House Price Prediction System

CS Capstone

Andrew Rusnac

Student ID: 008860864

C964

Table of Contents

Part A

Letter of Transmittal

3/22/2022

Ana Boren

2827 SE Ankeny St.

Portland, OR 97214

Dear Mrs. Boren,

I am happy to share this project proposal for the housing price predictor project we discussed previously. Your business has exciting and ambitious goals, and I have no doubt this application will help achieve these goals.

Due to the increasing cost of building materials and labor, you expressed the need for a solution that would allow your business to cut costs, maximize profits, and forecast future earnings. Your business has been growing at a rapid pace, and the current solutions you use are insufficient and time-consuming. The proposed Housing Price Predictor application will be a solution to your current problems and will lead to future growth. When provided a set of data, the application utilizes a machine learning model to predict the sale price of a home based on its features such as location, square footage, number of bathrooms, number of bedrooms, etc. Through testing, it has been shown to be 99% accurate in its predictions. In addition, the application provides a browser-based interface to allow for easy visualization of the housing data and allows the user to input new housing data to see the predicted sale price based on the features provided. This will save your company time and will also help fine-tune your planning and future developments so you can maximize profits.

The solution can be implemented within a month, and the total working hours will come out to 228 hours at a total cost of $11,400. The developer that will be working on the solution has over 5 years of professional programming experience with Python and has worked on numerous data science specific projects, where machine learning was successfully applied. We are excited to work further with you and look forward to hearing back. Do not hesitate to reach out with any questions or comments.

Sincerely,

Andrew Rusnac

Problem Summary

With building materials and the cost of labor continually on the rise, more now than ever, it has become paramount for Stumptown Development to find ways to reduce costs, maximize profits, and get an accurate forecast for their earning potential on each home they develop. With Stumptown's Development large development portfolio, it is crucial to improve the accuracy of estimated investment returns of their new construction homes.

Previously, to obtain this information, they relied primarily on expert opinion and manually monitoring prices of recently sold homes that match the features of the homes in their new developments. This has become far too time-consuming as the firm has grown, with numerous new housing developments under construction in several different cities in Oregon. A new process with greater accuracy is needed. The proposed Housing Price Predictor, using a machine learning algorithm, with an accompanying dashboard will provide the predicted sale price of any home based on the features of that home in addition to data visualization.

Benefits

The application will provide direct benefits to the company by using historical data to build a model that accurately predicts a home's sale price based on its various features. With this tool, the company can input the features of the homes that are under development and get an instantaneous selling price. This data can then be used to estimate upcoming earnings once those homes are finished and ready to be sold. The application can also be used to plan future development. It displays charts and graphs that show how each feature of a home relates to the selling price. The company can make decisions based on this information such as deciding to build in the zip codes with the highest returns or adjust homes to have the specific features that have the highest return on investment. This will allow for a more focused approach, cutting down on time-consuming activities, which will directly cut costs. The ability to plan and maximize return on investments will increase profits and allow the client to forecast future earnings accurately.

Application Description

The application uses a data set of home listings and sales in the Portland metro area. It then uses the features of an individual home such as the home's zip code, size, number of bedrooms, and bathrooms along with other features to produce a prediction on how much that home will sell for. This prediction feature will be utilized in a web browser. Along with the input and prediction, visualization of the data in graphs and charts will be presented in the web application, allowing for quick and easy analysis.

Description of the Data

The current iteration of the application will be using data that was published on Kaggle.com and collected by Jen Wadkins. The data is Portland Housing Prices taken off Zillow.com in CSV format. It features data from July 2020 to July 2021. In its raw format, it has over 25,000 entries. One of the big pros of this data set is the size of the data set, not only in the number of homes listed but also in the number of features of each home. There are numerous columns of information to choose from regarding the attributes of the house from which to build models and make predictions. However, for this proof of concept, we have only selected a handful of these features which we believe give a full and accurate picture of the power of the model. These features include zip code, home type, living area, and lot size, among others. Although the size of the set is a pro it also comes as a con due to the substantial number of columns, since some of this data is not useful for this application and must be pruned before use.

Before the data can be used, a few anomalies need to be ironed out, such as rows with missing values, which will be handled in the application. The values that are strings (Single Family Residence or Carpet for example) must then be converted to numerical values before being applied to the machine learning algorithm.

Objective and Hypotheses

The objective of this product is to take housing data as input and use the features of the homes in the data set to provide the ability to make predictions on the selling price of any other home that is input by the user. Additionally, a dashboard for the visualization of data and ease of use will be a part of the application. This will save the company money, be user-friendly, allow for better planning and increase profits.

If we apply machine learning to the task of finding a home's selling price based on its features, then we can get an accurate and instantaneous price, saving the company time and money.

