**Residential Valuation Intelligence System**

Karthik Nimmagadda

Department of Applied Data Science, San Jose State University

Data 270: Data Analytics Process

Dr. Linsey Pang

February 11, 2024

**Abstract**

Residential Valuation Intelligence System (RVIS) is a smart assistant, launched into the dynamic real estate market, in that it helps revolutionize by predicting house prices very accurately. This makes for quick and informed decisions in a very changing market. It involves the preprocessing of data and deriving features to identify the important determinants in the first step, followed by building advanced predictive models, namely Elastic Net, Random Forest, XGBoost, and Gradient Boosting, to fine-tune their performance. The models are built on integrated development environment that is easy to use, and they detail the insights in the market. The intention is to present a predictive model that is very accurate in the forecasting of price changes. It calculates the performance of the system based on some metrics like MAE, MSE, and R-squared which reflect the precision and reliability of the model. This highlights how proper implementation, testing, and validation will help to guarantee that the system is robust and reliable. The system is a true game-changer in the real estate industry, given that it is a unique tool for providing stakeholders with strategic pricing and decision-making information support to maximize value for sellers, buyers, and agents with data-driven transactions. This really does represent a new era when it comes to data-driven decision-making in real estate. This process really does illustrate how important intensive testing and validation are in the creation of robust and reliable technology solutions.

**1. Introduction**

**1.1. Project Background and Executive Summary**

        Valuing residential properties has always been a very tricky proposition, as there's a lot of room for error mostly due to manual assessments and only a small set of data points being considered. The "Residential Valuation Intelligence System" (RVIS) project sits at the crossroads of real estate with advanced machine learning technologies, looking to bring forth a transformation in the way real property is appraised. The RVIS would cover an ambit that has a wider span of data points and uses pretty sophisticated machine learning algorithms to provide a more precise, efficient, and transparent process of valuation.

        Property valuation is a very key issue to many stakeholders in the real estate market, such as investors, homeowners, lenders, and insurance companies. Errors in valuation could result in financial losses, misjudged decisions, and inefficiency of the market. Our project tries to overcome these challenges by leveraging Machine Learning to make property valuation way more accurate and faster, which will ultimately make real estate transactions and property-related financial services a lot more reliable.

        This project is an initiative designed solely to challenge the inefficiencies and inaccuracies inherently ingrained in the traditional ways of property valuation. If used, in any case, these traditional approaches would fail to realize the scope of the factors that affect the pricing of an estate, including physical features and location specifics reaching up to market trends. The project aims at developing a valuation model that is more dynamic and inclusive therefore it can be flexible enough to adopt real-time observable changes in the market and, at the same time, wide enough to accommodate a huge variety of data points.

        In effect, the main goal for the Residential Valuation Intelligence System project is to leverage machine learning towards the transformation of the real estate valuation horizon. First, there is an aim at developing a strong valuation framework with the machine learning, characterized by the approaches to data analysis that are all-inclusive of various relevant factors affecting property values. Our project is aimed at improving the level of accuracy and effectiveness in the valuations to a level that will save time and resources that normally go into the assessment of properties. Our project will lead to a more transparent market, and the decision-making process of the stakeholders would be able to realize improvements for clearer and accessible valuation information. Ultimately, our project intends to be a pioneering force of change within the paradigm of real estate valuation by setting new standards in which the integration of machine learning technologies might later open up to them broader application possibilities for such advancements of technology in various facets of real estate and finance.

        Our project ensures a procedural and rigorous approach that could be applied to the steps of data collection, model building, and deployment. To begin with, the project will focus on the complete gathering and scraping the data from the website zillow.com that will incorporate so many things, including property attributes, the current market trend situations, and any other information necessarily required for valuation. Then Data Exploration and Cleaning (Dealing Outliers and Null Values) and then, Model Building by using regression models of Machine Learning like, Elastic Net, Random Forest, XG Boost and Gradient Boosting are used in the development of the collected information to develop projections of the real estate values. Model Evaluation assessment will be done based on MSE, MAE, R2 after every model has been constructed to find the precision and effectiveness of every model in forecasting.

        Our project makes important strides in real estate valuation and therefore, it will definitely make a huge impact on the larger real estate market in several key main ways. It, therefore, gives the hope of better accuracy in valuations with minimum differences, if any, between the property values in the case of disputes by using the potential of machine learning algorithms. The ability to easily obtain near real-time, data-derived valuations—especially if such valuations become an expected part of the due diligence process—should bring about new levels of market efficiency as more of those involved will have such information readily available to make transactions more transparent and aid their investment decisions. Looking ahead to pure assessment, this platform will be evolving with predictive analytics, which will provide real estate trend foresights, investment opportunities, and risk assessment for investment. This project would also contribute very vastly to the scope of research and development, since it would be highly profitable in application within the real estate domain and provide an invaluable insight along with methodologies that can be used within many other sectors. Our project also helps in Insurance Risk Assessment and Urban Planning Development.

**1.2. Project Requirements**

          Residential value prediction has become an essential element of the rapidly growing sector of real estate estimation, made possible by using predictive modeling. Building up a robust model for real estate value prediction is our main aim. We selected to use the dataset of Zillow, representing an array of homes listed from many locations; it gave us the most realistic and diverse sample of the market. This dataset has been scrapped by the following methods: web scraping and API methods. The data set is too complex, which allows for an extensive analysis of various factors. There is an assurance that such important characteristics of properties will be considered in the projection of prices. These are the number of bedrooms and bathrooms, location of the property (city and country), type of home (condone family home, etc.), size of the lot, living area in square feet, geographical coordinates (latitude and longitude), and the price of the property, which includes any changes that might have occurred previously.

          The users engaging this project should be an expert not only in the core principles of machine learning but also in the domain of real estate. They should have the expertise to prepare data in such a way that the approach ensures accuracy in the information provided. As we're employing supervised learning techniques, a deep understanding of various regression models is paramount. This extends from understanding basic models such as linear regression and regularization methods of ridge and lasso to understanding the methods of knowing some more advanced and applied machine learning algorithms like random forest and gradient boosting. In addition, it is crucial to assess the performance of a model to determine its relevance in different scenarios. This involves understanding the assessment of the model metrics, including mean absolute error, mean squared error, and R-squared, that will assure the output of the models in predicting property prices as minimum errors.

**1.3. Project Deliverables**

         This project gives an in-depth approach on how to leverage predictive analytics in the estimation of residential property value, specifically in the analysis and prediction of house prices using the Zillow data set. This is a structured methodology that will start by the acquisition and preprocessing of the Zillow dataset, making sure that quality and consistency are maintained in the data. After that, exploratory data analysis (EDA) techniques were applied in order to unravel the underlying patterns and relationships in the dataset. Next, feature engineering is applied to derive meaningful features from data and retain its model predictive power. Advanced machine-learning algorithms, including regression techniques like linear regression, random forest regression, and gradient boosting, shall be put to use for model development with rigorous evaluation using appropriate metrics to check model accuracy and generalization capabilities. The project will result in delivering comprehensive documentation reports, including a summarizing presentation that will encapsulate the methodology used, the findings, and the key insights from the analysis.

This structured approach ensures clear expectations and measurement of success underlined by the Agile Principle through iterative development cycles during the project lifecycle. Adhering to a well-defined roadmap and employing adaptive strategies that are tailored to specific goals and constraints, this project will deliver actionable insights and a strong predictive analytics solution for residential property value estimation to its stakeholders.

**1.4. Technology and Solution Survey**

Dealing with this complexity of residential Valuation Intelligence Systems is a host of modern-day technologies and solutions emerging that offer diverse methodologies meant to improve the accuracy and reliability of property valuation. It therefore emerges that standing at the forefront is the line of machine learning models that include linear and logistic regression, elastic net, kernel ridge, Lasso, Random Forest, SVM (Support Vector Machine), XGBoost, LGBM (Light Gradient Boosting Machine), and Gradient boosting, all because of their adaptive efficiencies to parse through the volume of the dataset and recognize the underlying patterns relevant in real estate pricing (Patel et al., 2023). These models allow for the valuation from all dimensions: by use of historical data, trends in the market, and characteristics of the property in making predictions that will give an estimate of the correct price. Further cross-validation techniques fine-tune model performance for robustness and avoiding overfitting toward enhancement of predictive reliability. A novel way in this direction is the use of model stacking, through which the predictive power of base models gets channeled for the development of an ensemble model that would have better predictive power and capture the confluence of strengths of individual constituent models. This harmonization of the diverse paradigms of machine learning provides for a comprehensive frame in real estate valuation that encapsulates the broad spectrum of factors of basic property characteristics to the very intricate market dynamics, therefore providing a nuanced, detailed perspective concerning the same (Patel et al., 2023).

