September 26, 2025

C964: Computer Science Capstone

Warning: Though it is not stated in the official resources, evaluators do not like outlines. Write narratively using paragraphs with complete sentences. Use these C964 examples to see what evaluators typically expect.

Task 2 parts A, B, C and D

Part A: Letter of Transmittal

Letter of Transmittal Requirements

The *Letter of Transmittal* should convince senior leadership to approve your project. Write a brief cover letter (suggested length 1-2 pages) describing the problem, how the application (part C) applies to the problem, the practical benefits to the organization, and a brief implementation plan. Include all artifacts typical of a professional (business) letter, e.g., subject line, date, greeting, signature, etc.

The letter should be concise and target a non-technical audience. Include the following:

□ A summary of the problem.
□ A proposed solution centering around your application.
□ How the proposed solution benefits the organization.
□ A summary of the costs, timeline, data, and any ethical concerns (if relevant).
□ Your relevant expertise.

September 30, 2025

George Jefferson

Moving On Up Real Estate
1099 Stewart Street
Suite 600
Seattle, Washington 98101

Dear Mr. Jefferson,

It has come to my attention that your company, Moving On Up Real Estate, intends to expand its services provided to include direct seller purchases, or iBuying. While there are a number of challenges to address, this proposal comes to you with the intention of resolving one of the main challenges encountered by iBuyers, valuation.

Over the years I'm sure you've used a number of the traditional valuation processes. As you know, they are not conducive to iBuying due to their sluggish pace, labor intensity, price and, often, accuracy.

Full appraisals remain the gold standard of valuation in the real estate realm. However, they suffer from all the aforementioned issues. Appraisals were developed as part of a lengthy process, often 30-45 days for closing, under which their 5-10 business day turnaround times are inconsequential. Part of the appeal of iBuyers to sellers is the short turnaround time and quick response in lieu of the lengthier traditional process.

Additionally, in recent years, it has been discovered that they often include biases of the individual appraiser issuing the report. This could affect iBuyers disproportionately as properties being considered may require improvement or renovation before they are ready to market. Appraisers may not be able to overlook the cost or appearance of improvements needed to arrive at a fair final sales price.

Broker Price Opinions (BPO's) have a similar format as an appraisal, cost less, and often are returned much more quickly than a traditional appraisal. Their lower price is indicative of two key issues: less experienced evaluators and incomplete access to the property. Broker Price Opinions are often completed by real estate agents seeking additional income opportunities. They often are not sufficiently trained and certainly have not endured the same apprenticeship experience as an appraiser. These valuations are often completed on a drive-by basis, with the evaluator only examining the exterior of the home from the street.

Desktop appraisals attempt to reduce the turnaround time and reduce cost as well. The main drawback is that the data considered is often not of similar quality across properties. Those completing these valuations are limited by the accuracy and consistency of those putting data into the systems, often Realtors inputting data into the MLS. As a result, some listings have more complete and accurate listing data than others. The variation in number and quality of photos is even greater, making a fair and balanced comparison among several properties a challenge.

Machine Learning provides a superior solution to the valuation problems identified. Most importantly, a machine learning algorithm can ingest and process datasets that are much larger than the traditional 3-6 comparable properties used by appraisers and brokers to generate their reports. While it is true that machine learning can often reiterate biases that were apparent in the training data, in this instance, the training dataset was generated by as neutral a third party as possible. The property assessor for King County is not involved in the sale, improvement or resale process. They are also not tied to the MLS system. Property assessors have their own systems, quality control and measurements definitions that provide a level of consistency that does not exist in the MLS. Additionally, the broad dataset utilized ensures that property values are not based on the current condition of a home, but rather the potential final, improved value.

After training, the machine learning model is applied and accessed via a page on your company's intranet. It will be secure behind the measures you already have in place to protect your intranet site, safe from the prying eyes of your competitors. As the product is refined over time, additional sales information can be added to improve accuracy. Your employees will be able to input some basic property information and receive a final sales price prediction with the click of a button.

With over 20 years' experience as a real estate broker combined with my recent computer science degree, I will be able to develop and assess test models to identify the best candidates for inclusion in the final product. I'll also be able to provide market insight when evaluating the product to determine whether an application beyond a proof of concept is viable.

My team consists of myself, a data scientist to ensure we are gleaning all possible benefits from the dataset, and a machine learning engineer to tune the machine learning model and establish a data pipeline that can be used for periodic updates. Initial estimates indicate a timeline of 4 months and an initial outlay of $75,000. A dataset provided by the King County Property Assessor's office will be used to test and develop the machine learning model.

Enclosed are the project details and timeline as well as expected expenditures. I appreciate the opportunity to present this product to you, and look forward to demonstrating its applications for your new venture.

Sincerely,

John Ames

Part B: Project Proposal Plan

The project proposal should target your client's middle management. This audience may be IT professionals but have limited computer science expertise. Use appropriate industry jargon and sufficient technical details to describe the proposed project and its application. Remember, you're establishing the technical context for your project and how it will be implemented for the client. **Write everything in the future tense.**

Project Summary

□ Describe the problem.
□ Summarize the client and their needs as related to the problem.
□ Provide descriptions of all deliverables. For example, the finished application and a user guide.
□ Provide a summary justifying how the application will benefit the client.

Problem Description

Direct-buying, or iBuying as it is popularly known, is an alternative path for homeowners who want to sell their home. The traditional model involves several time consuming and costly steps for sellers. Not all homeowners are willing to go through the process. Common steps in the traditional model involve home inspections, repairs, staging, lengthy exposure to the market, showings, additional inspections, negotiations and even contract failures that begin the process all over again.

For any number of reasons, some sellers are not able or willing to go through the traditional process to sell their home. In the past, their only alternative would have been local real estate investors offering 30% below market value in exchange for a simpler process. This is the opportunity seized by direct-buyer companies. They can offer sellers closer to market value by taking advantage of modern technology, economies of scale and in-house handling of some traditionally 3rd party tasks.

Many times, sellers seeking an alternative home sale method are also seeking a shorter contract-to-close timeframe. In all instances, consumers contacting a company have come to expect a quick response, regardless of how complex the process might be behind the scenes. For Moving On Up Real Estate, one of the challenges of moving into the direct-buying market will be the accuracy and timeliness of the valuation process.

Traditionally, appraisers and Realtors might be consulted for a selection of comparable properties in comparable condition with value adjustments for features that do not align. Often, the focus is on a small subset of features such as living square footage, bedrooms, and bathrooms, and very little is taken into account beyond the physical features of the properties themselves. A machine learning algorithm can digest a significantly larger set of datapoints and find both linear and non-linear relationships that might be much more complex than those used in the traditional methods.

By developing and implementing a valuation tool based on trained machine learning algorithm, Moving On Up Real Estate will be able to take advantage of the speed and accuracy of the tool to grow their business. Faster turnaround times will translate into more customer conversions. Greater accuracy will grow the bottom line.

Client and Needs Summary

Moving On Up Real Estate is a real estate broker in Kings County, WA. Moving On Up Real Estate would like to increase their share of the proverbial pie in their market. To accomplish their goal, they would like to break into the direct purchase or "instant buying" (iBuyer) market. To begin their analysis of a property they would like to rely on a Machine Learning model to predict the sales price of a potential investment.