Methodology

Since this project will be linear with fixed requirements, we will be using the waterfall methodology for development. The waterfall method is easy to understand and follow. For the requirements phase, requirements will be collected in collaboration with the client, Stumptown Development. They will be finalized in a single, readily accessible document. Following this, we will move into the design phase where we will design the technical solution to the problems set out in the requirements. First, with a high-level design describing the purpose and scope, then onto the physical design of hardware and software technologies. For implementation, the application will be coded based on the requirements and specifications set out. After implementation, verification will follow, where we will ensure the application is free of errors and all the requirements have been completed. During verification, we will perform acceptance testing with the relevant stakeholders to get feedback. If the stakeholders request changes or introduce new requirements we will return to the design phase to implement them. Finally, we will enter deployment and maintenance, where the application will be deployed to the client and maintenance begins. If any defects are found or change requests are made, they will be taken care of by the team.

Funding Requirements

The project's cost will be for labor, totaling 228 hours for one developer, coming to $11,400. The programming tools will be included in this cost and there will be no use of proprietary software or data. Yearly maintenance will be billed at $100 per hour based on Stumptown Development's needs and requests.

Stakeholder Impact

The solution will have the greatest impact on the stakeholders who make business decisions. It will provide information through its machine learning algorithm and display it through the dashboard. This will allow those stakeholders to make quicker and more informed decisions, which will both increase profits and reduce costs. In addition, this will save the stakeholders time as it will automate some of the work they previously did manually. This functionality will also allow for easier and simpler reporting, which will benefit the stakeholders whose function is to plan and forecast for the future and for those who make decisions based on that reporting.

Data Precautions

The data used is not sensitive or protected and will not run into any issues with the various data protection laws or regulations. For the first iteration and for the proof of concept the data used is in the public domain. As such, it can be used for any purpose, including commercial, without asking for permission.

Developer's Expertise

The developer is a university graduate with a computer science degree. They also have 5 years of professional experience specializing in Python and Machine Learning. Through those five years, they have worked on both small and large projects, from start to finish (including projects that have utilized the waterfall methodology). They have built complex machine learning applications for a variety of clients and established companies.

Part B

Problem Statement

Stumptown Development has a need for an application that will provide them with accurate housing sale price predictions to make better-informed decisions. This application will help them cut costs, increase profits, and more accurately forecast future earnings. It is important that the application saves them time by automating some of the processes they previously did manually. These problems can be solved by using machine learning and providing a web-based dashboard to allow for easy access and user-friendly interaction with the application.

Customer Summary

The application will be used internally by employees of Stumptown Development, specifically those who are responsible for decision-making and those who are responsible for putting reports together. The dashboard will be a web application with data visualization and input fields for the user. The input fields will be used to enter features of a house and get a prediction of how much that house will sell for. Since it is a web application, no specific skills or extra knowledge will be needed - if the user is familiar with using a browser. The data visualization aspect will have proper documentation and labeling to ensure it can be clearly understood.

Existing System Analysis

The existing system is a collection of excel sheets stored in a directory in the company's file storage system. The data for homes, their features, and selling price are all collected manually. Prices for homes currently under development are manually researched by comparing them to those that are similar in the surrounding area. After completion of the solution, the company will have a web application that replaces most of the manual process with an automated process, particularly in researching prices for homes currently under development and homes planned for future development. The research will no longer be needed to determine prices as that data can be obtained simply through the application.

Data

The data is a collection of housing data taken from Kaggle.com. It is a CSV file that will be imported using the Python CSV reader function and turned into a Pandas data frame. Only those columns/features that are deemed most important through analysis will be kept (number of bedrooms, bathrooms, living area, etc.) and the rest will be dropped. Following this, the rows with missing data will be dropped (if the impact is low) or the mean value will be found and inserted. Those values that are strings will be converted to numerical values to fit the machine learning model. The columns will be split into features and prices, and these will be used to train the machine learning model to make accurate predictions on a house’s selling price.

The data will need to be periodically updated as time passes due to the nature of home prices fluctuating through the years. The currently used data set will suffice for now as it contains housing data from July 2020 to July 2021.

Project Methodology

Due to the nature of the project being small, linear, and having fixed requirements, the waterfall methodology will be applied to development. This will allow for an easy-to-follow development process ensuring that the client's requirements are met, and they are satisfied with the application.

1. Requirements – The developer will communicate directly with stakeholders to define requirements. It is essential these are well understood and fleshed out, as changes can add additional challenges when using the waterfall methodology.
2. Design – The high-level scope will be defined which will include a user-friendly dashboard and house price prediction functionality. This will be followed by the physical design of the hardware and software.
3. Implementation – The application will be developed to meet the client's design requirements, the machine learning algorithm will be implemented, and the dashboard will be built.
4. Verification – Ensure that the application is free from errors using different testing methods, determine if it is user-friendly through a usability test, and verify that it meets all requirements. During this phase, we will also get feedback from the stakeholders in the form of user testing. If the stakeholders request any changes or define new requirements, these will be discussed with the appropriate stakeholders and the developer. If they are approved and accepted by both parties, we will return to the design phase and implement them. Depending on the changed requirements, the project timeline may be extended.