The modern and competitive field of valuation in residential real estate, increasingly demands cutting-edge machine learning (ML) models to make sure it reaches the most accurate predictions in house pricing. The most existing methods in this field are linear regression, decision tree regression, k-means regression, random forest regression, logistic regression, elastic net, kernel ridge, lasso, support vector machines (SVM), extreme gradient boosting (XGBoost), light gradient boosting machine (LGBM), and gradient boosting. These models will also come in handy, not only for better price prediction but also for finetuning the predicament according to individual budget constraints and preferences. The technique of cross-validation proves handy not only in making the model more reliable by not letting it get overfitted but also for the sake that there is an assurance that the developed model is making more generalized predictions. It will mix and further improve the model stacking strategies to achieve precision by combining the predictive power of the best-performing models. This is through a highly sophisticated ensemble approach that captures the predictive performance from the best-predicting models. This comprehensive survey brings out the importance of embracing a multivariate ML approach, characterized by regression and ensemble techniques, to effectively navigate the labyrinth of the real estate market. This would be imperative in customizing the orientation of Residential Valuation Intelligence that best meets the complex requirements of the real estate sector in ensuring the accuracy yet adaptability of forecasting methodologies (Rawool et al., n.d.; Patel et al., n.d.).

Even in the emerging area of residential real estate valuation, the most complex machine learning (ML) models gain space, such as Lasso Regression and Gradient Boosting Regression, as further epitomized by Lu et al. What makes this approach especially interesting and relevant is the prediction of the price of an individual house considering myriad factors like location, type of house, and size along with local amenities. This paper by Lu et al. underlines the importance of creative feature engineering and the use of a hybrid model of Lasso and Gradient Boosting Regression, which showed some promising results on the Kaggle Challenge, as it ranked 1% among the competing teams. This is exactly in line with the needs that such a system would require: Linear and Logistic Regression, Elastic Net, Kernel Ridge, Lasso, Random Forest, SVM, XGBoost, LGBM, and Gradient Boosting are only a varied range of ML models that the Residential Valuation Intelligence System would require. As mentioned above, it increases the validity and accuracy of the predictions with cross-validation and model stacking. All these contribute to understanding the market dynamics and customer tastes in such a manner that a sturdy and elastic system of real estate valuation can be designed (Lu et al., n.d.).

In real estate appraisal and, notably, in forecasting housing prices at the individual level, Truong et al. (2020) revealed a wider reviewed landscape of traditional, advanced, and state-of-the-art machine learning (ML) methods. It indicates the use of various ML models to predict the complex interconnections of housing prices with multiple influencing factors applied to datasets that capture such complexity in the real estate market in Beijing. The methodologies proposed would be highly based on rigorous data preprocessing, rich feature engineering, and the application of various state-of-the-art ML models like Random Forest, eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM). The study investigates further with hybrid regression models and stacked generalization to optimize predictive accuracy. Just the kind of approach in line with model selection and implementation would appeal to a Residential Valuation Intelligence System, harmonizing models ranging from Linear Regression to Lasso, Elastic Net, and Kernel Ridge, among many others. Cross-validation and model stacking, with a focus on concentration, are pointers to the rise of model reliability and improvement of predictions in the complex field of residential valuation. The innovative data handling and model optimization strategies of this paper provide valuable insight and methodology toward operationalizing both data and technology in building a robust and adaptive real estate valuation system. The 2020 study of Truong et al. provides a valuable approach and methodology toward operationalizing both multifaceted data and technology in building a robust and adaptive real estate valuation system.

In that regard, Aghav, Avhad, Nanaware, and Gudekar undertook a study that applied machine learning algorithms decision tree, lasso, and linear regression in the prediction of house prices by taking a dataset with features that include location, square footage, number of bedrooms, and bathrooms. This might also be because the public dataset data and the Kaggle website are also involved, such as preprocessing techniques such as missing value handling, normalizing features, and one-hot encoding of categorical variables. As their approach, they highlighted how the machine learning model works better than the previous traditional regression ones, with Linear Regression being best with a rate of accuracy at 85%, followed by Lasso at 72%, and then the Decision Tree at 71%. With the Grid Search Cross-Validation, hyperparameters of the Linear Regression model were fine-tuned, hence proper optimization of the hyperparameter. It is in line with the objective of the residential valuation intelligence system project of incorporating a broad spectrum of ML models such as Linear Regression, Logistic Regression, Elastic Net, Kernel Ridge, Lasso, Random Forest, SVM, XGBoost, LGBM, and Gradient Boosting models. This section is of prime importance to the project wherein the focus is maintained on the cross-validation and model stacking by Aghav et al. This comparison will make clear how advanced ML models and ensemble techniques might improve the accuracy of residential property valuations to lay down strong security in building up a comprehensive system for market analysis of real estate (Aghav et al.).

**1.5. Literature Survey of Existing Research**

          Accretions about forecasting housing prices through a wisp of machine learning techniques happen to be quite a hard task under discussion in the research by authors Quang Truong Minh Nguyen, Hy Dang, and Bo Mei. The determination of what many factors influence such a change in housing, be it location, physical characteristics, or economic condition, is an extremely complicated task the study conducted a thorough investigation process and thus gave sound forecasts on the housing prices with both conventional and front-runner machine learning models.

Home price forecasting has been done using a variety of machine learning methods, each with unique benefits and drawbacks. The research paper thoroughly compares and analyses a variety of models, including Random Forest, XGBoost, LightGBM, Hybrid Regression, and Stacked Generalization Regression, to ascertain which approach is most effective for forecasting housing prices. These models are evaluated using performance criteria; the Root Mean Squared Logarithmic Error (RMSLE) is emphasized as an accuracy statistic.

Although Random Forest—a model widely recognized for its reliability and ease of use performs well, its high temporal complexity and tendency toward overfitting cause it to perform badly. Particularly notable for their quickness and effectiveness in handling large datasets are the gradient-boosting models XGBoost and LightGBM, which provide competitive accuracy at the expense of reduced processing time complexity.

In this research, two novel approaches stacked generalization regression and hybrid regression that combine the best features of many models are presented in order to increase prediction accuracy. Hybrid Regression is a more straightforward method that enhances generalization by integrating several models, whereas Stacked Generalization Regression employs a more complex architecture and provides better accuracy by using the predictions of base-level models as inputs for a higher-level model. Despite their effectiveness, one should nevertheless consider the temporal complexity of these ensemble approaches, especially Stacked Generalization Regression with its K-fold cross-validation component.

The paper's findings present the computational efficiency of the trade-offs made by the investigated models. However, for example, ensemble methods like Stacked Generalization Regression have delivered outstanding accuracy; nevertheless, they are in many cases not justified by computational needs or the complexity involved. With this, the paper goes on to suggest further research on the synergetic benefits by combining, such as a multivariate regression model, re-learning the potentiality of models in machine learning, and integration with deep learning techniques to come up with new avenues for predicting housing prices.

What sets their research apart is a critical look into the performance of individual models, which is often left in favor of more popular ones, and a try to validate multiple techniques in the implementation of models on regression. The following part of the study will reveal that the House Price Index (HPI) is inadequate in informing the proper prediction of the housing price of the individual house, considering the location, area, and population. This kind of comprehensive understanding and prediction of the housing prices resonates well with my project objective on the Residential Valuation Intelligence System, which involves using a set of different diverse ML models such as Linear, Logistic Regression, Elastic Net, Kernel Ridge, Lasso, Random Forest, SVM, XGBoost, LGBM, etc. - among others - for making AI-empowered predictions and optimal decision support. Therefore, my work rightfully puts an emphasis on evaluation using cross-validation to improve the prediction precision by model stacking to find resonance of the importance given by the methodology of Truong et al. Further, it's necessary to stress the importance of synergy between advanced machine learning techniques and the intricate dynamics of the real estate market (Truong, Nguyen, Dang, & Mei, 2020).

The work done by Chaurasia and Haq took a great step in the use of machine learning for the prediction of housing prices. The study greatly demonstrates how complex it is to make accurate predictions concerning house prices since they constitute a prediction that involves quite a variety of data which is also affected by geography. The proposed model, consisting of a suite of machine learning algorithms, along with data preprocessing techniques, performs the existing prediction method clearly in comparison with the real-time housing data. The study's emphasis on linear regression and how it affects the accuracy of the model is one of its main conclusions. The chief statistical approach of predictive modeling, linear regression, is used to determine the home prices, which further contributes toward the very strength of the prediction of the model. It focuses on the performance of machine learning against traditional models of prediction, which require expert analysis, with a touch of statistical procedures, and gives information on how well the predictors predict the house price.

The work of Chaurasia and Haq is noted as an integrated disciplined approach to the majority of machine learning techniques with preprocessing of data, and its resultants give the model increased accuracy in forecasting, scalable at the different market situation levels. The results of this research are of great consequence to the real estate sector, as it avails to the stakeholders more sophisticated tools while opening the door for greater investigations into the use of machine learning for real estate price prediction in the future.

This contentious, difficult question of home price prediction is attempted to be answered by Aman Chaurasia and Inam Ul Haq based on their comprehensive study, which is conducted through a machine learning-based methodology. And they know how difficult it is to estimate house prices with such a range in data, but area-dependent. Largely improving the predictive accuracy in predicting house prices, their study presents a predictive model based on combined use with a variety of machine learning algorithms and data pre-processing approaches.

The authors first briefly outline the importance of accurate housing price forecasts: an endeavor that will ensure maximum profits accruing to investors, individual buyers, and sellers. Now, the traditional limitations, based essentially on expert interpretation and statistical methodology, are being compensated and in some cases replaced by more dynamic and adaptive machine learning models. A lot of data is exploited to find underlying patterns and relationships that, as a result, make the models give correct predictions, even for previously unknown data.