Over the past several years, there has been a reduction in competition in this market. Several large competitors including Zillow and Redfin exited the direct-buyer market. Less competition and recent indications from the Federal Reserve regarding future declines in interest rates make the owners at Moving On Up Real Estate feel that the time is right for them to take on a new segment of the real estate market. They will still be competing with more established competitors in the market. For their company, internal access to a quick and accurate valuation tool might be the difference maker in their market.

Deliverables Descriptions

My team will be responsible to discover and develop a cleaned, structured dataset from Moving On Up Real Estate's primary market based on historical property sales, market variables, and geographic data. We will select, train and validate a Machine Learning model for sales price prediction. The Machine Learning model will be integrated into the company intranet for use Moving On Up Real Estate's employees. They will be able to utilize their current identity and access management tools to ensure that only authorized users have access to the tool. An ongoing refinement plan will be developed to incorporate updated market data, monitor model performance, and update or retrain when accuracy falls below 5% of MAPE. Basic training and user guides will be added to their company wiki. User guides in the form of short videos demonstrating basic tasks will be developed.

Benefits to Client

To address the organizational goal of increasing market share and entering the iBuyer space, a Machine Learning-based property price prediction model will be developed to support rapid and accurate valuation of potential investments. The solution involves training a predictive model that evaluates past real estate transaction data along with property features and local market economic indicators to predict the final sales price of homes quickly.

Incorporating Machine Learning into the sales price prediction process would improve pricing accuracy since Machine Learning models can analyze complex non-linear relationships in large datasets. The addition of Machine Learning to the process should also reduce offer turnaround time enabling Moving On Up Real Estate to beat competitors to the table. Accurate sales price forecasting should also result in increased profitability by helping to price assets accurately from the outset.

Data Summary

□ Provide the source of the raw data, how the data will be collected, or how it will be simulated.

□ Describe how data will be processed and managed throughout the application development life cycle: design, development, maintenance, etc.

□ Justify why the data meets the needs of the project. If relevant, describe how data anomalies, e.g., outliers, incomplete data, etc., will be handled.

□ Address any ethical or legal concerns regarding the data. If there are no concerns, explain why.

Data Source

The dataset utilized will be downloaded from www.kaggle.com. It is named "House Sales in King County, USA". The original dataset can be downloaded directly from the following link: https://www.kaggle.com/datasets/harlfoxem/housesalesprediction/data

Data Processing

After initial inspection, the data will be processed to identify and resolve any deficiencies. After deficiencies and omissions are resolved, the data will be transformed and scaled appropriately for consumption by a machine learning algorithm. Once a model has been trained, a data pipeline will be assembled to ensure that as new data is added in the future it undergoes the same resolution, transformation and scaling processes as the original data. As new data is collected and passed through the pipeline, it will be utilized to refine and retrain the model as necessary.

There are 21 columns in the dataset. The following column definitions were found for the dataset:
id - Unique ID for each home
date - Date of the home sale
price - Price of each home sold
bedrooms - Number of bedrooms
bathrooms - Number of bathrooms, where .5 accounts for a room with a toilet but no shower
sqft_living - Square footage of the interior living space of the home
sqft_lot - Square footage of the lot of the home
floors - Number of floors
waterfront – An indication whether the home was overlooking the waterfront or not
view - An index from 0 to 4 of how good the view of the property was
condition - An index from 1 to 5 on the condition of the home
grade - An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design
sqft_above - The square footage of the interior housing space that is above ground level
sqft_basement - The square footage of the interior housing space that is below ground level
yr_built - The year the house was initially built
yr_renovated - The year of the last renovation of the house
zipcode – Zipcode where the house is located
lat - Latitude
long - Longitude
sqft_living15 - The average square footage of interior housing living space for the nearest 15 neighbors
sqft_lot15 - The average square footage of the land lots of the nearest 15 neighbors

A number of Python libraries will be used within a Jupyter notebook to analyze, visualize and transform the data. These libraries include: pandas, numpy, seaborn, scikit-learn, and matplotlib.

We expect to encounter a number of issues that need to be resolved in the preparation process. Missing data and null entries are common among large datasets. An effort will be made to preserve property data by searching for properties with multiple sales in case missing data is contained in one instance of the property. If the missing data cannot be located within the dataset, and the number of affected properties is low, the affected properties will be omitted.

Data input errors are also likely. Transposition and duplication of digits are commonly encountered. If the error is obvious, a correction will be made. Otherwise, the data will need to be eliminated.

Categorical data such as zip codes, grade, condition, view and waterfront will need to be properly encoded using one hot encoding to be most useful for a machine learning algorithm.

In this case, the date will likely not be useful as the collection lasts for one year. Market trends are not likely to bear themselves out over the course of a single year. If the dataset contained several years' information, we might be able to extract seasonal or longer term trends. For the initial training, date will not be included in the training data. However, it will not be dropped from the dataset during the data processing and feature engineering operations in the event it becomes useful after the project has been operating for several years. Instead, the date will be included in a list assigned to a "drop_columns" variable utilized just prior to dividing the dataset into test and train splits. So, it the transaction date becomes significant in the workflow in the future, it will be simple enough to eliminate from the drop_columns list.

Outliers will also be encountered. In a market area such as Seattle it is expected that some property values will dwarf many or most of the other property values in the dataset. If the gap between several of the highest priced properties is too great, the outliers will need to be eliminated. It may be the same case at the low end of the market as well.

Grade is another likely candidate to contain outliers. With an established average level of 7 in the definition, it may be simply human nature for the grades of homes to aggregate toward the middle value.

iBuyers will likely want to operate safely in the middle of a market and avoid the extreme fringes, both high and low. iBuyers business model is not the same as home flippers, who often take on properties with exceptional repairs or renovations required. iBuyers are attempting to capitalize on a convenience factor for sellers and use volume to increase their profit. With confirmation from the principals at Moving On Up Real Estate, it would likely be satisfactory to focus on accuracy in price ranges and grades with the most turnover and ignore or eliminate the extreme minimum and maximums as they are likely not useful in their business model.

Periodic update of the dataset is expected. Property assessor offices typically update their property assessments at least once annually. While the valuations of the model should be checked on a regular basis for drift, a batch update will be made once annually when the updated information is available from the property assessor's office.

Data Justification

This dataset was selected for several reasons. First, it covers a number of basic features pertaining to residential real estate specifically such as living square footage, bedrooms, bathrooms, and sale price. Additionally, the dataset includes some assessments of each property's quality and condition via the condition and grade categories. Some additional features are also captured that are relevant to a property's overall appeal such as view, waterfront, floors, basement square footage, and even the year of the most recent renovation. Lastly, there is some location information included, via zip code, latitude, and longitude, which is often a component of value estimation in residential real estate. Based on the design requirements, this data should provide a sufficient basis on which to train a machine learning model.

Legal and Ethical Concerns

On initial inspection of the dataset, the "id" column appears to be a candidate for removal as "personally identifiable information". However, after additional consideration "id" does not seem to translate into a specific house outside the dataset. It does not correlate with property tax id or any similar identification that would allow individual properties to be singled out based on the dataset.

In this case, "Id" is helpful in that it can be used to search for home that were sold multiple times during this year, and eliminate all but the most recent. "Id" is also helpful to search for and confirm inclusion or removal of particular records as the data is cleaned and processed. Finally, if the dataset is expanded and the model improved and re-trained in the future, the "id" column might be useful in combination with "date" to track property history, analyze and monitor the model, and for effective data pipeline management.

"Id" is not useful for model training, though. The column will be included in the list of drop columns that are removed just prior to splitting the data into training and testing sets. Other than "id" there were no other legal or ethical concerns about the dataset.