5. Maintenance – If any bugs or errors are found after deployment they will be fixed. Additionally, any minor changes that will not cause the scope or timeline to change will be taken care of. Maintenance will occur on an as needed basis.

Project Outcomes

The project deliverables should include any documents generated during the development process. This includes requirements documents, schedules, and any design documents such as GUI mockups or flowcharts. The product deliverables will include the house price prediction system that allows for custom inputs, a dashboard for user interaction, and data visualization built into a web application with a login function.

Implementation Plan

Following the waterfall methodology, the project will begin with communication between the developer and the stakeholders to clearly define the requirements. Once the requirements are clearly defined, the first major milestone will be completed. Next, the developer will move on to designing the application based on the high-level scope, which is also defined in this step, marking another milestone.

Following that, implementation can begin, and a wireframe will be constructed based on mockups and flowcharts. Functionality will be introduced in steps: import, read, clean the data, explore data, fit the data to a machine learning model and finally implement the dashboard.

After the data has been imported, read, and cleaned, data exploration will be conducted to define which features are most important when it comes to predicting house prices. Data visualization showing relationships between price and features will be developed. This will be used for the analysis of the machine learning model and will be part of the final deliverables. Once the key features are determined, the machine learning functionality will be implemented. It will be tested to ensure an elevated level of accuracy and a low rate of error using various metrics. To finish implementation, the dashboard will be built.

Evaluation Plan

During the verification phase, appropriate test cases will be developed, and integration testing will be done. Any errors that are discovered will be resolved and documented. The application will be compared to requirements to ensure all of them are met.

User testing will be conducted to make sure they meet the needs of the stakeholders and to get feedback. During this testing, if the stakeholders request changes or add new requirements, these will be discussed with the stakeholders and developer. If they are approved, we will move back into the design phase and implement them. The timeline may need to be adjusted for bigger changes or requirements.

Finally, the application will move on to deployment and acceptance testing to make sure that it meets the needs of the client. After deployment, any errors that are discovered can be reported and will be fixed by the developer in a timely manner.

Resources and Costs

Programming Environment

The programming environment will be Visual Studio Code on a Windows 10 desktop that the developer already owns. The application will be built in Python 3, using libraries such as SciKit Learn, Matplotlib, Seaborn, and Pandas. There are no costs for the programming tools as these will all be provided by the developer.

Environment Costs

The web application will use already existing company resources so the environment costs will be effectively $0.

Human Resources

The development of the product will take 228 hours to complete, which comes to a total cost of $11,400. Maintenace and any other requests will be billed at $100 an hour on an as-needed basis.

Timeline and Milestones

Activity	Start	End	Duration	Dependencies	Resources
1. Gather and Document Requirements	**4/1/2022**	**4/2/2022**	**16**	**None**	**Developer, Stakeholders**
2. Product Design (milestone 1)	**4/4/2022**	**4/7/2022**	**24**	**Activity 1**	**Developer**
3. Data Analysis	**4/8/2022**	**4/12/2022**	**24**	**Activity 2**	**Developer**
4. Data Visualization (milestone 2)	**4/13/2022**	**4/14/2022**	**16**	**Activity 3**	**Developer**
5. House price predicting System development	**4/15/2022**	**4/22/2022**	**48**	**Activity 2,3**	**Developer**
6. Dashboard Development (milestone 3)	**4/25/2022**	**4/26/2022**	**16**	**Activity 5**	**Developer**
7. QA testing, verification (milestone 4)	**4/27/2022**	**4/28/2022**	**16**	**Activity 1-6**	**Developer**
8. Deployment (final milestone)	**4/29/2022**	**4/29/2022**	**8**	**Activity 1-7**	**Developer**

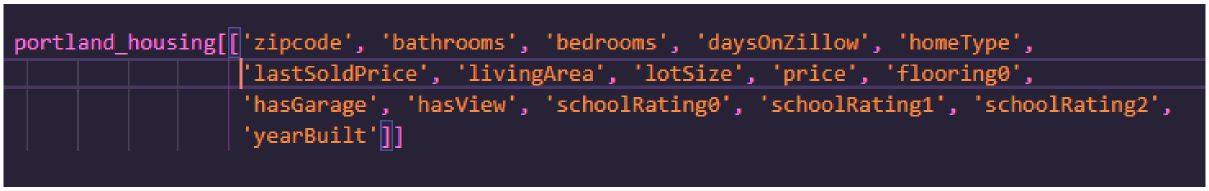
Part D

Project Purpose

The project was created to solve the business needs of a real estate development firm located in Portland, Oregon. The firm builds new homes in the Portland metro area and expressed the need for a faster and more effective way to predict how much a new home would sell for. This was needed because they utilized a manual process before which was much too time-consuming. With the rising cost of materials and the size of their business increasing they wanted a solution that would cut costs down, increase their profits, and provide a more accurate way to forecast future earnings. The product developed uses machine learning to take in features of homes and provide an accurate prediction of the home's selling price. Additionally, it has data visualization and a dashboard to make the application more user-friendly and useful, meeting all the requirements set out by the client.

