximity is the second important feature, which was predicted from EDA and can be highlighted as one of the major findings for this model.

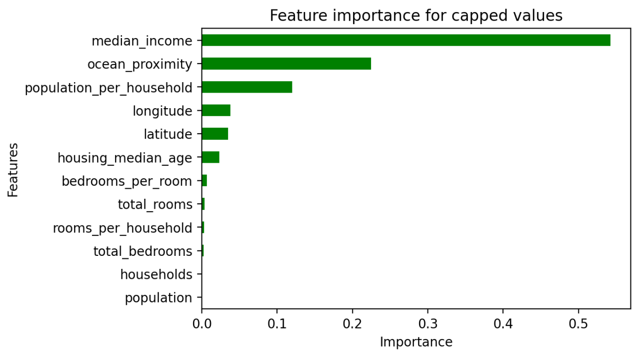

Figure 9. Decision tree feature importances (capped dataset)

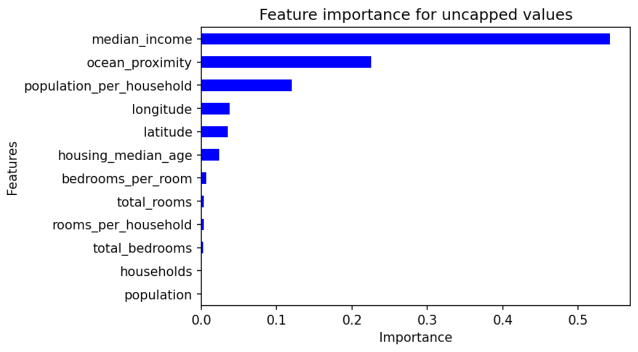

Figure 10. Decision tree feature importances (uncapped dataset)

D. Random Forest Regression

The feature importance or variable importance describes which features are relevant and it can be helpful for a better understanding of the problem. Figures 11 and 12 show the importance of each feature obtained with the best model generated with Random Forest for capped and uncapped data respectively. These figures were also obtained by using the attribute feature_importances. After the feature importances were obtained, a feature vs feature importance barplot was created for each dataset with the previously sorted values.

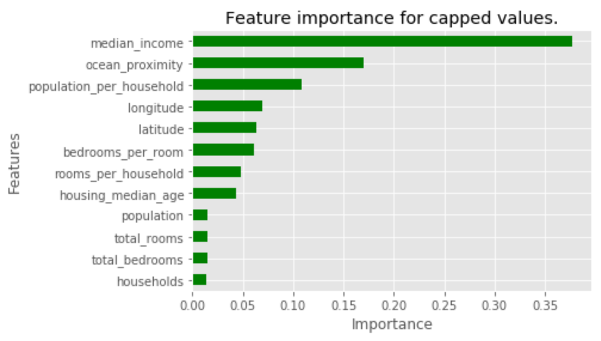

Figure 11. Random Forest feature importances (capped dataset)

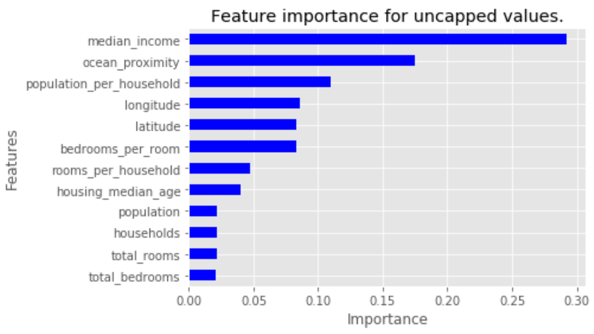

Figure 12. Random Forest feature importances (uncapped dataset)

It is possible to see that the most important feature for this model is the median income, followed by the ocean proximity and the population per household. The same thing occurs for capped and uncapped data. So the results are very similar to the Decision Tree model's one.

Besides ocean proximity/location and income, another important feature was population per household. Then, longitude, latitude and bedroom per room shared around the same importance. The next set of features sharing around the same level of importance were rooms per household and housing median age. And, finally, the last set consisted of total rooms, total bedrooms, population and households. The significance of the first few features was expected, since larger households and larger number of bedrooms per room imply the bigger and more expensive properties and thus, higher prices as well. As previously mentioned, EDA showed that houses that are closer to the ocean tend to be more expensive. People show a preference for houses located closer to the coastal line. EDA also showed that people were moving to the continental area and that this area offered cheaper houses. This explains why ocean proximity, latitude and longitude features are quite significant compared to other features. People that have the resources to buy a house near the coastal line are willing to spend more money, but those who don't, choose a cheaper alternative in a more continental area. A more precise relation between latitude, longitude and price is obtained when complementing the results from this model with the results from linear regression, since the barplots of the coefficients also add if a feature is directly or inversely related. The remaining features have quite a low importance. This could be because, as it is possible to see from EDA and other models, location is the most important feature for people, and the rest of the features end up being less important.