Implementation

□ Describe an industry-standard methodology to be used.

□ An outline of the project's implementation plan. The focus can be the project's development or the implementation of the machine learning solution.

Development will follow the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology. CRISP-DM was developed specifically for conducting data mining projects in a common, standardized way. The goal is to utilize a clear format to organize complex tasks, avoid common mistakes, and make the process repeatable. CRISP-DM also supports iterative workflows so that objectives and strategies can be adjusted, if necessary, as new data is introduced.

Business Understanding

This stage will consist of meetings and conversations with the principals and employees at Moving On Up Real Estate to ensure the final product meets their needs and functions as intended. These conversations will help to identify potential pain points in their process that might be alleviated by the machine learning valuation tool. It is important to understand their business model as well. If they intend to focus their business on a certain price segment of the market, that can be used to refine the data processing and preparation stage. We will also want to understand what data they are collecting from sellers that is available for input into the widget. Grade and Condition might be useful data points for training a model, but are not likely to be consistent or match with the assignments from the assessor's employees.

During this stage we should also assess our resources, project requirements and conduct a cost-benefit analysis. We will determine the software, hardware, and time required to complete the project. We should also determine how a successful outcome is defined.

At the conclusion of this step, we should have an agreed upon list of business objectives and our team should have developed a project plan.

Data Understanding

During the Data Understanding phase our team's first responsibility is to collect the dataset to be utilized in the project. Once the dataset has been collected, we will perform the basic analysis needed to identify basic properties such as data formats, number of records and structure. More detailed analysis should also include visualizations and attempts to identify relationships and category definitions. Issues with the dataset such as missing data, outliers, data input mistakes, and other quality issues should also be identified.

When this step is complete we should have a complete understanding of the information we will be using as well as how it needs to be transformed, imputed, feature engineered, and otherwise processed prior to training an algorithm.

Data Preparation

The majority of the work in a machine learning project is completed in the data preparation phase. In this step we take everything we learned in the data understanding stage and use it to prepare the dataset for model training. The correlated columns are kept, with non-correlated columns discarded prior to training. Data cleaning is performed which involves a number of steps that can be summarized by correction, imputation, or removal of missing and erroneous values. Data formats are converted as necessary to be useful for training. In the future, the date attribute could be converted to a datetime object if seasonality or other long term trends are deemed important during improvement cycles.

Feature engineering, which can be considered data construction, is also performed by either combining features or deriving useful features from less useful data within the dataset. For this dataset an example would be converting the "year built" for a house into "house age" by subtracting the year built from the current year. Another common valuation method used in real estate is price per square foot. In this dataset we could divide the sale price by the living square footage to arrive at price per square foot and add that attribute into our training dataset. Another common attribute that is calculated is distance to local amenities. This might be calculated using mapping features with the included latitude and longitude of each property.

For some models, another important step in data preparation is data transformation. During this process, data is often undergoes several processes to convert the scale of some attributes so that all are similarly proportioned. The most often used processes are scaling or normalization of numerical data and log transformation of numerical data. Categorical data often requires some king of encoding during this step as well. Categorical data such as zip codes, grade, and condition would be processed via one-hot encoding, label encoding or ordinal encoding.

Once data preparation is complete, it should be ready to divide into training and test sets before ultimately being used as intended to train machine learning models.

Modeling

During modeling, the data will initially be split into features and labels, in this case the label will be the sales price and the features will be the remaining data that is not dropped prior to splitting. Since the outcome is known in the form of the sales price, this will be supervised training. For a first version, a single linear regression model will be tested to ensure the process is working and establish a benchmark. Once the test version is complete, additional models will be run in tandem for comparison. After evaluation the number of models trained will be reduced in order to tune hyperparameters via grid sampling and random sampling.

At the conclusion of the modeling phase several machine learning algorithms will have been trained on the same training dataset. Initial assessment of the larger group will refine the algorithm list to one or two models producing the most promising results for final tuning.

Evaluation

Evaluation is where the machine learning rubber meets the proverbial road. At this point we should be able to determine whether this dataset and the selected learning models produce a useable product in the form of predicting sales values. Did any of the models meet the business selection criteria?

Evaluation also includes a look back and look around to examine the process. Were all of the steps followed correctly? Did all planned processes for data preparation take place successfully? Did we miss any opportunities to enhance or improve the condition of the data prior to model training? Looking back, have any errors become apparent that were missed during the planning phase? Based on the outcomes of the performance metrics and the answers to these questions, we will be able to plan next steps, either move forward refining the existing product or find another approach to test that might yield better results.

Performance of the models will be measured using the following metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R-squared), and Mean Absolute Percentage Error (MAPE). Mean Squared Error and Root Mean Squared Error both square the differences between predicted and actual values for results that are always positive. RMSE then takes the square root to return the scale to the same as the training scale, where MSE results are squared compared to the training units. The Mean Absolute Error measures the average absolute differences between predicted and test values without squaring so all errors are equally weighted. The Coefficient of Determination, or R squared, shows the proportion of the variance in the target variable accounted for in the model. This metric ranges from 0 to 1, with higher values indicating a better model fit. The Mean Absolute Percentage Error will give us the absolute error as a percentage of the actual values, a commonly understood measurement. For this project, we will be targeting a Mean Absolute Percentage Error of 5% or less.

Deployment

The team will integrate the Machine Learning model with the most consistent performance via company intranet for internal use. The initial version of the widget will allow staff to input features selected during the Business Understanding phase into text boxes and generate a sales price by clicking a Predict button. The User Guide to be incorporated into the company wiki and accompanying videos will be developed. In house white box testing and user acceptance testing will ensure the widget operates as intended before release to company staff. Staff feedback will be utilized to incorporate expanded features or track bugs that slipped throught the development and testing process. Annual batch updates will be performed as the new datasets are available from the assessor's office. Quarterly monitoring will test for performance degradation or data drift.

Timeline

□ Provide a projected timeline. Include each milestone and deliverable, its dependencies, resources, start and end dates, and duration. (a table is not required but encouraged).

□ Dates should be in the future. Write ‘NA’ where an item is not applicable.

Milestone or deliverable	**Project Dependencies**	**Resources**	**Start Date End Date**	**Duration**
Meet with principals, outline expectations, agree on desired outcomes	N/A	Project Manager Principals	10/06/2025 10/07/2025	2 days
Project and outcome approval, Gather design team	Initial meeting Outcome Agreement	Project Manager Design Team	10/08/2025 10/09/2025	2 days
Workspace, Hardware, Software setup	Project Approval	Project Manager Hardware, Software	10/10/2025 10/14/2025	3 days
Data Access, Exploration, Analysis	Setup Complete	Data Scientist	10/15/2025 10/17/2025	3 days
Data Prep, Cleaning, Transforming, Feature Engineering	Data Analysis	Data Scientist ML Engineer	10/20/2025 10/31/2025	10 days
Widget mockup presented / approved	Expectations Agmt	Project Manager Principals	11/03/2025 11/05/2025	3 days
Initial model training, evaluation, selection	Data Prep	Data Scientist ML Engineer	11/06/2025 11/12/2025	4 days
Process Evaluation, Refine process, pipeline	Initial Model Training	Data Scientist ML Engineer	11/13/2025 11/19/2025	5 days
Model Refinement, Hyperparameter tuning	Process Evaluation	Data Scientist ML Engineer	11/20/2025 11/24/2025	3 days
Widget Programming	Widget Mockup Approval	Project Manager	11/25/2025 11/26/2025	2 days
In house white box testing	Widget Programming	ML Engineer Project Manager	12/01/2025 12/02/2025	2 days
In house browser testing	Widget Programming White Box Testing	ML Engineer Project Manager	12/03/2025 12/04/2025	2 days
Product Delivery and User Acceptance Testing	Design and In House Testing Complete	Project Manager Principals Staff	12/05/2025 12/09/2025	3 days
Final Deployment and Training	User Acceptance Testing Approval	Project Manager Staff	12/10/2025 12/12/2025	3 days
Ongoing Monitoring and Maintenance	Final Deployment	Data Scientist Project Manager	Quarterly beyond deployment	3 days
Annual Batch Training	Final Deployment	Data Scientist Project Manager	Annually beyond deployment	3 days

Evaluation Plan

□ Describe the verification method(s) to be used at each stage of development.