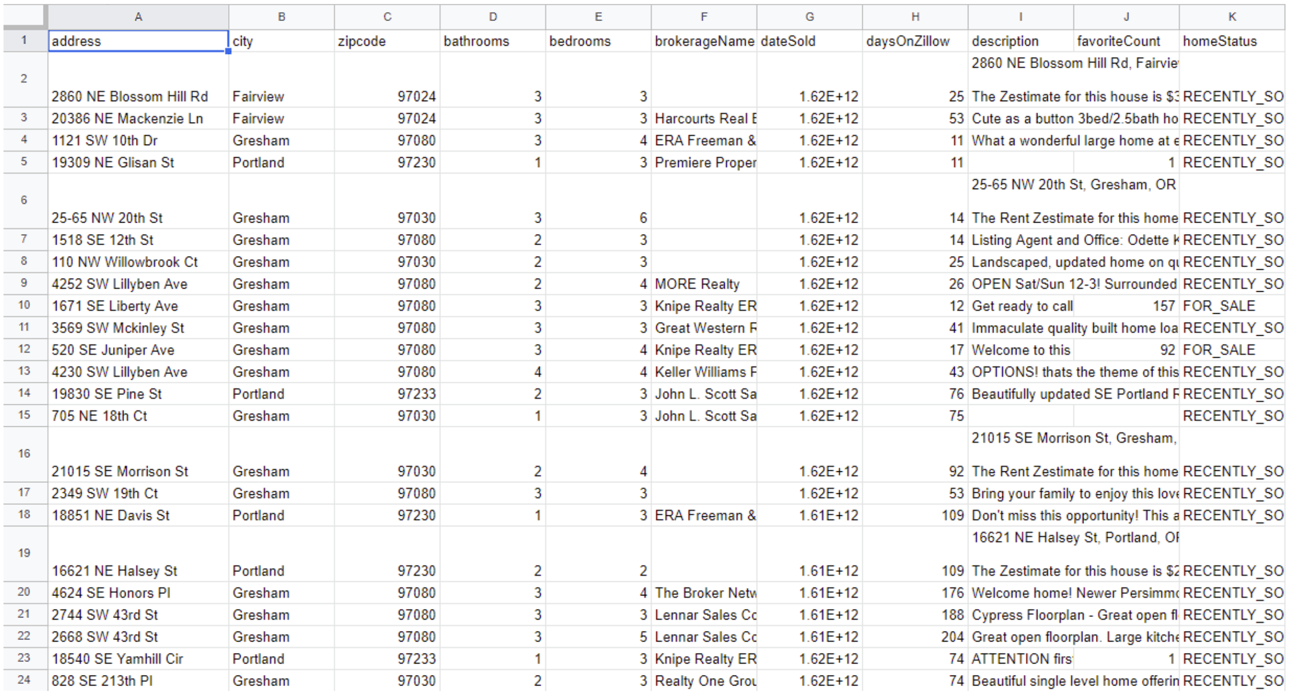
Dataset

The dataset is a collection of Portland house prices/sales from July 2020 to July 2021 scraped off Zillow. The dataset was found on Kaggle.com (https://www.kaggle.com/threnjen/portland-housing-prices-sales-jul-2020-jul-2021). The data was cleaned (rows with empty values were dropped with portland_housing.dropna()) and manipulated (strings were converted and categorized using sklearn's OneHotEncoder()) after being imported and read into a Pandas data frame in Python. Only a select number of columns, that were most important as features, were kept after analysis of data. The columns kept can be seen in the code sample below.



```
portland_housing[['zipcode', 'bathrooms', 'bedrooms', 'daysOnZillow', 'homeType',
                  'lastSoldPrice', 'livingArea', 'lotSize', 'price', 'flooring0',
                  'hasGarage', 'hasView', 'schoolRating0', 'schoolRating1', 'schoolRating2',
                  'yearBuilt']]
```

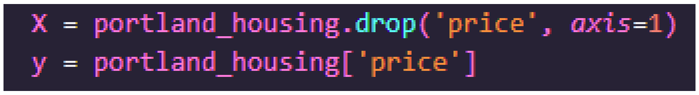
A sample of the raw data can be seen below:



	A	B	C	D	E	F	G	H	I	J	K
1	address	city	zipcode	bathrooms	bedrooms	brokerageName	dateSold	daysOnZillow	description	favoriteCount	homeStatus
2	2860 NE Blossom Hill Rd	Fairview	97024	3	3		1.62E+12	25	2860 NE Blossom Hill Rd, Fairvie The Zestimate for this house is $3		RECENTLY_SO
3	20386 NE Mackenzie Ln	Fairview	97024	3	3	Harcourts Real E	1.62E+12	53	Cute as a button 3bed/2.5bath ho		RECENTLY_SO
4	1121 SW 10th Dr	Gresham	97080	3	4	ERA Freeman &	1.62E+12	11	What a wonderful large home at e		RECENTLY_SO
5	19309 NE Glisan St	Portland	97230	1	3	Premiere Proper	1.62E+12	11		1	RECENTLY_SO
6	25-65 NW 20th St	Gresham	97030	3	6		1.62E+12	14	25-65 NW 20th St, Gresham, OR The Rent Zestimate for this home		RECENTLY_SO
7	1518 SE 12th St	Gresham	97080	2	3		1.62E+12	14	Listing Agent and Office: Odette K		RECENTLY_SO
8	110 NW Willowbrook Ct	Gresham	97030	2	3		1.62E+12	25	Landscaped, updated home on qu		RECENTLY_SO
9	4252 SW Lillyben Ave	Gresham	97080	2	4	MORE Realty	1.62E+12	26	OPEN Sat/Sun 12-3! Surrounded		RECENTLY_SO
10	1671 SE Liberty Ave	Gresham	97080	3	3	Knipe Realty ER	1.62E+12	12	Get ready to call	157	FOR_SALE
11	3569 SW Mckinley St	Gresham	97080	3	3	Great Western F	1.62E+12	41	Immaculate quality built home loa		RECENTLY_SO
12	520 SE Juniper Ave	Gresham	97080	3	4	Knipe Realty ER	1.62E+12	17	Welcome to this	92	FOR_SALE
13	4230 SW Lillyben Ave	Gresham	97080	4	4	Keller Williams F	1.62E+12	43	OPTIONS! thats the theme of this		RECENTLY_SO
14	19830 SE Pine St	Portland	97233	2	3	John L. Scott Sa	1.62E+12	76	Beautifully updated SE Portland F		RECENTLY_SO
15	705 NE 18th Ct	Gresham	97030	1	3	John L. Scott Sa	1.62E+12	75			RECENTLY_SO
16	21015 SE Morrison St	Gresham	97030	2	4		1.62E+12	92	21015 SE Morrison St, Gresham, The Rent Zestimate for this home		RECENTLY_SO
17	2349 SW 19th Ct	Gresham	97080	3	3		1.62E+12	53	Bring your family to enjoy this love		RECENTLY_SO
18	18851 NE Davis St	Portland	97230	1	3	ERA Freeman &	1.61E+12	109	Don't miss this opportunity! This a		RECENTLY_SO
19	16621 NE Halsey St	Portland	97230	2	2		1.61E+12	109	16621 NE Halsey St, Portland, OF The Zestimate for this house is $2		RECENTLY_SO
20	4624 SE Honors Pl	Gresham	97080	3	4	The Broker Netw	1.61E+12	176	Welcome home! Newer Persimmo		RECENTLY_SO
21	2744 SW 43rd St	Gresham	97080	3	3	Lennar Sales Cc	1.61E+12	188	Cypress Floorplan - Great open fl		RECENTLY_SO
22	2668 SW 43rd St	Gresham	97080	3	5	Lennar Sales Cc	1.61E+12	204	Great open floorplan. Large kitche		RECENTLY_SO
23	18540 SE Yamhill Cir	Portland	97233	1	3	Knipe Realty ER	1.62E+12	74	ATTENTION firs	1	RECENTLY_SO
24	828 SE 213th Pl	Gresham	97030	2	3	Realty One Grou	1.62E+12	74	Beautiful single level home offerin		RECENTLY_SO

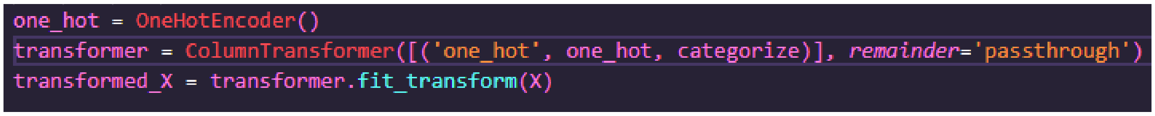
Data Product Code:

Following the data preparation, seen in the above section (dropping columns, cleaning), the data was then divided into features and a target which was the house's selling price:



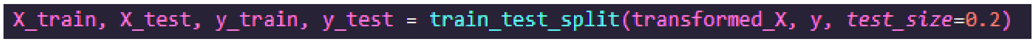
```
X = portland_housing.drop('price', axis=1)
y = portland_housing['price']
```

Next the values that contained strings, such as homeType, were converted into numerical values and categorized using sklearn's OneHotEncoder:



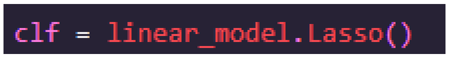
```
one_hot = OneHotEncoder()
transformer = ColumnTransformer([('one_hot', one_hot, categorize)], remainder='passthrough')
transformed_X = transformer.fit_transform(X)
```

Following this the data was then split into test data and training data:

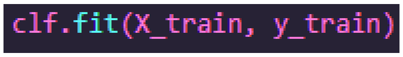


```
X_train, X_test, y_train, y_test = train_test_split(transformed_X, y, test_size=0.2)
```

The model uses Lasso Regression to make its predictions. Then the training data was fit to the model:

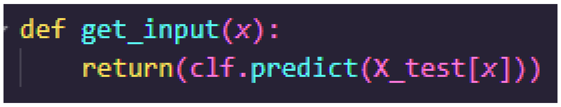


```
clf = linear_model.Lasso()
```



```
clf.fit(X_train, y_train)
```

A function is used to predict the home’s selling price based on the input of the user on the dashboard:



```
def get_input(x):
    return(clf.predict(X_test[x]))
```

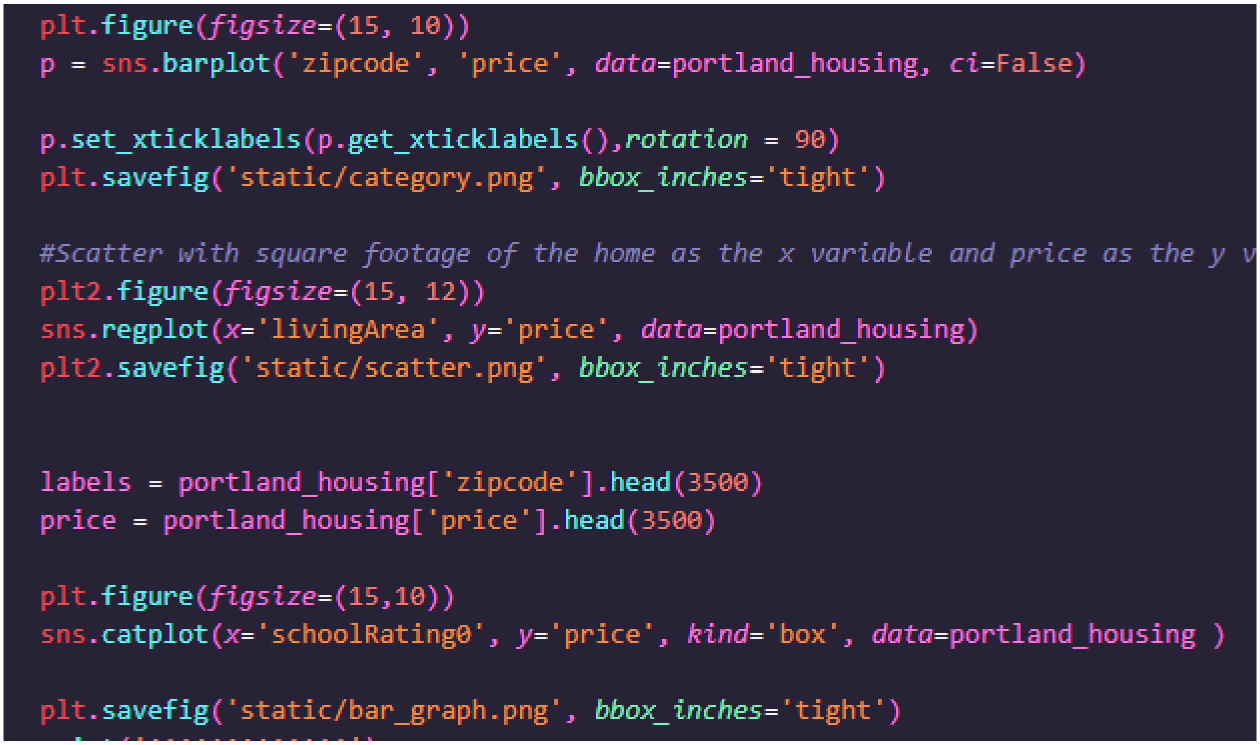
Lasso regression works well because it uses the technique of shrinkage; this is where the coefficients of determination are shrunk towards zero. This allows us to avoid overfitting the model. It can deduce which features are less important and make their coefficients zero, effectively eliminating them and giving us more accurate predictions.

Hypothesis Verification

The hypothesis that “If we apply machine learning to the task of finding a home's selling price based on its features, then we can get an accurate and instantaneous price, saving the company time and money” was accepted by the client and found to be true. However, it will take more time to fully evaluate how much time and money is saved after the application has more real-world use. The aspect regarding accurate and instantaneous price, however, can be seen from testing and use of the application as evidenced from the model score of just over 98% and manual verification that the prices predicted are close to the actual sale price.