The model developed by Chaurasia and Haq tells about an outstanding fact: this model uses two different approaches. First, the quality of data has been preprocessed, and in another, it uses different ways or machine learning to train the model. It, therefore, gives room for more advanced means of data manipulation and analysis; thus, the predictions reached through the advanced means may be more accurate. This further indicates the usefulness and superiority of their proposed model in comparison to real-time home price data by very large margins in almost all cases, with significant improvements over past prediction methods. However, there is the extra cautious model that extensively includes linear regression in the study. Linear regression is underpinned as the core statistical approach to predictive modeling, claiming to add precision to the contribution of their home price forecast model. Analyzing how the different variables affect the accuracy of the model is one of the ways through which the authors have been able to reveal some predictors of the home price that are likely to be of great use for the information to the players in the real estate sector.

With that in mind, the works of Chaurasia and Haq are seminal when it pertains to the domain of work with machine learning for the prediction of real estate prices. Thus, the proposed model tends to do better than the conventional means of prediction in these aspects: more accuracy in the results and more easily accommodating different market situations through integrating data pre-processing methodically with a well-chosen bouquet of machine learning algorithms, yet giving a scalable solution. This is an important placing of the finding and critically relevant to the field of real estate in that it provides more understanding tools in aiding the numerous actors, be they investors, sellers, or buyers, in understanding the dynamics of the environment with which housing markets operate.

Chaurasia and Haq's research introduces a sophisticated approach that cleverly blends data-cleaning techniques with a variety of machine-learning tools to get better at predicting how much houses will sell for. They tested their approach with up-to-date info from the housing market, and the results were pretty impressive, showing that their method could do a better job than the usual ways of guessing house prices. This is super relevant to a project I'm working on called the Residential Valuation Intelligence System, where I'm also using a bunch of different machine learning techniques, like Linear and Logistic Regression, Elastic Net, and a few others, to try and figure out property values more accurately. My project also uses some smart strategies like cross-validation and stacking models on top of each other to make the predictions as good as they can be, which is pretty much what Chaurasia and Haq were doing. Their work highlights how using machine learning can make a big difference in understanding and predicting real estate prices better (Chaurasia & Haq, 2023)

Maida Ahtesham, Narmeen Zakaria Bawany, and Kiran Fatima tackled the tricky task of figuring out house prices in Karachi, a big city that's changing all the time. They used some smart computer programs called Gradient Boosting Model and XGBoost because they wanted to get a fresh look at Pakistan's housing market, a place not many people have studied this way before. They looked at a bunch of things that can change a house's price, like how nice the house is, where it's located, and how many rooms it has. Usually, you'd need to know a lot about local real estate to make good guesses about these things, but the authors thought computers could do it better and cheaper. They used a huge pile of information for their study almost 39,000 pieces of data about houses in Karachi that anyone can look up online. This was a big step in learning about house prices in the area. Through the research, it was found that the XGBoost program was good at working through all this complicated info and could guess house prices with about 98% accuracy. That's thanks to some clever tricks it has, like filling in missing pieces of data, making sure it doesn't jump to conclusions too fast, and knowing when to stop adding more details to its calculations.

Their deep dive into Karachi's housing market, using all this data and the XGBoost program, showed exactly what makes house prices go up or down in the city. This is super helpful for people looking to buy or sell houses because it gives them the inside scoop to make smart choices. Plus, it gives experts and city planners a heads-up on what might happen next in the market. The works of Maida, Narmeen, and Kiran are a big deal because it shows how we can use technology to understand housing markets better, especially in places like Karachi where this kind of study hasn't been done much before. The computational experiments are quite thorough, developing a model with high accuracy and low Mean Absolute Error (MAE), by using different train-test ratios for robustness and reliability of the model. This makes it be an added tool to the stakeholders within the real estate. XGBoost was applied by Maida Ahtesham, Narmeen Zakaria Bawany, and Kiran Fatima in their research "House Price Prediction in Karachi, Pakistan" for the preparation of a dataset having 38,961 records taken from an open real estate portal in the country to enhance the level of accuracy in the estimation of property in the location where former analysis has never been done. This is a very appropriate methodology that suits well for use in my project on the Residential Valuation Intelligence System, using many machine learning models, including XGBoost, in a bid to give a best-fit model by emphasizing cross-validation and stack. 98% excellent prediction accuracy by Ahtesham et al. is testimony to the outstanding potential of machine learning in real estate valuation.

  Ahtesham, Bawany, and Fatima did a novel work in applying machine learning to predict the price of real estate in Pakistan, especially Karachi. Their successful predictive model opens the way for further investigation into regions that are less explored and sets a precedent toward the use of advanced ML techniques, like XGBoost, for the betterment of the predictability and transparency of the real estate market. This contribution is invaluable for both academic research and practical applications in real estate economics (Ahtesham, Bawany, & Fatima, 2020).

The project proposed in "Real-Estate Price Prediction System using Machine Learning" by Veerraju Gampala, Yaznitha Sai Nalajala, and Bhavya Tadikonda Naga Sai considers ambitiously using machine learning techniques to estimate the value of real estate for an approach to investments, which may be more accurate and empirically based. Some of the algorithms that are going to be considered in the evaluation of estate value will include Random Forest Classifier, Naive Bayes, Logistic Regression, and Decision Trees. These are common questions of felt needs that should be answered by the project: what are the best ways to invest in real estate for maximum profit? When and where to purchase, and how much rent or selling price will be expected in the future. It is based on the practical application of artificial intelligence and machine learning to real estate investments. This study applies machine learning to efficiently manage and appraise the large quantity of data that is available to the academic fraternity, hence setting the platform for more research and problem-solving. One of the reasons that the project is conducted is the realization of the most appropriate method for forecasting house pricing through the comparison of the number of machine learning algorithms. The paper runs data with the use of various charting methods. In addition, it applies different types of algorithms in the estimation of the prices of estates, and, lastly, it applies to determining which of the used clusters of algorithms is the most accurate. This evidence marshaled toward this conclusion is awesome. The best way is linear regression, with amazing accuracies at 95% and mere errors at 5% in the mean square. This, therefore, proves that while there are so many other fancy models available for the same purpose, linear regression still stands as the most relevant and valid way for prediction regarding the prices of pieces of real estate today.

Besides, the further recommendation from the research is to integrate a feature-rich user interface with the best and improved machine learning model that may be further added to show the nature with which machine learning may add to real estate price forecasting. Furthermore, the interface has a high potential of enabling clients to make informed decisions on the purchase of real estate and receive good customer support, thus extending the practical implications of this study. Research by Gampala, Sai, and Bhavya is appreciable work in the field of machine learning-based prediction of real estate prices. Most successful of the tested algorithms is, according to the research of Gampala et al., linear regression. This sets the base for future improvements and wider applications—those that may include other geographic regions and advanced UI technologies. This paper presents how machine learning is likely to disrupt the real estate industry with a wave of new resources and insights for consumers and investors (Gampala, V., Sai, N. Y., & Bhavya, T. N. S. (2022)).

This research has dealt with the complexities of the Melbourne housing market in a machine learning price prediction manner. The papers have reported enormous variations in property prices within Melbourne due to many factors, including location, property feature mix, and economic condition. Whereas Phan uses historical transaction data analytic approach models, with the use of machine learning and more centralized in Stepwise regression and Support Vector Machine (SVM) technique models.

The data comprised 34,857 records ranging between the years 2016 and 2018. All had an array of variables containing details of the transaction, house features, and location specifics. Pre-processing of the data, including missing values and handling outliers, was highly emphasized to give quality and reliable analysis. It, therefore, can be said that the combination of Stepwise regression with SVM competes effectively in predicting housing prices with a very high degree of accuracy, according to Phan. This, therefore, concludes the study and, perhaps, recommends the impacts that machine learning in real estate forecasting would have on buyers, sellers, and policymakers. This research is brief about a study on the applicability of machine learning in real estate price forecasting, necessary to bring out the potential benefit of advanced analytical techniques toward understanding and prediction of market dynamics (Phan, 2018).

Chen Chee Kin, Zailan Arabee Bin Abdul Salam, and Kadhar Batcha Nowshath (2022) identify the changing nature in real estate in a digital era with specific reference to tools geared at modern buyers' budgets and awareness. Subsequently, the following is a synthesis of technologies that try to predict house prices, precisely for individual affordability and desires: Machine Learning (ML), the so-called Artificial Neural Networks (ANN), and Chatbot. This special method is particularly applied to house affordability analysis in Malaysia, whereby there are efforts to come up with a model acting as a shortcut in the prediction process, hence an overall synopsis about the market trend and price. This work is an important contribution in that it integrates diverse technological tools in improving the accuracy and accessibility of house price predictions. On the other hand, this provided valuable insights to buyers and sellers on how the dynamics of the real estate market are taking place. Likewise, different ML models are incorporated with close attention paid to the evaluation of effectiveness. That closely relates to the methodologies being employed in my Residential Valuation Intelligence System Project. It also includes the utilization of various ML algorithms, including Linear and Logistic Regression, and Elastic Net, among others, further being cross-validated and hyper-optimized by stacking the best models. The extremely relevant study of Chen et al. therefore provides further backing for the importance of ML and the role of increased digital tools in the betterment of predictive perfection and user experience related to the RS in residential valuations (Chen, Kin, Salam, & Nowshath,2022).