□ Describe the validation method to be used upon completion of the project.

At the conclusion of the business understanding phase, our team should be able to successfully communicate the business model, target audience and expected data collection of the principals at Moving On Up Real Estate. We should also be able to envision the average user of the final product and how its use will impact their job. The principals should understand the process they are embarking on as well as be able to identify their target for measuring success.

At the conclusion of data acquisition and exploration, our team should understand the dataset that will be used for model training. By the end of this phase, we should know what the dataset contains, what items are missing and can be imputed, what data is duplicated and/or should be eliminated, as well as have a good idea what useful features can be developed from the supplied dataset.

After data preparation, we should have a thorough understanding of the included data, the feature engineering to take place and the correlation between features. At this point all code should be verified and peer-reviewed, and version control should be in use to track changes over time. The experiment steps to this point, configurations, and parameters should be documented. To the extent possible, data pipelines should be automated.

Once modeling is complete, the highest performing model should be selected by utilizing the performance metrics. Unit testing will be in place to ensure all code performs as expected. Grid search and random search will have identified the hyperparameters that produce the best outcomes. Parameters will have been inspected to ensure they are the correct type, are defined correctly, and are not duplicated.

Process evaluation will conclude after automated pipeline tests are in place, along with integration tests.

Widget programming and testing will be validated via functional testing, peer code review and complete browser compatibility testing.

The final validation step will be user acceptance testing. At this point, the machine learning model will have been thoroughly tested and performance rated. The widget design and compatibility will have been tested for full functionality. User acceptance testing will be performed by Moving On Up Real Estate principals and staff once it has been fully integrated into their company intranet. The final product will be tested by the actual users to ensure it lives up to the expectations of the principals, and just as importantly, functions as intended within the workflow of the end users while also producing usable output. At the conclusion of user acceptance testing, a final signoff by the principals will conclude the process.

Costs

Include the itemized costs of the project. Include specific item names where applicable, e.g., ‘PyCharm Professional Ed. 2024.3.5.

□ Itemize hardware and software costs.
□ Itemize estimated labor time and costs.
□ Itemize estimated environment costs of the application, e.g., deployment, hosting, maintenance, etc.

Resource	Description	Estimated Cost
Staff – Project Manager	Responsible for client Communications, Approvals, Installation, Training, & Minimal programming	10 weeks $50/hr x 400 hours $20,000
Staff – Data Scientist	Responsible for entire data Pipeline from initial assessment Thru model training and prediction	6 weeks $65/hr x 240 hours $15,600
Staff – ML Engineer	Responsible for model training, Selection, tuning, & evaluation	7 weeks $55/hr x 280 hours $15,400
Workspace	Co-working office with conference room & wifi	3 months $850 / mo $2,550
3 Workstations	Lenovo Thinkpad X1	$2,000 each $6,000
Software	Pycharm Pro 2025.2.2, Github	$300/ license, $36 3 months $936
Training	Included in proposal cost	No charge

Quarterly Assessment	3 days staff time	$170/hr x 24 hours $4,080 quarterly $16,320 annually
Annual Update	3 days staff time	$170/hr x 24 hours $4,080 annually
	Total	$60,486 - 80,886

Part C: Application

Part C is your submitted application. This part of the document can be left blank or used to include a list of any submitted files or links.

The minimal requirements of the submitted *application* are as follows:

1. **The application functions as described.** Following the 'User Guide' in part D, the evaluator must be able to review your application on a Windows 10 machine successfully.
2. **A mathematical algorithm applied to data,** e.g., supervised, unsupervised, or reinforced machine learning method.
3. **A "user interface."** Following the 'User Guide' in part D, the client must be able to use the application to solve the proposed problem (as described in parts A, B, and D). For example, the client can input variables, and the application outputs a prediction.
4. **Three visualizations.** The visualizations can be included separately when including them in the application is not ideal or possible; e.g., the visualizations describe proprietary data, but the application is customer-facing.
5. **Submitted files and links are static and accessible.** All data, source code, and links must be accessible to evaluators on a Windows 10 machine. If parts of the project can be modified after submission, matching source files must be submitted. For example, if the application is a website or hosted notebook, the `.html` or `.ipynb` files must be submitted directly to assessments.

Ideally, submitted applications should be reviewable using either Windows or Mac OS, e.g., Jupyter notebooks, webpages, Python projects, etc. If the source files exceed the 200 MB limit, consider providing screenshots or a Panopto video of the functioning application and contact your course instructor.

Part D: Post-implementation Report

Create a post-implementation as outlined below. Provide sufficient detail so that a reader knowledgeable in computer science but unfamiliar with your project can understand what you have accomplished. Using examples and visualizations (including screenshots) beyond the three required is recommended (but not required). **Write everything in the past tense.**

Solution Summary

□ Summarize the problem and solution.
□ Describe how the application solves the problem from parts A and B.

Moving On Up Real Estate, located in King County, Washington, was interested in expanding their business model to include direct buying from sellers, or iBuying. To enter the market, they needed to be able to estimate a property's value quickly and accurately. Several models were developed and tested using a dataset of residential property sales from 2015.

In all, 7 models were trained and evaluated via metrics for performance. Of the 7, Support Vector Regression and Extreme Gradient Boosting performed the best. Additional tuning of hyperparameters via grid search and random search should provide additional performance gains in accuracy. After training and testing, the models were also saved for incremental improvement. In the event they are found to be overfitting in the future, the models can be rolled back to a previously saved version.

In order to speed up response times to sellers, the principals at Moving On Up Real Estate envisioned a simple widget that could be deployed on their existing company intranet. The widget developed allows a user to input some basic property information and receive a sales price prediction. For this proof-of-concept model, users were able to input the data points found to have the highest correlation to sales price. For this version, those inputs are living square footage, square footage above ground, average square footage of nearest 15 homes, number of bedrooms, number of bathrooms, grade of construction and whether a property has a waterfront location.

Data Summary

□ Provide the source of the raw data, how the data was collected, or how it was simulated.
□ Describe how data was processed and managed throughout the application development life cycle: design, development, maintenance, etc.

DATA SOURCE
The dataset utilized was downloaded from www.kaggle.com. It is named "House Sales in King County, USA". The original dataset is a .csv file and can be downloaded directly from the following link:

https://www.kaggle.com/datasets/harlfoxem/housesalesprediction/data

DESIGN STAGE
The dataset selected was chosen for its size, collection timeframe and relevant details. With a sample size of 21,614, the dataset is likely sufficient to test out the feasibility of the tool and mockup a prototype without requiring extensive cloud computing resources. The data provided reflects the sales in King County over the course of one year which is sufficient to cover all seasons without worrying too much about larger economic factors and trends such as interest rates or market bubbles.