Effective Visualizations and Reporting

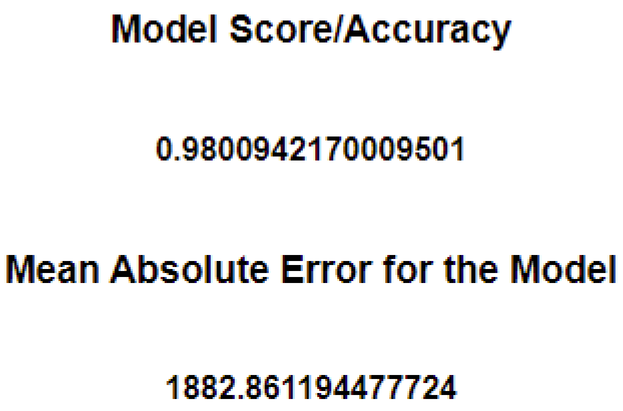
The application dashboard includes and displays three data visualizations, all of them showing the relationship of a specific variable to price. The first contains all the zip codes and the mean sale price of homes in each of the of those zip codes displayed as a bar chart. This allows the user to see how zip codes compare in relation to price – it is easy to pick out the zip codes with the highest priced homes and those with the lowest priced homes. The second visualization is a categorical box plot showing the distribution and outliers relating the home's school rating to price. It can be noted that the higher the school rating the more likely the selling price is going to be high. The final graph is a scatter chart with a regression line relating the home's selling price to the square footage of the house, the user is easily able to see how a higher square footage results in a higher selling price. These were used as part of the analysis and to determine which features would be best suited for the model and are displayed within the dashboard. The code for these can be viewed below:



```
plt.figure(figsize=(15, 10))
p = sns.barplot('zipcode', 'price', data=portland_housing, ci=False)

p.set_xticklabels(p.get_xticklabels(),rotation = 90)
plt.savefig('static/category.png', bbox_inches='tight')

#Scatter with square footage of the home as the x variable and price as the y v
plt2.figure(figsize=(15, 12))
sns.regplot(x='livingArea', y='price', data=portland_housing)
plt2.savefig('static/scatter.png', bbox_inches='tight')


labels = portland_housing['zipcode'].head(3500)
price = portland_housing['price'].head(3500)

plt.figure(figsize=(15,10))
sns.catplot(x='schoolRating0', y='price', kind='box', data=portland_housing )

plt.savefig('static/bar_graph.png', bbox_inches='tight')
```

Accuracy Analysis

In the planning phase, it was determined a high level of accuracy would be essential for this application to prove successful. By manually verifying some entries from the data set we were able to conclude that the application is indeed accurate enough to be useable. For instance, the features of the home in row 12008 were tested on the model and it predicted a home selling price of $561,080 when the actual selling price was $560,500 which is remarkably close. Using SciKit Learn's score function on the model it provided a score of just over 0.98 which is satisfactory. Additionally, using the metric of mean absolute error for the model the output was 1882, which is in line with the previously discussed example. These outputs are shown below and viewable within the application.

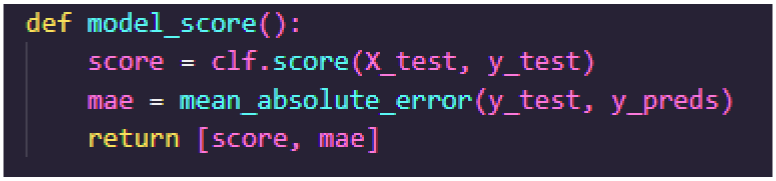


Model Score/Accuracy

0.9800942170009501

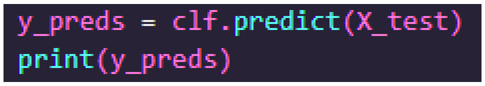
Mean Absolute Error for the Model

1882.861194477724



```
def model_score():
    score = clf.score(X_test, y_test)
    mae = mean_absolute_error(y_test, y_preds)
    return [score, mae]
```

Additionally, when we apply the predict function to the test data, we can see from the console output that the predictions are very much in line with what we would expect for a home's selling price.



```
y_preds = clf.predict(X_test)
print(y_preds)
```

```
[479738.63558412 554980.0388263  442325.79102994 ... 478596.68923446
 651767.22315767 550205.58087132]
```

Application Testing

Integration testing was used throughout the development of the application. As modules were added, each was tested before and after implementation to ensure that they worked with all other modules. The outputs were verified through the development process. Usability tests were conducted when the application was close to its finished state to ensure that the dashboard was easy to understand and user-friendly. During the development process, a handful of test cases were used to verify that the application performed as intended and the inputs matched the desired outputs. Additionally, a function to log certain errors into a text file for maintenance and health checks is built into the application.

The text log, along with stakeholder feedback, will be used for maintenance every month. Any bugs or issues will be fixed and feedback from the stakeholder will be considered. If the stakeholder's feedback translates into new requirements those can be discussed with the developer and implemented as a new project. Minor changes or updates can be implemented at the rate of $100/hr.

After six months, the developer will meet with the stakeholders to discuss the application's performance. If it does not meet the standards that were set out in this document, steps will be taken to correct the issues.