In this research, Quang Truong, Minh Nguyen, Hy Dang, and Bo Mei handled different dimensions that exist between house prices and influencing factors such as location, area, and population, when predicting housing prices away from the House Price Index (HPI) prediction to a more nuanced model. The critical nature this research takes on—apart from both traditional and advanced machine learning models—calls out for the much-needed subtlety concerning the complexities of the lesser-spoken models. Such an approach not only broadens the scope of the use of predictive analytics in real estate but also offers the comprehensive validation of various techniques within regression modeling. Their contribution is important because they were able to illustrate the varied impacts that feature selection can produce on prediction accuracy and also yielded optimistic results for the forecasts of housing prices. This well-researched method, underlying the Residential Valuation Intelligence System project, is in resonance with a wide array of machine learning models to be used, including Linear and Logistic Regression, Elastic Net, Kernel Ridge, Lasso, Random Forest, SVM, XGBoost, LGBM, and Gradient Boosting I was supposed to use. Cross-validation and model stacking resonate with the intensive validation and performance optimization techniques of Truong et al., thus providing an invaluable source in improving model predictive accuracy and efficiency for the valuation of residential property (Quang Truong, Minh Nguyen, Hy Dang, & Bo Mei, 2018).

Rahal's research takes a deep dive into predicting where the housing market is headed by looking closely at six different sets of data, each with lots of detailed information. What makes this study stand out is how carefully Rahal compares different types of models. These models vary by the kind of information they consider, how they look back at past data, and whether they use traditional or more modern approaches. Rahal also looks at the best ways to choose between these models and how to use a mix of them to get the best predictions.

A key part of the study is testing these models in what's called a "pseudo-real-time" setting, which is like simulating making predictions for the future. Here, Rahal finds that a specific method called approximate Bayesian Model Averaging (BMA) does a really good job at making predictions, doing better than simpler, more straightforward methods in many cases. But, there's also a finding that, on average, the simpler methods might have a lower error in their predictions, pointing out that the more complex models might not always be the best choice because they can sometimes make bigger mistakes, especially when the data they're trying to predict is all over the place.

This research is super important because it takes a close look at how to make good predictions in the housing market, showing the pros and cons of different methods. This is especially interesting for my project, the Residential Valuation Intelligence System, where I'm also using a bunch of different machine learning methods. I check how well they're doing with something called cross-validation and then combine the best ones to improve accuracy, which is pretty similar to some of the ideas Rahal depicts in the study (Rahal, 2015).

Before working with the dataset, the following shall be the method of study: preprocessing a large dataset to ensure the quality of the data and running regression analysis afterward to find key factors affecting the price of a house. The uniqueness of these methods is that they allow due considerations in the predictive model to external factors, fully recognizing their decisive, if not crucial, impact on the real estate price. The center, in this regard, specifically on how each of the algorithms is doing concerning the common-place problems that are, apparently, bias and variance always faced in predictive models.

In this research, the objectives of a project the Residential Valuation Intelligence System, in which I am involved, are supported: one of them also applies to the area of extensive sets of ML models, including Linear and Logistic Regression, Elastic Net, Kernel Ridge, Lasso, Random Forest, SVM, XGBoost, LGBM, and Gradient Boosting. The applied methodology of the project uses cross-validation and model stacking for optimization to bring out the prowess of machine learning in the real estate domain. Their findings, especially on how the outside factors influence house price changes, provide me some valuable insights that could improve the ability of my system to predict (Reddy, Padmalatha, & Devi, 2023).

The necessity for the forecast model is substantiated in the critical area of providing decision support to real estate consumers for reliable prediction of house prices. On the paper "House Price Prediction using Machine Learning Algorithm" by Shailendra Sharma, Deepti Arora, Gori Shankar, Priyanka Sharma, and Vihaan Motwani (2023), it is evident that the model implementing forecasts needs to be substantiated in the critical area of proving decision support for consumers. They compare the implementation of many different machine learning algorithms: Linear Regression, Gradient Boosting Regressor, Histogram Gradient Boosting Regressor, and Random Forest Regressor to find the best way to predict house prices. The approach of the work varies from data preprocessing for good quality data to the use of regression analysis for finding the effect of each of the factors in the determinants of house prices. From the tested algorithms, the Gradient Boosting Regressor and Histogram Gradient Boosting Regressor worked most precisely, which may indicate the high potential for applying advanced machine learning techniques in the area of real estate valuation. This goes together with my project on the Residential Valuation Intelligence System. It also applies an extensive range of machine learning models, cross-validated with the best stacking-optimized models. One of the critical findings that will enhance further the predictive accuracies is a focus on the recommendation of the Gradient Boosting technique, particularly for their effectiveness. This will go a long way in showing the benefits of the usage of diverse machine learning approaches in the valuation systems of residence. (Sharma, Arora, Shankar, Sharma, & Motwani, 2023).

The research articles reviewed provide a lot of knowledge and insights that would be highly relevant for use in your project on the Residential Valuation Intelligence System. So, each study referred to gives a slightly different view on the use of machine learning models in price prediction for housing. It kind of makes sense for them to be somewhat diverse when you use diverse models, including Linear and Logistic Regression, Elastic Net, Kernel Ridge, Lasso, Random Forest, SVM, XGBoost, LGBM, and Gradient Boosting. Moreover, your approach toward the evaluation of models using cross-validation and stacking of the best models to improve the predictions seemed like another approach that was being discussed in these papers. For example, this closely follows your approach of stacking ensemble techniques to get improved accuracy. This equally provides high efficiency of XGBoost, as reported in the case of Karachi city of Pakistan, and thus enforces one's choice of XGBoost and LGBM in its project from such proven high-performance points for your project on complex datasets with the highest accuracy.

Furthermore, focusing on the evaluation of different models and integrating a variety of machine learning techniques in the same fashion one observes throughout these studies is just an affirmation of the comprehensive approach you have taken with many algorithms. The success of such models in various geographical and data contexts highlights a relevant potential for applicability and effectiveness of the chosen models by you in the Residential Valuation Intelligence System. Thus, the approaches, results, and findings summarized herein from research papers not only further reinforce the relevance and potential of the selected machine learning models applied to this project but highlight the importance of model evaluation and stacking for improved prediction accuracy. This will place the project on a very firm footing, leveraging the best practices that were identified from challenges and successes in the above studies and poised to make a big contribution to the field of residential valuation using machine learning.

**2. Data & Project Management Plan**

**2.1 Data Management Plan**

The project Residential Valuation Intelligence System (RVIS) gives developed methodologies that could support streamlined collection, management, storage, and utilization of the data for the development of a sophisticated model of real estate valuation. The approach will be anchored on firm data collection approaches, management methods, storage formats, and mechanisms of usage that will ensure high accuracy and reliability in the way real estate valuations are done.   
***Data Collection approaches***   
 The RVIS project is planned as the first one to use the JSON format while collecting data from web scraping and API integration to collect comprehensive data from Zillow.com, a market-leading real estate service. Dual compliance ensures that richness is full and covers as many aspects of the dataset as property attributes, geographical information, and market trends. The project is starting with a focus on California but will then widen the data collected to other leading states in the USA. This is, however, in line with the target of getting a broad and diverse dataset to cover subtleties accounting for the national real estate market.

***Data management methods***  
 The collected data would undergo a stringent cleaning and pre-processing regime to ensure quality, consistency, imputation for missing values, and even outliers. In many cases, this is a critical process in preparing data for analysis, which includes outliers and missing value format standardization. This is now done with an script, written in Python, that allows this process to be done much faster and with many more states as the project scales.

***Data storage methods***  
 Cleaned and processed data are stored in CSV format due to its simplicity and flexibility, and it can be read by most data analysis tools. These files store big data sets, in a simplified form, without the inclusion of complex database systems, so that members can interchange, access, and cooperate in sharing data.   
***Data Usage Mechanisms***   
 For model building and development purposes, the RVIS project proposes to use state-of-the-art machine learning algorithms to analyze the data. To test the effectiveness and reliability of these models in value prediction, the measures subjected include a series of measures: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared, among others. Those selected models are built into the interface, giving actionable insights to the stakeholders and giving data-driven valuations.   
 This Data Management Plan provides and encapsulates an integrated approach within the RVIS project to harness data toward the advancement in real estate valuation. The project offers a revolutionizing tool for the stakeholders in a highly dynamic real estate market; it sets standards for the accuracy and efficiency of property valuation by comprehensively analyzing data and applying innovative analytical techniques.

***Data Collection Methods***  
 **Web Scraping**. Example: The project utilizes web scraping to extract detailed property listings and attributes from Zillow.com. This will involve the development of Python scripts that automate the process of visiting the real estate pages, extracting the needed information, like size, number of bedrooms and bathrooms, location details, and listing prices of the property; thereafter, they are to be saved and used for analysis.