Following download, the .csv file read into a Jupyter notebook via the Pandas .read_csv() function and converted to a DataFrame for further analysis. During the initial review, the definitions for the columns had not been discovered and the meaning of several column headings was not readily apparent, ie. grade, sqft_living15, and sqft_lot15. Later, the column definitions were located, and they are included elsewhere in this report as well as within the Jupyter Notebook. The data consisted of an identifier column with a unique id for each property, several categorical columns such as waterfront, grade, and condition, as well as continuous features such as several columns regarding square footage, and a sales price column.

DEVELOPMENT STAGE

Initial analysis was started by utilizing the .head() method to show the shape of the DataFrame as well as the first few rows of data. Examining the first few rows allowed confirmation of the number of columns, column names, and initial impression of data types that would be encountered. Before modifying the DataFrame, a check for duplicates was run using .duplicated().

Initially, the intention was to drop the "id" column as potentially identifying data. However, became apparent later in the process that the "id" might be the best method to use when preforming data manipulation on columns whether it might be imputing missing data, removing rows, etc. While the collector of the information may have known how the id related to each problem, that information was not available during this project so id was used as a unique numerical identifier instead of something that might be used to identify a property in the real world.

Data types were checked to determine whether any manipulations were necessary. If "date" were to be used for modeling, it would have needed conversion into a datetime format. However, since the collection period was only one year, it was determined that "date" should be omitted from model training.

Both .inf() and .isnull() indicated that there were no null values. Every column contained 21,613 values. The .isnull() method also report zero null values for every column. A more complete inspection might also search for common characters that are used to indicate null values such as dashes, forward slash, underline, and question mark since it is not clear which format was used when the data was collected and input.

Initially, it was fairly apparent that the "view" and "waterfront" columns would be categorical data. Applying .valuecounts() to each confirmed that there were 5 view options and two waterfront options. In order to check for additional categorical data columns, .nunique() was run on the entire DataFrame. Results showed that "condition" and "grade" were also likely categorical columns. With 70 options, zip code was not as readily apparent, but after considering that a zip code is a grouping of items rather than a mathematical representation, it was determined that zip code should be treated as a categorical identifier as well.

At this point, a fairly good image of the data had been developed. The next step would be to look for errors, identify outliers, and find missing data that would require imputation. The first step was to run .describe() on the DataFrame. At the same time the date column was converted so that a minimum and maximum could be confirmed as well. The average home sold in this dataset was a 3 bedroom, 2 bath with just under 2,100 square feet of living space with a grade just over the middle grade of 7 and a sales price of roughly $540,000.

Due to unusual minimums of 0 for bedrooms and bathrooms, it was clear that these would need further examination for potential elimination or imputing of data. Price, bedrooms, bathrooms, and sqft_lot also had what seemed to be exceptionally high values. These would also be examined as potential errors or outliers that could be eliminated to improve the data for the algorithms.

After examining the rows that contained either zero bedrooms, zero bathrooms or both, there were 16 transactions total that were affected. Using the property ids, they were also tested to determine whether the same property had be sold more than once. However, none of the id's occurred more than once in the dataset, so the missing data could not be located elsewhere in the data. With only 16 of the 21,613 rows affected, it was decided to drop the transaction that had missing bedroom and bathroom data.

Histograms were also plotted for the entire dataset to look for skewing and for outliers. Several categories were identified for further investigation in the future. Price was heavily skewed to the left. Since price was going to be the label for the training, it warranted a closer examination. Bedrooms, with a max of 33, also was a candidate to get a closer look. Other categories that needed more attention included bathrooms, sqft_living and sqft_lot.

Since latitude and longitude were included in the dataset, they were also used to plot the locations of the transactions. In the initial plot, the shape of King County can clearly be seen. The second plot used transparency to help identify where the most transactions were taking place. Not surprisingly, the coastline and areas north and south of Seattle were the most dense with the east side of the county being more sparse as it approaches the Cascades mountain range.

Initial checks for correlation were run using a correlation matrix. The first run looked for correlation over .70 and highlighted those in yellow. The second run utilized a heat map to look for correlation with price. Sqft_living had the highest correlation to sales price, as might be expected, with grade and sqft_above also showing high correlation. The fact that sqft_living_15, which indicates the square footage of the 15 nearest homes, has a high correlation is also interesting in that it is similar to traditional methods that use comparable properties, one of the conditions being that the comparable properties be as near to the subject as possible. It was also interesting to see that bathrooms were more highly correlated with price than bedrooms. This might be because of some abnormalities in the data that require additional attention. The final correlation run sorted them by ascending values just to make it plain how each ranked and simplify the identification of candidates for model training.

Before training, some feature engineering was completed by using log transformation on sqft_living, sqft_above, price, and, sqft_living15 to reduce their scales to align more closely with the other features and eliminate the skew in their histograms.

At this point the 7 features to be used for initial training were identified and the remaining features were dropped. The .head() method was used again just to confirm the desired columns remained.

The training phase started with the splitting of the data into a training set and a test set while randomizing it as part of the same process. Data was also scaled using StandardScaler() which was fit on the training data only, then used to transform the test data to the same scale. After the data was split and scaled, .shape() was called to confirm that the training and test sets were the appropriate size.

The final steps in development were the training, evaluation and saving of the models. 7 models were chosen to be evaluated. The training was completed on the training data for each. Then, each model was evaluated on the test data. Metrics were run on each as well. At the conclusion of the testing phase, it appeared that Support Vector Regression and Extreme Gradient Boosting had an edge over the other models. Importances were also examined to see what factors the models found most important. Lasso Regression showed importances of all zeroes, which is a common problem with Lasso Regression. Additional tuning would be needed to prevent it from squashing all parameters to zero. At the opposite end of the spectrum, Extreme Gradient Boosting relied very heavily on the grade assigned to each property.

MAINTENANCE STAGE

Maintenance of the final product should consist of periodic monitoring to check for drift, data updates that incorporate new sales, and error logging and handling to address any issues encountered by users.

Due to the consistency of the "grade" applied to the properties, this dataset is likely produced by a tax assessor or county property appraiser for tax purposes. If that is the case, the assessment of properties dataset may only be updated once per year. Ideally, a dataset used to predict real estate sales values would be batch updated at least twice per year, or even quarterly if the market is changing quickly. However, for purposes of this project, batch updating was scheduled on an annual basis due to the availability of new data.

To support the update process, a data pipeline was also developed to ensure that new data is preprocessed and feature engineered using the same methods and logic as the original models. The data pipeline provides automation so simplify the update process, and provides consistency so that updated models receive training and test data that was prepared exactly as during the original run.

Monthly monitoring and quality assurance tests should be performed to identify data drift. Monitoring and testing will use histograms and numerical analysis to locate changes in value ranges over time.

Lastly, any errors reported by users after rollout should be logged and handled to improve the product over time.

Machine Learning

For each machine learning model (at least one is required), provide the following:

- Identify the method and what it does (the "what"). It's advisable to include an example of the model's output.
- Describe how the method was developed (the "how").

□ Justify the selection and development of the method (the “why”).

This project developed a proof-of-concept widget to be deployed on the intranet of Moving On Up Real Estate. The widget is used to predict sales prices of homes after taking input regarding some basic features of the home. Once the features are entered in the respective text boxes, the user clicks on the predict button to submit the values to the trained machine learning model. The model returns a predicted value which is displayed just below the Predict button.