Application Files

The application files are structured with folders in bold and files with an * (also included is a screenshot of the workspace):

Flask

Static

CSS

*main.css

Templates

    *base.html (template html file)

    *data.html

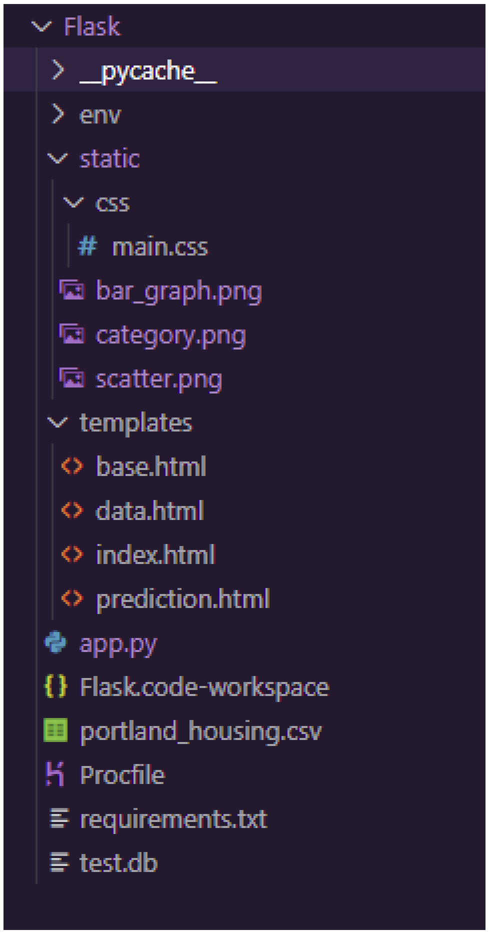
    *index.html

    *prediction.html

*app.py

*log.txt

*requirements.txt (used to pass the requirements to Heroku)



```
Flask
  __pycache__
  env
  static
    css
      main.css
    bar_graph.png
    category.png
    scatter.png
  templates
    base.html
    data.html
    index.html
    prediction.html
  app.py
  Flask.code-workspace
  portland_housing.csv
  Procfile
  requirements.txt
  test.db
```

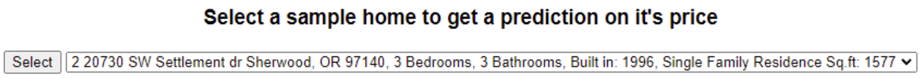
All the files rely on one another for the application to properly function and display correctly.

User's Guide

The project is hosted on Heroku and can be accessed following these steps:

1. Using Chrome, navigate to: https://capstonehouse.herokuapp.com/

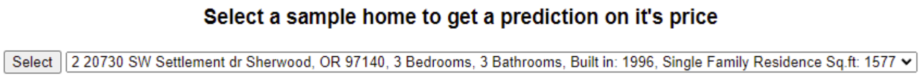
2. It may take some time to load but you will be directed to a login page, enter **admin** as the username and **c964** as the password. All lower-case, then click login.
3. You will be directed to the dashboard which contains the three different visualizations and the query function.
4. To utilize the query function, click the dropdown box:


Select a sample home to get a prediction on it's price

2 20730 SW Settlement dr Sherwood, OR 97140, 3 Bedrooms, 3 Bathrooms, Built in: 1996, Single Family Residence Sq.ft: 1577

5. Make your selection then press select. You will be directed to the prediction page which displays the predicted price based on the selection. It also provides information on the model's score and a metric used to determine the effectiveness of the model.
6. Press the back button to return to the main page. You can now make another selection if desired.

To Run the Project in an IDE:

1. Open Visual Studio code or download and install from: https://code.visualstudio.com/
2. Extract the Project Source Code folder.
3. In VS Code select File, Open Folder then select the Project Source Code folder.
4. Open the app.py file in the IDE and select Run, Run Without Debugging.
5. Open Chrome then input http://127.0.0.1:5000/ into the browser address bar.
6. Enter **admin** as the username and **c964** as the password. All lower-case, then click login.
7. You will be directed to the dashboard which contains the three different visualizations and the query function.
8. To utilize the query function, click the dropdown box:
9. 

Select a sample home to get a prediction on it's price

2 20730 SW Settlement dr Sherwood, OR 97140, 3 Bedrooms, 3 Bathrooms, Built in: 1996, Single Family Residence Sq.ft: 1577

10. Make your selection then press select. You will be directed to the prediction page which displays the predicted price based on the selection. It also provides information on the model's score and a metric used to determine the effectiveness of the model.
11. Press the back button to return to the main page. You can now make another selection if desired.

Summation of Learning Experience

My experience from previous programming classes proved especially useful here. After coding in Python for a previous project and knowing about Python's substantial number of data science libraries, it was an easy choice for this project. Having been interested in web development, I was led to Flask which allowed me to gain a deeper understanding of how Python can be used as the backend language for a web application, which is something I was personally interested in and inspires me to delve even deeper. Overall, this project has inspired me to continue pursuing a deeper understanding of programming and a desire to explore all the options available.