**APIs*.*** Example: RVIS uses RapidAPI to access the Zillow full database via its API. This would be done by shooting requests to the API endpoints to get data about property listings, history of sales, and geographic information data from Zillow. To avail of the project with well-current and -structured data directly from Zillow, the project will have what it needs in its valuation models.   
 **Surveys**. Hypothetical Example: For a time, if there will be a need for the foregoing data coming from the requirements stipulated in the project, more surveys can be executed to gather more data, especially if there is qualitative data to be collected on homeowner satisfaction, neighborhood facilities, and subjective property characteristics. For example, developing online questionnaires directed at real estate agents and homeowners is such a way of being able to acquire some of the factors of property valuations that are not easy to quantify and may include the quality of local schools, community safety, and neighborhood appeal.

***Data Storage Formats***  
 **SQL Databases**. In this project, the details of real estate could be stored in the SQL database, where each table could represent properties, sales transactions, user queries, or market trends. The format also lends itself to the data being effectively queried and analyzed, such as looking for all properties within a given price range in a given neighborhood, to track price trends over time. The relational nature of SQL databases supports the project's need for complex data analysis and model training.   
 **CSV Files***.* In the case of an RVIS application, raw data that is being scraped from the internet or got from APIs is stored in CSV files. This form of storage is more comfortable and has wider compatibility with most software and applications for the storage of initial data, especially raw and unprocessed data. Quick sharing of data and collaborating among team members with ease to import data for analysis to the analytical tool for preliminary analysis, data cleaning, and preprocessing is done before more advanced analysis in an SQL database or other data analysis tools.

Data preprocessing is one of the major parts of the whole project of the Residential Valuation Intelligence System (RVIS) and is very essential to get the data ready for analysis and model building. These will be a series of steps in cleaning and preparing the dataset to not hamper the accuracy and efficiency of the predictive models. Key preprocessing needs for the RVIS project will include the following:   
***Data Preprocessing Methods***

**Data Cleaning**.The datasets from real estate have a problem with missing values. Sometimes the property information about size, number of bedrooms, and such entries in the listings do not exist or are not available. Imputation is handled in the RVIS project using the median or mean value of the respective feature or prediction models that estimate the missing values using other available data.

One step is removing outliers. This is because their presence would compromise the results either in data analysis or in predictive modeling and, therefore, lower the reliability in question valuation. Any outliers present in the data are removed using statistical methods such as Z-scores or IQR (Interquartile Range) for the dataset to represent the market under normal conditions.   
Next step isstandardizing data formats. Consistency with the various formats that include data, dates, and notation for addresses, among others, is important for drawing out an analysis that may not bring out success due to discrepancies and errors of interpretation.

**Data Labelling**.First, we are going to do categorization of types of property**.** The ranges would be anything from detached homes to condos and townhouses. The labelling type of the property is of utmost importance because, with such a type, it has a different model value. Properties would thus be categorized according to their characteristics so that targeted analysis of the property is made easy.   
Then we are going to label geographical tagging. This is one of the main areas that affect where a property is located. Therefore, one step in the RVIS project tags every property with geographical labels like neighbourhood, city, and state, which allows one to make location-based analyses and develop models that account for geographical price variations.  
We are going to do feature engineering.It draws on raw data collected and derives further features likely to influence the value of a property. For example, it calculates a property's age, price per square foot, and distance to amenities. Such derived features give a lot more insight into property valuation than just the raw data itself.  
Then we are going to do encoding categorical variables. Most of the attributes related to real estate properties come in categorical form, such as if there is a garage or not, if a pool is present or not, etc. The project uses various encoding techniques, such as one-hot encoding or ordinal encoding, to change such categorical variables into a suitable numerical form for feeding machine learning models. At lastrescaling features.Some features take very different scales, like property size and number of bedrooms. This may compromise some machine-learning algorithms. RVIS makes use of rescaling techniques, such as Min-Max scaling or standardization (Z-score normalization), to make all the features more comparable.

**2.2 Project Development Methodology**

The methodology that would apply for the project "Residential Valuation Intelligence System" is the Cross Industry Standard Process for Data Mining (CRISP-DM). The process is time-pro driven and ensures the delivery of success for a data mining project with a well-structured process. Initially, the project can be segmented into six primary stages, relevant tasks and smaller tasks will be assigned to each stage. The specifics of every stage are elaborated in the following sections.

***Business Understanding***

The foundational step involves a thorough comprehension of the business's goals, needs, and the constraints it faces. Valuing properties accurately is one of the very essential demands of the real estate world, impacting buyers, sellers, investors, financial institutions, and even insurance and governmental bodies. However, in some instances, gross misjudgments in property appraisals can at times lead to some forms of financial detriments and inefficiencies in the market if not potential poor decision-making.

At this stage, it lays a lot of emphasis on speaking to the relevant parties in detail, carrying out massive research about the market, and doing much analysis on the domain to minimize these risks. The approach of this kind is aimed at the revealing exact demands of the business and thus giving an opportunity for a project to operate effectively in a complex field. The assimilation of such information allows a clearer definition to be made of its scope and possible objectives to be outlined in clear terms.

Such a preparation before ensures that the project in concern is aligned to meet real-world industry needs; therefore, one should be prepared that the subject would equip them well to face the forthcoming challenges with an informed strategy. This will thus be laying the groundwork for a project that is solid from the technical, deeply interfaced point of view with the business context to assure that results are actually practical and have an impact for all the involved stakeholders.

***Data Understanding***

It is at this important juncture that the focus shifts to seeking and, in detail, scrutinizing the sources of data that are of prime importance to the successfulness of the project. This would cover the information collection that is as wide as possible, such that all details needed for the strong foundation of property valuation models are brought out.

Primary data is sought to be exhaustive and precise, encompassing residential property listings. This goes quite in detail, even to the precise location. It gives you the exact location up to the address, ZIP, and neighborhood, the physical attributes of the property (including size, total bedrooms and bathrooms, type of property, age of construction, lot size), the available amenities, and a record of its historical sale prices. Such detailed primary data is indispensable for creating accurate and reliable property valuation models. Each such factor likely has a serious bearing on the property values, so these must be considered in the valuation models.

All the data have undergone quality assurance processes with a strict quality, completeness, and relevance criterion for the conducted analysis. This will ensure that the data which is collected indeed meets the high standard which has been put in place for the project, and it will also be directly applicable and beneficial towards the development of accurate and effective models for property valuation. Such a comprehensive approach to data collection and validation is aiming at equipping the project with a solid and comprehensive dataset that will allow one to develop valuation models which, after realization, will stand up to scrutiny.

***Data Preparation***

Once a good grip over the data is ensured, detailed data cleaning and preprocessing tasks will be done. Data cleaning was initiated: the handling of missing values, removing duplicates, checking for the consistency of data, and ensuring there were no data entry errors present. Outliers and extreme values will be identified and treated with domain knowledge and statistical analysis during the analysis. This project employs feature engineering techniques such as one-hot encoding, numerical scaling, and the generation of derived features to extract features relevant from data that may influence the property values. Advanced methods such as natural language processing might be used to enable information to be extracted from unstructured data, such as property descriptions. The prepared data set should then be partitioned into training, validation, testing subsets of data that will help during model development, tuning, and the overall model evaluation exercise.

***Modeling***

In the step of modeling, we design and check a great number of models based on machine learning so that out of all these models, one can be identified from which accurate prediction of the value of residential property can be derived. We will benchmark techniques using Linear Regression, Elastic Net, Kernel Ridge, and Random Forest, XGBoost, and Gradient Boosting. The final cross-validation is done, including k-fold, during the model hyperparameter fine-tuning, ensuring that the chosen model is robust. Other ensemble methods, such as stacking, blending, and boosting, can also further be added to combine the power of different models in unison to improve the overall predictive power.

For interpretability techniques, feature importance and partial dependence plots have been used for the valuation process to arrive at a better understanding of the most influential factors in the modeling.

***Evaluation***

It shall include, but not be limited to, rigorous evaluation of the models with appropriate metrics for Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared. For purposes of comparing the performance of alternative models, given distinct geographic regions, property types, and price ranges, these several scenarios and subsets of data will be used. Special emphasis, while comparing models, will be on other aspects, such as model complexity, interpretability, and computational efficiency, apart from predictive accuracy.