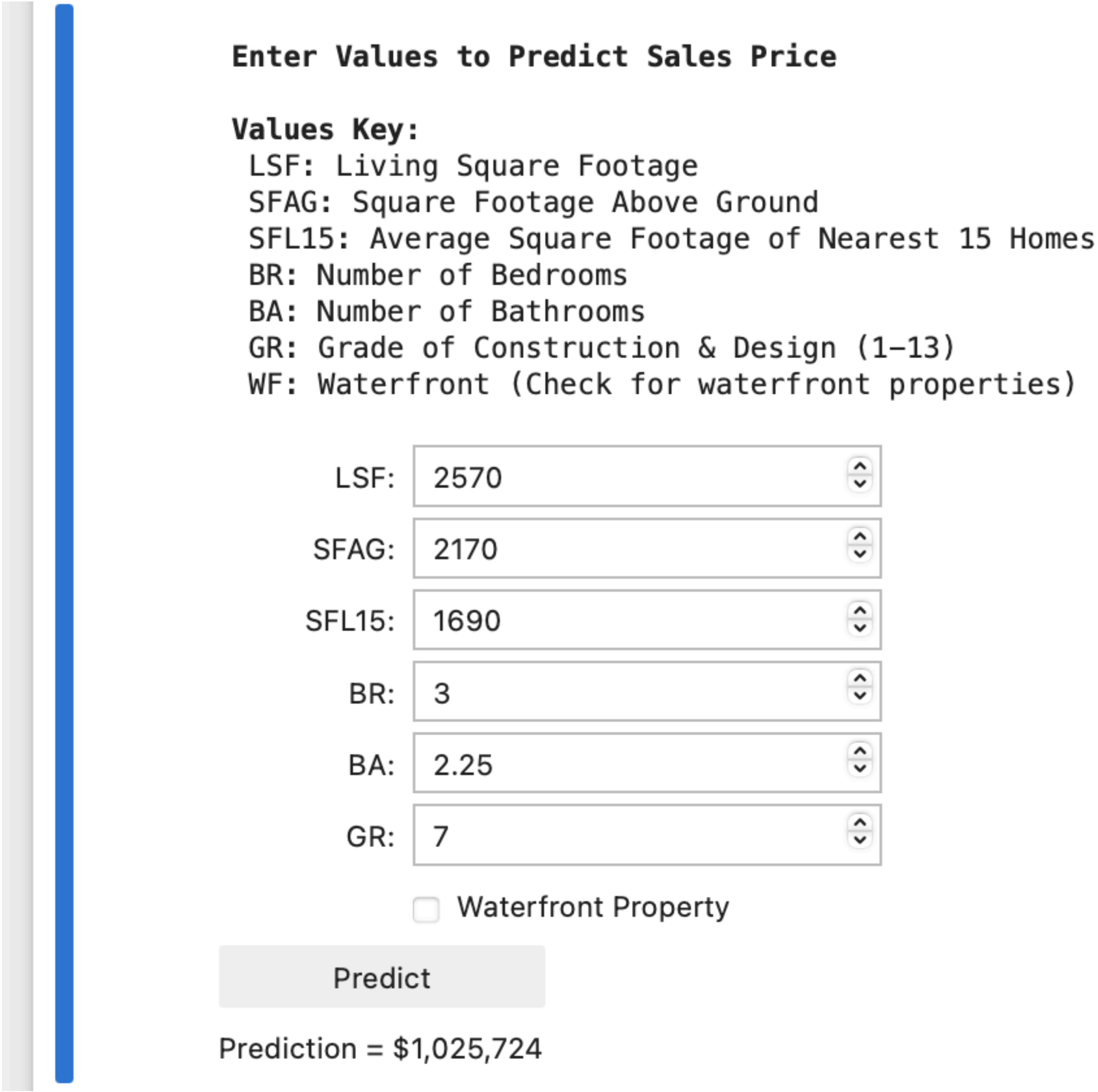
Machine learning was selected as a solution for this project because real estate prices can be affected by many factors. It is possible that traditional methods using simple price per sf models or manual appraisals miss patterns or relationships in the data that machine learning is especially good at identifying.

Machine learning excels at analyzing large amounts of data and locating subtle patterns and relationships over the course of many training runs. In this case, the models automatically detected which of the categories of data presented were the most important as well as how they affected each other, especially when the relationships are non-linear.

After development and initial testing, it appears that the Extreme Gradient Boosting model performs the task the best. Based on the evaluation metrics it is apparent that additional data manipulation, feature engineering and refinement of training parameters is required to reach a useful prediction. However, it appears that the desired outcome is a possibility.

Extreme Gradient Boosting is a machine learning algorithm based on decision trees. Multiple layers of decision trees are utilized, with each successive layer attempting to correct the difference in the predicted and actual values of the previous layer. This process is called sequential learning, where the output of the previous learning cycle is used as input for the next learning cycle. Extreme Gradient Boosting also uses gradient descent to iteratively improve predictions by using the errors of the previous trees. Since overfitting is a regular concern in machine learning, Extreme Gradient Boosting also uses regularization to prevent overfitting. Lastly, Extreme Gradient Boosting can build trees in parallel to speed up the process, making it faster especially on larger datasets.

Below is an example of the output of the Extreme Gradient Boosting model prediction output.



```
Enter Values to Predict Sales Price

Values Key:
 LSF: Living Square Footage
 SFAG: Square Footage Above Ground
 SFL15: Average Square Footage of Nearest 15 Homes
 BR: Number of Bedrooms
 BA: Number of Bathrooms
 GR: Grade of Construction & Design (1-13)
 WF: Waterfront (Check for waterfront properties)
```

LSF: 2570

SFAG: 2170

SFL15: 1690

BR: 3

BA: 2.25

GR: 7

□ Waterfront Property

Prediction = $1,025,724

Validation

For each machine learning algorithm described in the section above, do the following:

□ Identify the model's machine learning category, e.g., supervised, unsupervised, or reinforced. For blended approaches, identify the category most relevant to the model's application.
□ An appropriate validation method for the model's performance.

For supervised learning and reinforced learning

□ Describe an appropriate metric(s) for testing the model's performance.
□ Provide results of testing using the described metric.