V. CONCLUSIONS

As most of us want to work in the industry sector once we finish our studies, this type of project equips us with valuable machine learning tools that we can use later on in our professional lives. For example, in this project we applied regression models, which is a valuable tool. Predicting prices, as mentioned in the introduction, is extremely valuable. So the main findings from the project include:

- EDA is an important part of the data science project that helps to make decisions on the data preprocessing, which defines the models performance. In our case, the observation of price distribution helped to notice the "capped" values, the removal of which improved all the models performances significantly. In addition to that, EDA helped us to observe the preliminary feature importances, like strong correlation of prices-income and prices-location.
- These correlations and feature importances were confirmed in most of the models. The higher income and the closer proximity to the ocean implied purchases of more expensive houses.
- Among all the tested models Random Forest performed the best, as it was expected.
- The best hyperparameters of Random Forest were 'max_features': 4 and 6 and 'n_estimators': 30 , generating the lowest RMSE among all the models. These values for RMSE were between $43,000 and $44,000
- The most relevant features obtained using Random Forest were median income and ocean proximity/location. The rest of the features were divided into 3 tiers that had around the same importance. The first and most relevant tier out of the three contained longitude, latitude and bedroom per room. The next tier contained rooms per household and housing median age. Finally, the last set of features was total rooms, total bedrooms, population and households.The relevance of the first few features was expected, since larger households and larger number of bedrooms per room imply a bigger house, which tends to have a higher price. EDA explained how location affects pricing and helped us understand the feature importance results of this model. People that have the resources prefer a house near the ocean and are willing to pay more for it, explaining how relevant ocean proximity is. On the other hand, those who want to pay less tend to choose continental areas, since they are cheaper. This explains why latitude and longitude are also quite important. We were able to see that location and income tend to be the deciding factors when talking about price, leaving the other features as less important.

Population was inversely proportional as predicted, but median age didn't follow what we expected, being directly proportional instead of inversely proportional. This can happen due to the fact that even though the properties are older, the location increases the value anyways.

This project helped us see which methods are most effective and what information needs to be considered to make a proper prediction. Under-performing models helped us find which parameters needed to be changed. Taking this into consideration helped us determine which combination of parameters will produce the best model. This type of model can be used for housing prices predictions in other regions. And the structure of the code can be followed and applied in other projects related to pricing.

The future work of improvement for this project will include the possible application of Principle Component Analysis (PCA) or Non-negative Matrix Factorization (NMF) to reduce the dimensionality, which might improve the models performance in both terms of error reduction and faster estimation. Moreover, other models like GaussianProcesses, Ensemble learning models can be tested and tuned more to get even better results. Finally, for the future work and easier representation of categorical features "ocean proximity", it might be helpful to create one binary attribute per each category: one attribute would equal to 1 when the category is true (and 0 otherwise).

REFERENCES

[1] Byeonghwa Park and Jae Kwon Bae. 2015. Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert Systems with Applications 42, 6 (2015), 2928–2934. https://doi.org/10.1016/j.eswa.2014.11.040

[2] Chaitali Majumder. [n. d.]. House price prediction using machine learning. Retrieved November 5, 2021 from

https://nycdatascience.com/blog/studentworks/machine-learning/house-price-prediction-using-machine-learning-2/

[3] Subham Sarkar. September 6, 2019. Predicting House prices using classical machine learning and Deep Learning Techniques. Retrieved November 5, 2021 from https://medium.com/analytics-vidhya/predicting-house-prices-using-classical-machine-learning-and-deep-learning-techniques-ad4e55945e2d4

[4] Okmyung Bin. 2004. A prediction comparison of housing sales prices by parametric versus semi-parametric regressions. Journal of Housing Economics 13, 1 (2004), 68–84. https://doi.org/10.1016/j.jhe.2004.01.001

[5] Bruno Afonso, Luckeciano Melo, Willian Oliveira, Samuel Sousa, and Lilian Berton. 2019. Housing Prices Prediction with a Deep Learning and Random Forest Ensemble. In Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional (Salvador). SBC, Porto Alegre, RS, Brasil,
389–400. https://doi.org/10.5753/eniac.2019.9300

[6] A. Géron. 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media. https://books.google.com/books?id=HHetDwAAQBAJ

[7] Kaggle. California Housing Priceshttps://www.kaggle.com/camnugent/california-housing-prices