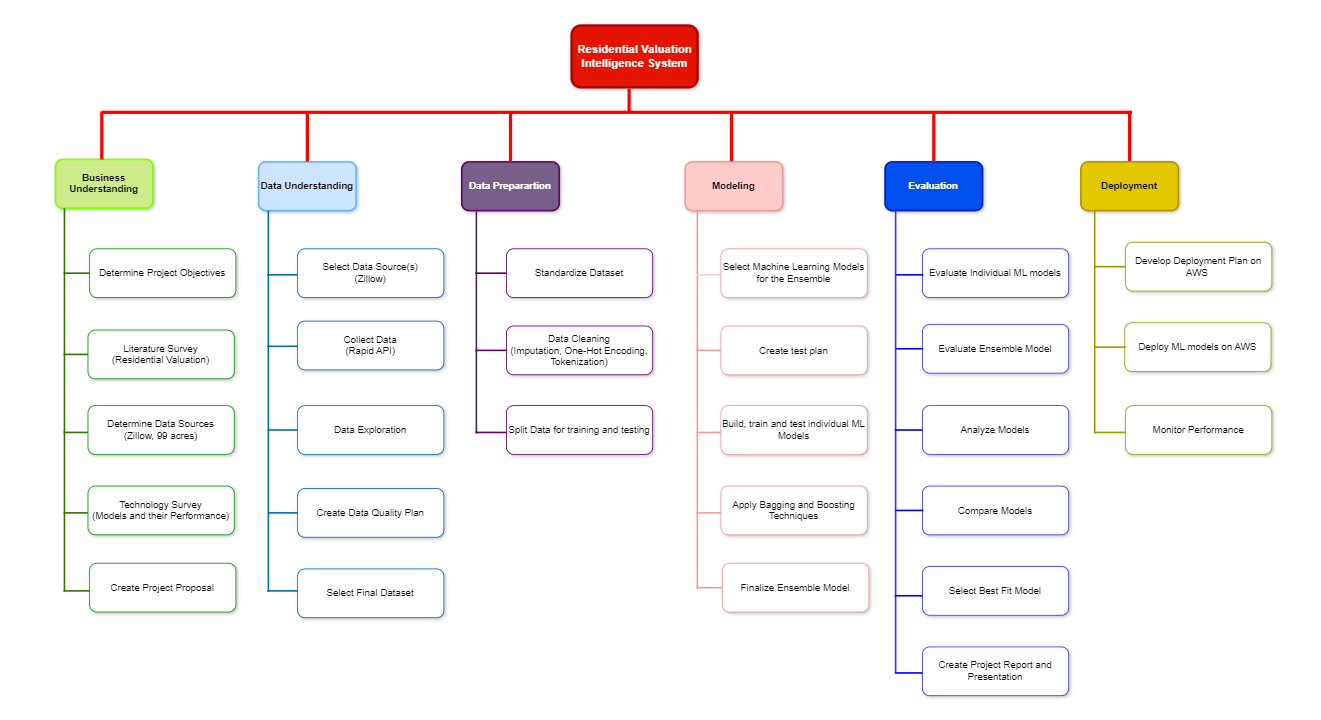
After all, best-performing model or models must be selected for deployment to realize the project business objectives.

***Deployment***

Finally, the deployed model(s) will be translated to an application or user interface that is user-friendly and open for access by stakeholders, including real estate agents, lenders, and consumers. This will provide an interface in which users are going to input details of the property and be in return to get an estimated accurate valuation that explains even factors constituting the same. In other words, the monitoring and maintenance of these deployed model(s) bound to be ensuring continued performance that adapts to drifts in the real estate market, the regulatory environment, and the changeable landscape of the data. The adoption and use of the Residential Valuation Intelligence System will be painless, supported by detailed documentation, user manuals, and training materials available to them. It will collect information and, therefore, it will need tuning to a constant feedback loop for the inputs of the users about the likelihood of areas that need improvement and ongoing fine-tuning of the system. The way of working we would adopt in these projects would be of an iterative and agile nature, with frequent checkpoints and reviews of the work together with the principal stakeholders. The project team is going to work very closely with subject matter experts, data scientists, and software engineers in terms of assurance that due coverage gets extended both to the entire width and depth of business requirements and technical feasibility. In that sense, risk management processes, change control, and quality assurance are implemented to mitigate such possible risks as part of the valuable delivery of the Residential Valuation Intelligence System.

**Figure 1**

*Work Breakdown Structure*



**2.3. Project Organization Plan**

***Work Breakdown Structure (WBS)***

The Work Breakdown Structure is divided into six major tasks which correspond to the six stages of CRISP-DM. Following this order, each stage will include tasks and deliverables which apply to the corresponding CRISP-DM phase. It provides us a manner in which we can manage all our resources and time effectively by breaking down all the requirements of the project to manageable units. Here below is the WBS of our project, as indicated in the Figure 1 (Work Breakdown Structure).

First step is Business Understanding. In this, we shall define the objective and scope of our project one by one and then refer to the literature survey so that we can have an overview of the related works. Further, we need to search for the source of data such as 99acres, Zillow and conduct a technical survey in order to get inputs on which machine learning algorithms, software, tools, and relevant skills are used and implemented for the research-related problems at hand. Finally, the details of the background, project, objective, deliverables, and goals will have to be prepared considering the chronology of the work to be written in the project proposal.

Second step is Data Understanding, where we would do five tasks. Finalize the data source (Zillow), collect the dataset (web scraping using Rapid API), and take a look at the different kinds of data collected. We conduct research on the dataset regarding the appearance, size of the dataset, how many rows, features it possesses, and more importantly, figure out any issues regarding the dataset that are supposed to be written in the data quality plan. In the final task, we will finalize the dataset we will use for this project.

Third step is Data Preparation. Preparation of the data step that merges the dataset and further does two processes that are data cleaning such as Imputation, One-Hot Encoding and Standardizing to help us handle issues in the dataset such as duplicate, missing values, outlier, etc., for getting a best-fit dataset for our machine learning model in the further step. Finally, the dataset will be split into training and testing.

Fourth step is modeling. After preparing the dataset, we will proceed with the finalization of machine learning models that can be used with the data. After which, each member will develop and train the model with the prepared data. We also applied some boosting techniques to enhance the performance of the models.

Fifth step is Evaluation where all the models are built and trained and in fact, the evaluation is to be done in order to analyse the results of those models. Later on, the results are compared, and with the help of these comparisons, it is seen that for the project which model is best. We are then going to prepare the project report, and thereafter, we prepare the presentation.

**2.4. Project Resource Requirements and Plan**

***Hardware and Software Requirements***

The project is implemented and deployed using AWS services, Python Jupyter Notebook. A local machine must have at the very minimum of 8GB RAM or more, a 64-bit windows processor, and at least a 1GB graphic card to properly run AWS cloud services.

**Table 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Utility** | **Resource Type** | **Tool/Application** | **Duration** | **Estimated Cost** |
| Data Storage | Hardware | CSV | 3 months | $0 |
| Local Machine | Hardware | 64-bit version machine | 3 months | $650 |
| Cloud Service | Software | AWS | 3 months | Free |
| ML Modeling & Evaluation | Software | Jupyter Notebook | 3 months | Free |
| Machine Learning Frameworks | Software | Scikit-learn, Pandas, XGBoost, LightGBM | 3 months | Free |
| Data Pre-Processing | Software | Python Jupyter Notebook | 3 months | Free |
| Web Scraping | Software | Rapid API | 1 month | Free |

*Resources and Cost Estimation*

As from Table 1 (Resources and Cost Estimation) the data set, along with all other files related to the project, is stored in the csv file. Thus, all preprocessing and data cleaning were carried out in this project. The statistical analysis is done with Python Jupyter Notebook. Machine learning algorithms to build, train, test and deploy the models are implemented using Python libraries in Jupyter Notebook. This helps at starting from labeling and preparation of your data to the selection of an algorithm, training, tuning, and optimization of your model deployment for predicting outcomes and implementing solutions in a production-like environment. The Machine Learning Frameworks used to carry out complex modeling implementations and other requirements. In order to achieve an end-to-end solution of the project, which provides a single integrated platform for interaction and access to various services used in the project.

**2.5 Project Schedule**

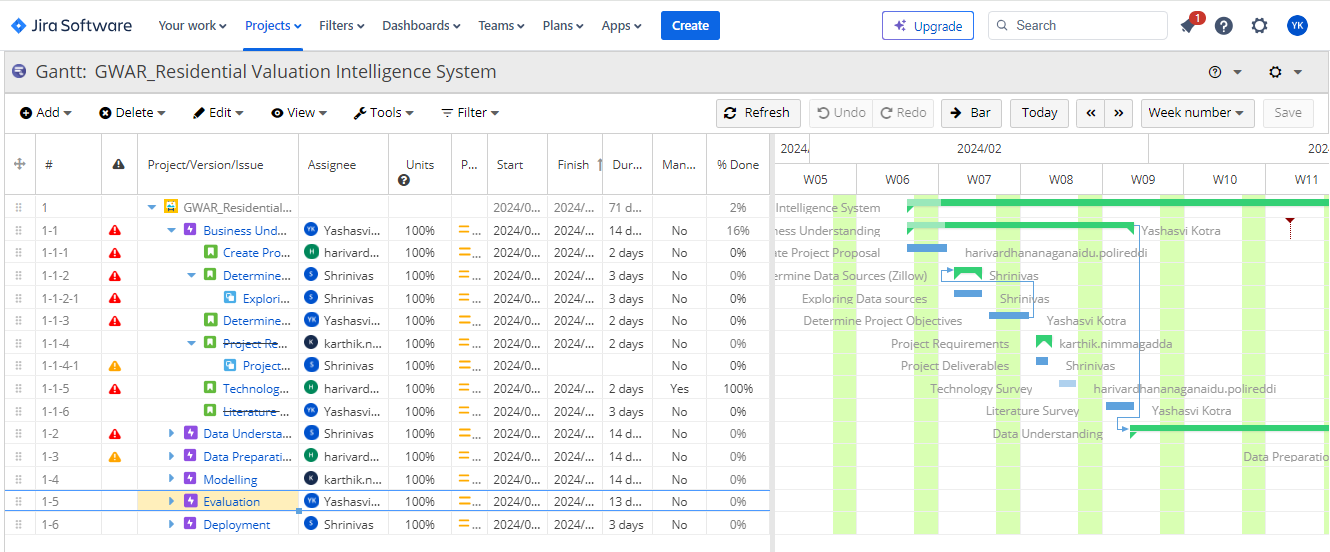
***Gantt Chart***

The Gantt chart has found great use in charting the course of our tasks against a defined timeline in orchestrating the project of the Residential Valuation Intelligence System. This chart type was developed in the 1910s; it was indispensable for the observation of the life cycle of the project and represented tasks along the view of the calendar to underline both the sequence and interrelation. This is especially best in exposing overlaps and dependencies, thus provides the exact identification of the project's critical path. The chart design also caters not only to schedules but also to strategic resource allocation, ensuring that tasks are put in place and assigned to the appropriate team expertise. The development of the Gantt chart within the project from the very inception of conceptualization and market analysis to the development and testing stages, evolving with far more detail than originally conceived gave a perfectly clear visual cue about the deviations which would accrue. This kind of support has been essential, enabling the project to be different and flexible concerning the changing needs and requirements of the project, in order to drive and quicken any emerging issues. Gantt chart, therefore, gives direction or acts as an orienting compass, and at the same time, it is a ledger to make sure that each of the stages and steps is properly navigated with effective documentation toward the final goal or completion of the project successfully.