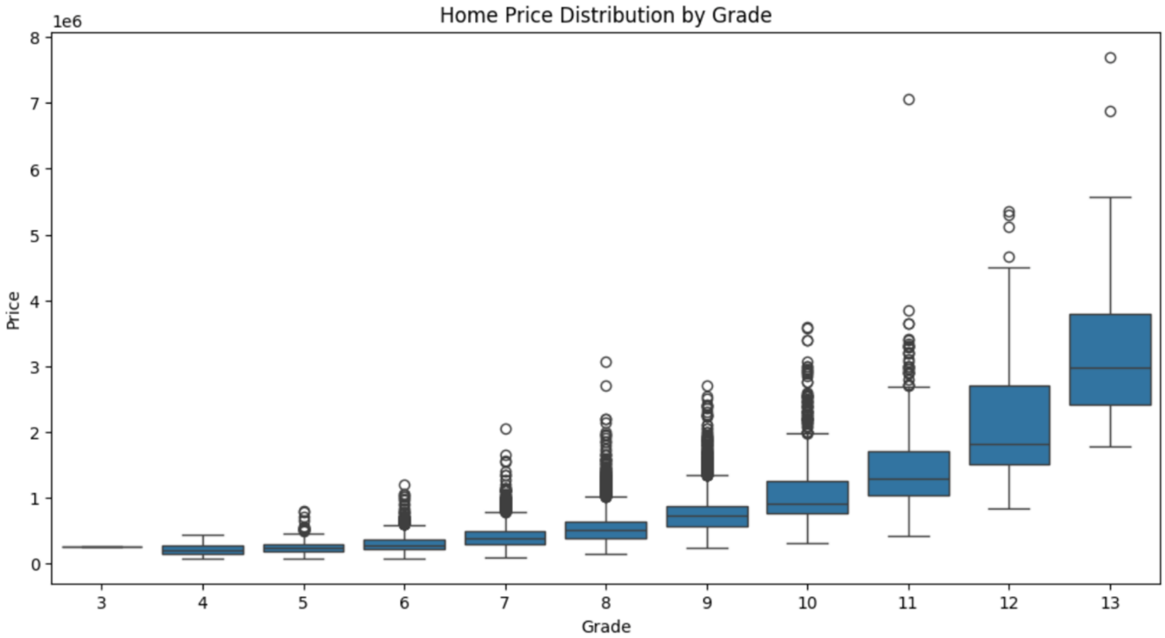
Extreme Gradient Boosting is an ensemble supervised model that uses boosting to combine lots of "weak learner" decision trees into a strong predictive model that works well for regression problems. The evaluation method selected was a randomized test/train split with 80 percent of the dataset allotted to training and 20 percent of the dataset set aside for testing. With this split, 5 metrics were used to compare the predicted values produced by the model to the known values reserved in the test set.

Performance of the models was measured using the following metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R-squared), and Mean Absolute Percentage Error (MAPE). Mean Squared Error and Root Mean Squared Error both square the differences between predicted and actual values for results that are always positive. RMSE then takes the square root to return the scale to the same as the training scale, where MSE results are squared compared to the training units. In this version, the MSE for Extreme Gradient Boosting was 0.11 and the RMSE was 0.33. The Mean Absolute Error measures the average absolute differences between predicted and test values without squaring so all errors are equally weighted. The MAE for this version was 0.27. When converted back to dollar equivalent, the result was just over $140,600 which is far too wide a margin of error for the proposed sales price predictor. The Coefficient of Determination, or R squared, shows the proportion of the variance in the target variable accounted for in the model. This metric ranges from 0 to 1, with higher values indicating a better model fit. In this version, the r squared was 0.61. To be useful in sales price prediction, we are looking for an r squared closer to .95 - .99, if possible. Finally, the Mean Absolute Percentage Error will give us the absolute error as a percentage of the actual values, a commonly understood measurement. For this project, we were targeting a Mean Absolute Percentage Error of 5% or less. Our metrics indicate MAPE of 27.76%, which is not even in the ballpark. Our team will need to revisit the data understanding, data preparation, and modeling steps to determine whether our data pipeline can be improved, data manipulations made more complete, and feature engineering more meaningful to the model.

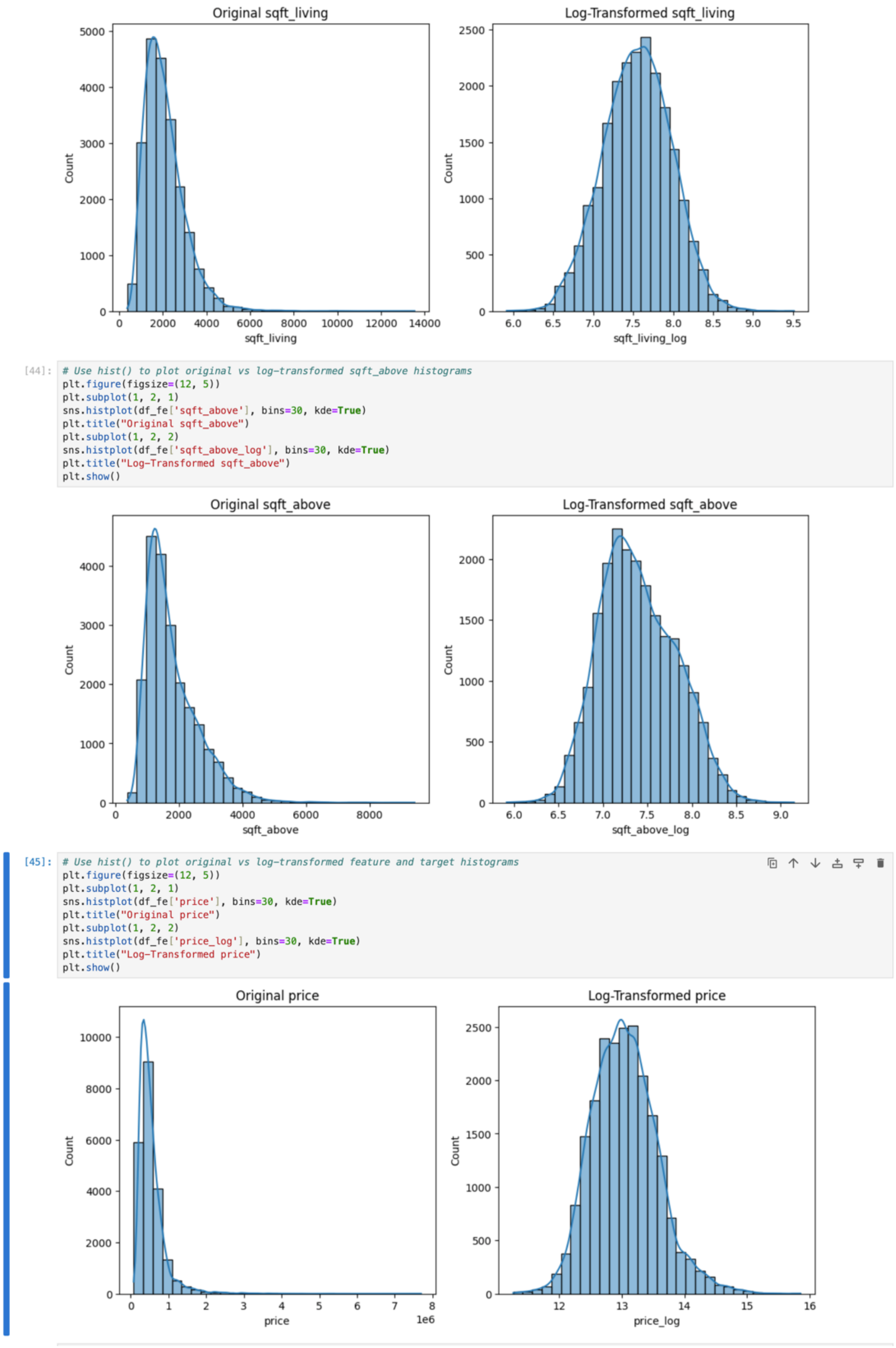
Visualizations

Identify the location of at least three unique visualizations. They can additionally be included here.

This visualization is a box plot comparing price to the assigned grade of the property.