[8] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[9] Scikit-Learn Library. Linear Regression https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[10] Datacamp tutorial. Ridge, Lasso, ElasticNet. https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net

APPENDICES

APPENDIX 1

Github link to the repository:
https://github.com/DSE511-Team-Avatar/Team-Avatar

APPENDIX 2

Table A1. Best hyperparameters for models

Model	Best hyperparameter values
Lasso	{alpha=100}
Ridge	{alpha=10}
ElasticNet	{l1_ratio=1; alpha=100}
SVR	{'C': 10, 'epsilon': 0.5, 'gamma': 1e-07, 'kernel': 'linear'}
Decision Tree	{'criterion': 'squared_error', 'max_depth': 8, 'max_leaf_nodes': 100, 'min_samples_leaf': 20, 'min_samples_split': 10}
Random Forest	{'max_features': 4, 'n_estimators': 30} {'max_features': 6, 'n_estimators': 30}

APPENDIX 3

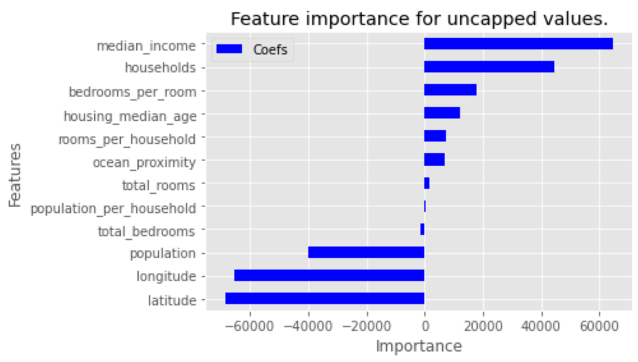

Figure A1. Lasso feature importances (uncapped dataset)

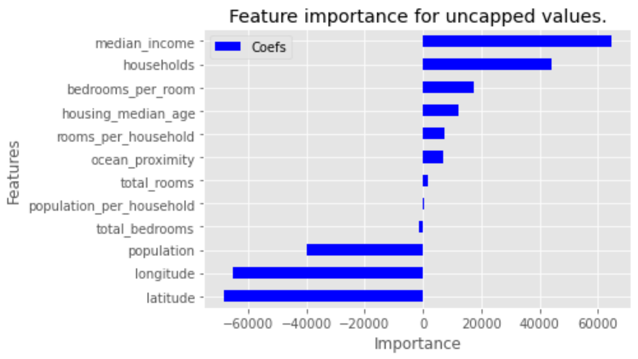

Figure A2. Ridge feature importances (uncapped dataset)

APPENDIX 4

Gaussian Processes.

Gaussian Processes (GP) are a generic supervised learning method designed to solve regression and classification, and optimization problems. GP is a useful technique that enables a non-parametric Bayesian approach to modeling [10]. The advantages of Gaussian processes include: 1) the prediction interpolates the observations (at least for regular kernels); 2) the prediction is probabilistic so that it is possible to compute empirical confidence intervals and decide based on those if refit (online fitting, adaptive fitting) is needed for the prediction in some region of interest. 3) GP is versatile: different kernels can be specified. Although common kernels are provided it is also possible to specify custom kernels. Nevertheless, there are important disadvantages of GP that should be considered: 1) GP are not sparse, i.e., they use the whole samples/features information to perform the prediction. 2) GPs lose efficiency in high dimensional spaces – namely when the number of features exceeds a few dozens.
It was decided to try a radial-basis function kernel with noise and an offset, and the hyperparameters for the kernels (ConstantKernel, ConstantKernel, RBF, WhiteKernel) were suggested values and those were optimized during fitting. The similar dataset was tested in one example and showed a good performance [11]. In our case, as the dataset was too large, the GP model could not run it and the code crashed. The warning that was received: "Unable to allocate 3.98 GiB for an array with shape (15686, 15686, 2) and data type float64." This most probably happened due to the disadvantages of GP mentioned before, where GP are not sparse and they lose efficiency in high dimensional spaces. To check this model performance on the smaller dataset, our data was cut to 800 train and 200 test sets and modeled with GP again. As it was found, this model performed worse than the baseline Linear Regression. The final results showed RMSE=94134 and MAE=76483. These large errors can be explained by the fact that not all data was used and only a small portion of it (5%), which does not allow the model to train and predict well. Thus, further investigation should be done on kernels and their parameters and more powerful hardware should be used for testing GP model.

References:

[10] Scikit-Learn Library. Gaussian Processes. https://scikit-learn.org/stable/modules/gaussian_process.html

[11] Towardsdatascience. Getting started with Gaussian Processes regression modeling. https://towardsdatascience.com/getting-started-with-gaussian-process-regression-modeling-47e7982b534d