This is the reason these one week and two weeks' sprints, therefore, are part of our GWAR Residential Valuation Intelligence System within the project management plan during this phase of understanding the business. The distribution of work between these sprints was done very diligently each of them had an even flow of work for every team member. This strategic allocation allows for an equitable contribution where the Gantt chart serves as the real-time books of account for task status updates and contribution of work. This is, however, supposed to be updated by the team members, respective of the progress of the subtasks and sections which have been provided. The former approach not only allows us to work together in a very collaborative approach but also makes us delve deep into scrutiny of the members' scope of work, effective time management, and quick liaise with backlogs to develop a good base for overseeing the transition of the project in great details.

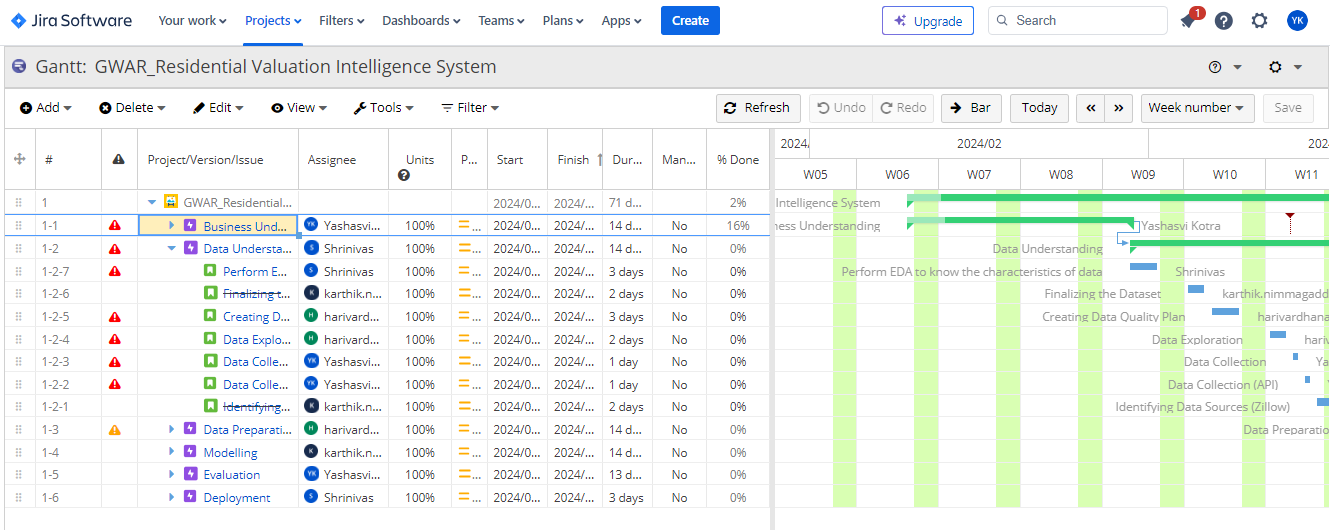
**Figure 2**

*Gantt Chart for Business Understanding*



Gantt chart for business understanding phase that explains dependencies, assignment of the tasks, start and finish dates of each task by the assignees (as mentioned in Figure 2).

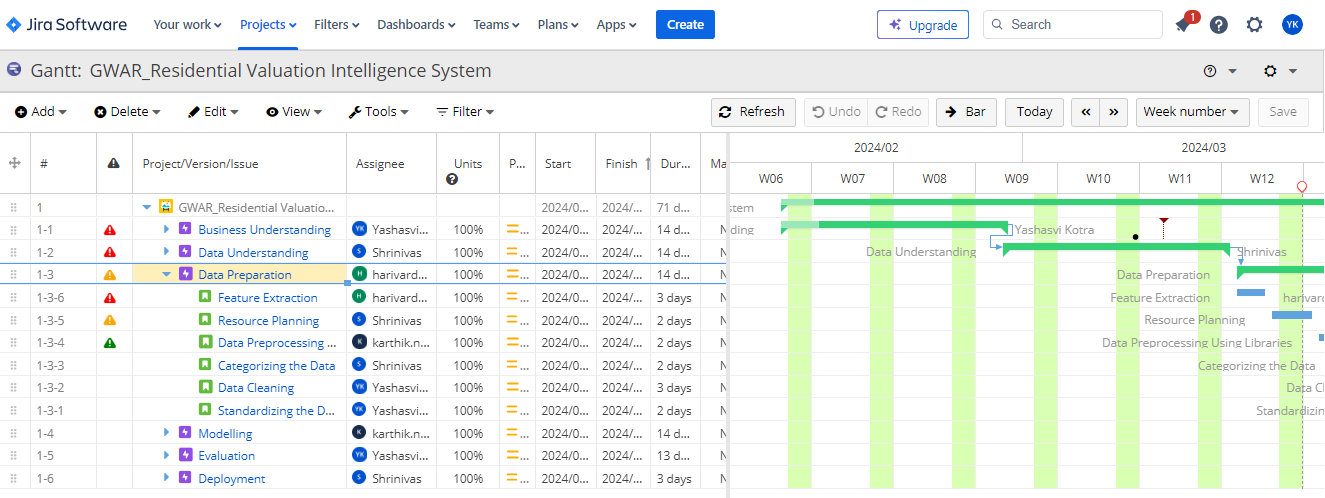
**Figure 3**  
*Gantt Chart for Data Understanding*



Gantt chart for data understanding phase depicts the assignee, priority of the task, start and the expected finish dates of the task and also the percentage of completion of each task according to the updation’s made (as mentioned in figure 3).