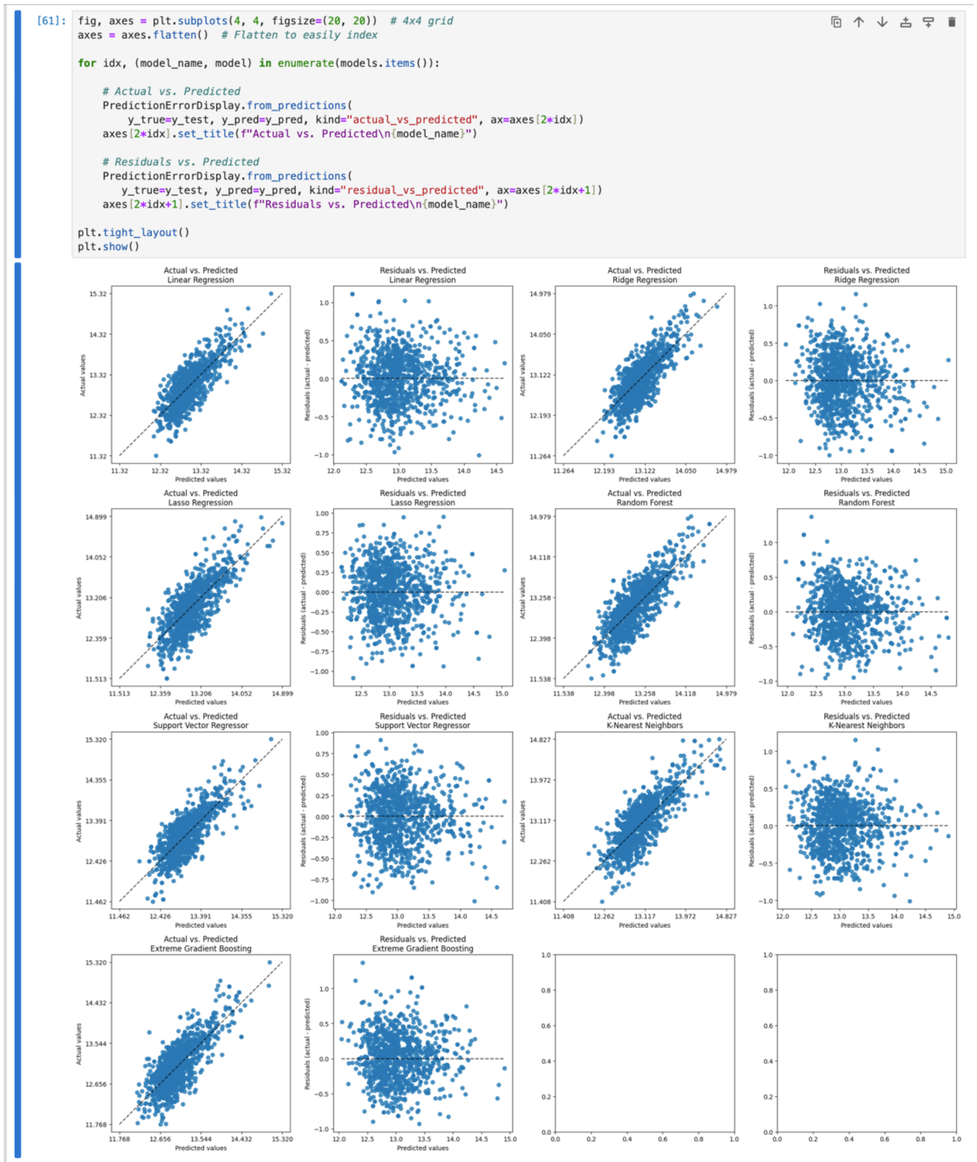
This visualization shows the transformation of the histograms after log transforming several features.



```
# Use hist() to plot original vs log-transformed sqft_above histograms
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.histplot(df_fe['sqft_above'], bins=30, kde=True)
plt.title("Original sqft_above")
plt.subplot(1, 2, 2)
sns.histplot(df_fe['sqft_above_log'], bins=30, kde=True)
plt.title("Log-Transformed sqft_above")
plt.show()
```

```
# Use hist() to plot original vs log-transformed feature and target histograms
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.histplot(df_fe['price'], bins=30, kde=True)
plt.title("Original price")
plt.subplot(1, 2, 2)
sns.histplot(df_fe['price_log'], bins=30, kde=True)
plt.title("Log-Transformed price")
plt.show()
```

This visualization contains the actual test values vs the predicted values on the initial version of the project.



[61]:
```
fig, axes = plt.subplots(4, 4, figsize=(20, 20))  # 4x4 grid
axes = axes.flatten()  # Flatten to easily index

for idx, (model_name, model) in enumerate(models.items()):

    # Actual vs. Predicted
    PredictionErrorDisplay.from_predictions(
        y_true=y_test, y_pred=y_pred, kind="actual_vs_predicted", ax=axes[2*idx])
    axes[2*idx].set_title(f"Actual vs. Predicted\n{model_name}")

    # Residuals vs. Predicted
    PredictionErrorDisplay.from_predictions(
       y_true=y_test, y_pred=y_pred, kind="residual_vs_predicted", ax=axes[2*idx+1])
    axes[2*idx+1].set_title(f"Residuals vs. Predicted\n{model_name}")

plt.tight_layout()
plt.show()
```

User Guide

Include an enumerated (steps 1, 2, 3, etc.) guide to execute and use your application.

- Include instructions for downloading and installing any necessary software or libraries.
- Give an example of how the client should use the application.

1. Ensure a web browser is installed
2. Install Python – Windows, Mac and Linux versions can be downloaded here
3. When prompted, be sure to select "add Python to PATH"
4. Install Jupyter Notebook – `python -m pip install jupyter`
5. Install numpy – `pip install numpy`
6. Install pandas – `pip install pandas`
7. Install matplotlib – `pip install matplotlib`
8. Install seaborn – `pip install seaborn`

9. Install scikit-learn – `pip install scikit-learn`
10. Install xgboost – `pip install xgboost`
11. Download Jupyter Notebook file and .csv file to the same directory.
12. Start Jupyter notebook with a console or terminal command – `jupyter notebook`
13. A browser window will open and display a file browser.
14. Browse to the directory where the Jupyter Notebook file and .csv file were saved.
15. Open the Jupyter Notebook file. It will open in a second browser window.
16. Code is segmented and each section is run by selecting it, then clicking the Play button at the top of the notebook window.
17. Once a segment is complete, it will automatically advance to the next section.
18. Comments at the top of a segment describe what is happening.
19. Comments at the bottom of a segment evaluate the outcome or note observations for future versions.
20. The price prediction widget is in the final cell.
21. The final cell will have default values in the prediction tool. Values can be updated and changed for testing.
22. Once complete, simply close the browser window containing the Jupyter notebook
23. Command C on Mac or Ctrl C on Windows will interrupt and present the shutdown confirmation.
24. Confirm shutdown by typing y.

Reference Page

Include references for cited works, e.g., (Author, year) following an accepted writing style. References are not required; this page can be removed if no references are used. To cite sources used for code, you should include the references as code comments within the source code.

1. Géron, A. (2022). *Hands-On machine learning with Scikit-Learn, Keras, and TensorFlow*(3rd ed.). O'Reilly Media.

2. Sruthy. Software Testing Help. (n.d). *Data mining process*. Retrieved 09/17/25, from https://www.softwaretestinghelp.com/data-mining-process/?openInDeviceBrowser=true#Data_Mining_Models

3. Glen, S. Data Science Central. *Decision Tree vs Random Forest vs Gradient Boosting Machines: Explained Simply*. July 28, 2019. Retrieved September 17, 2025, from https://www.datasciencecentral.com/decision-tree-vs-random-forest-vs-boosted-trees-explained/

4. Data Science PM. (2024) *What is CRISP DM?* Retrieved 10/01/25. https://www.datascience-pm.com/crisp-dm-2/