**Figure 4**

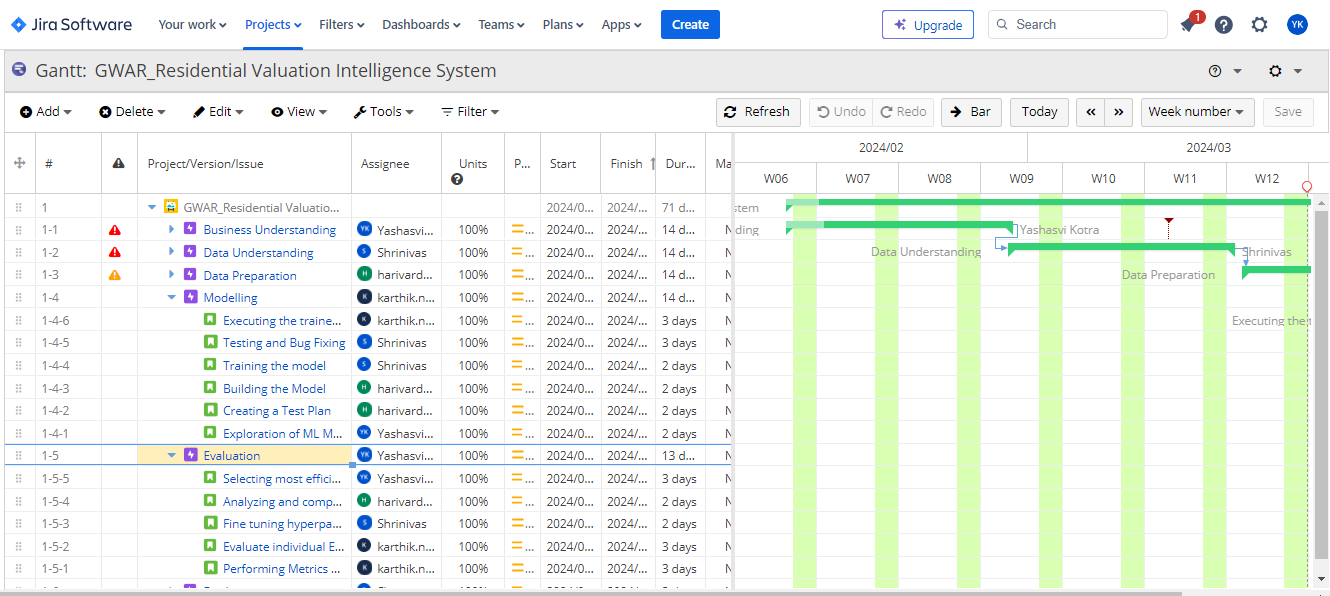
*Gantt Chart for Data Preparation*



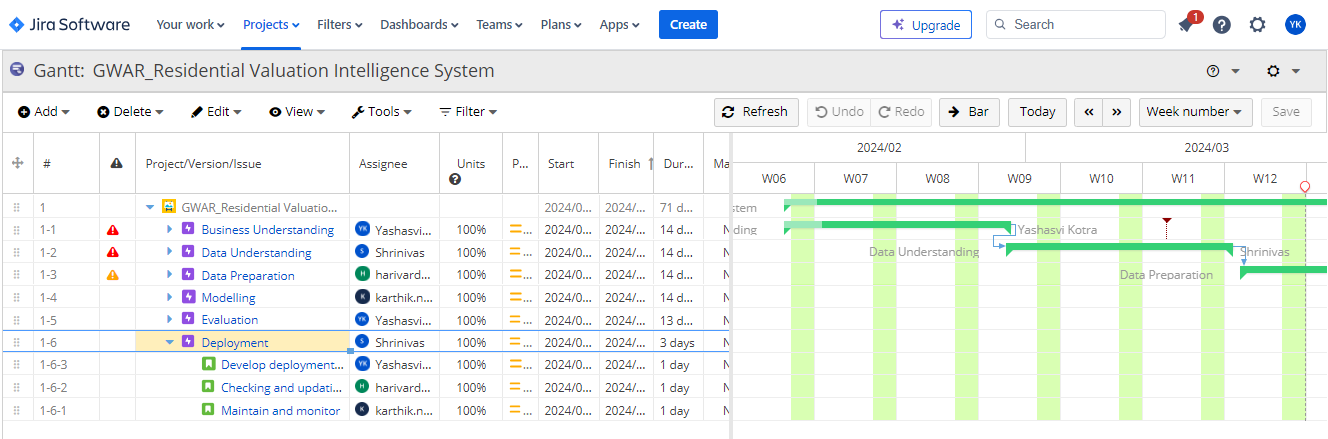
From Figure 4, Gantt chart of data preparation phase shows the assignees of the tasks, number of days took to accomplish the task, percentage of completion of each task. This phase happens to have the longest sprint among all the other sprints for the phases.

**Figure 5**

*Gantt Chart for Modelling and Evaluation*



**Figure 6**

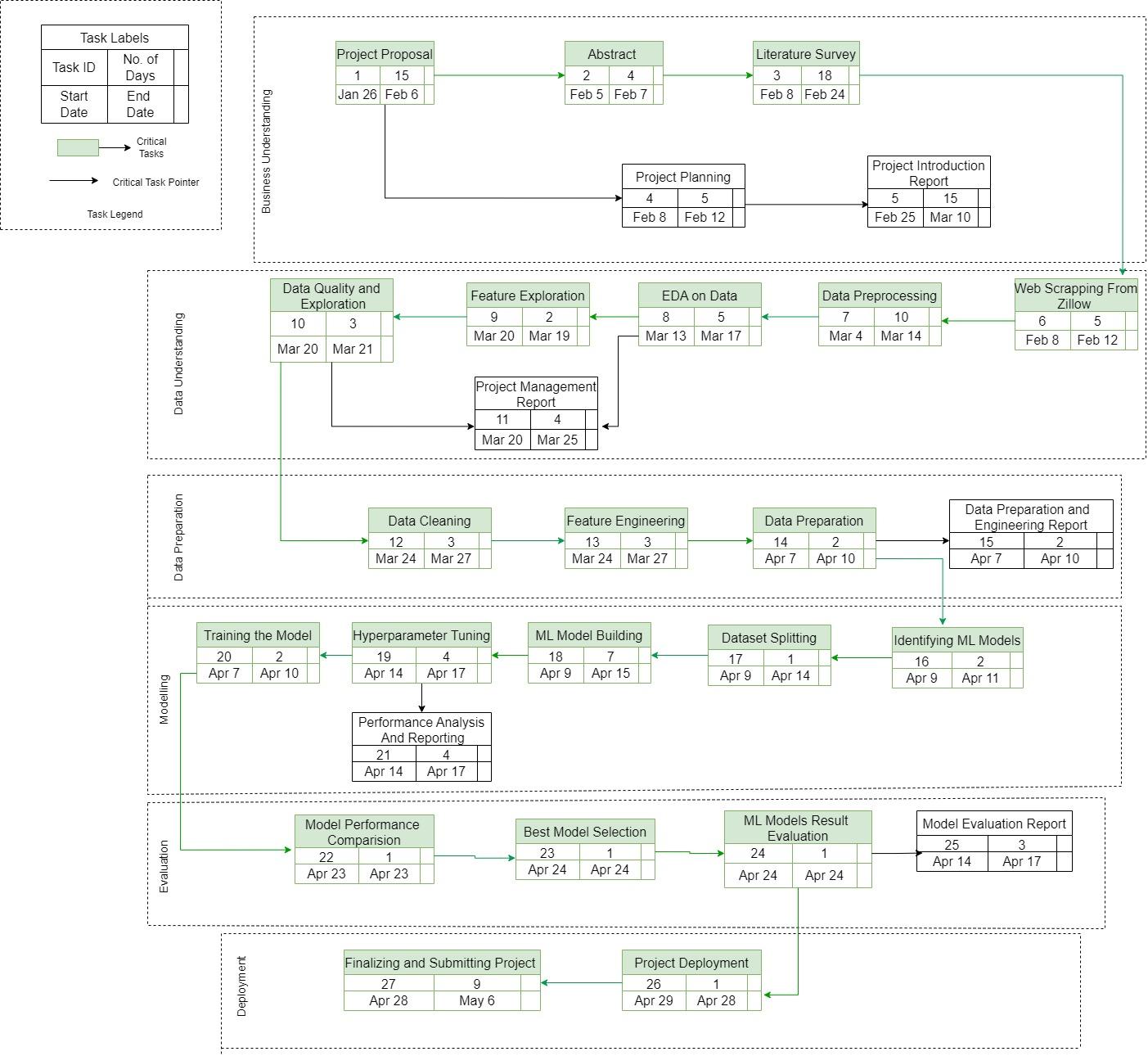
*Gantt Chart for Deployment*

***PERT Chart*** We used the PERT (Program Evaluation and Review Technique) chart in managing the project of the Residential Valuation Intelligence System, since it is a schematic approach that defines task relationships and tasks along the critical path of the project. It was developed by the U.S. The PERT chart was derived from the U.S. Navy in the 1950s for the Polaris missile program; it has few elements of a schedule than a network model. This type of network model is very helpful; it is keen on the most important prerequisites within a considerably complicated project landscape down to the smallest details. It is, therefore, especially effective in the pinpointing and study of the critical path, i.e., the associated chain of linked activities that carry maximum impact regarding the timescale for the given project. In our project, this would mean investigation of each of those steps to the core, from data collection right up to system deployment, giving the probable estimation of time frames that would suit the project management detail. The holistic view of the PERT chart showed not only the potential delays in time but also the changes in resources, which were important to have the flexibility to adapt, enabling steering the development of our cutting-edge valuation system to successful, timely completion.

In Phase 1 of the project for Residential Valuation Intelligence System, conducted with utmost precision between January 26 and February 24, a detailed project proposal is developed, abstract crafted, and literature survey taken up. In the period starting January 26th to February 6th, the importance of coming up with the project proposal was important in pointing out the strategic blueprint within the project. On the other hand, the crafting of the abstract running from 5th to 7th February was essential in making a brief overview of the project intentions and objectives. The literature review was carried out between February 8 and 24. This enabled them in a much better-informed position that would be able to gauge the technology landscape and the market, hence, inform the direction and methodology of the team.

We can see from Figure 7 from March 9th to March 27th, while moving into the exploratory phase, we have actually dug quite deeper in the data quality and feature exploration efforts to assure about the data reliability and relevance for the underlying predictive models. Data quality assessment and exploration during this time of sensitive care were taken from March 9 to March 21, making sure that the data was looked into, with its integrity and utility at hand. From March 9th to the 12th, carried out the exploration of the features that form the backbone of strong data preprocessing done from the 4th to the 14th of March.

**Figure 7**

*****PERT Chart for Residential Valuation Intelligence System*

This was followed by further data enrichment through web scraping from Zillow, from February 8th to 12th, and from March 20th to 25th with comprehensive project management reporting, ensuring that the project progressed on papers and through oversight in tandem was well-documented.

The culmination of all technical development for the project was between April 7 and April 28, in which rigorous data cleaning, feature engineering, model training, and hyperparameter tuning activities were carried out. Most of the work in model training was done between the 7th and the 10th, then; a very important and complex task that would build a foundation for the hyperparameter tuning, which followed on from the 14th to the 17th of April. It was finely adjusted to the model for an optimum performance, and from the 14th to the 17th, analysis of the performance was done with utmost care. After evaluating the results and applying thorough scrutiny, the project was concluded with the final selection of one model as the best performer on 24th April. The deployment of the project followed from 26th April to 6th May, where the system was up and running. Such PPERT chart-guided journey, from conception to deployment, underscores the commitment of the project towards a very disciplined and methodical approach that assures the delivery of a robust and reliable Residential Valuation Intelligence System.

**3. Data Engineering**

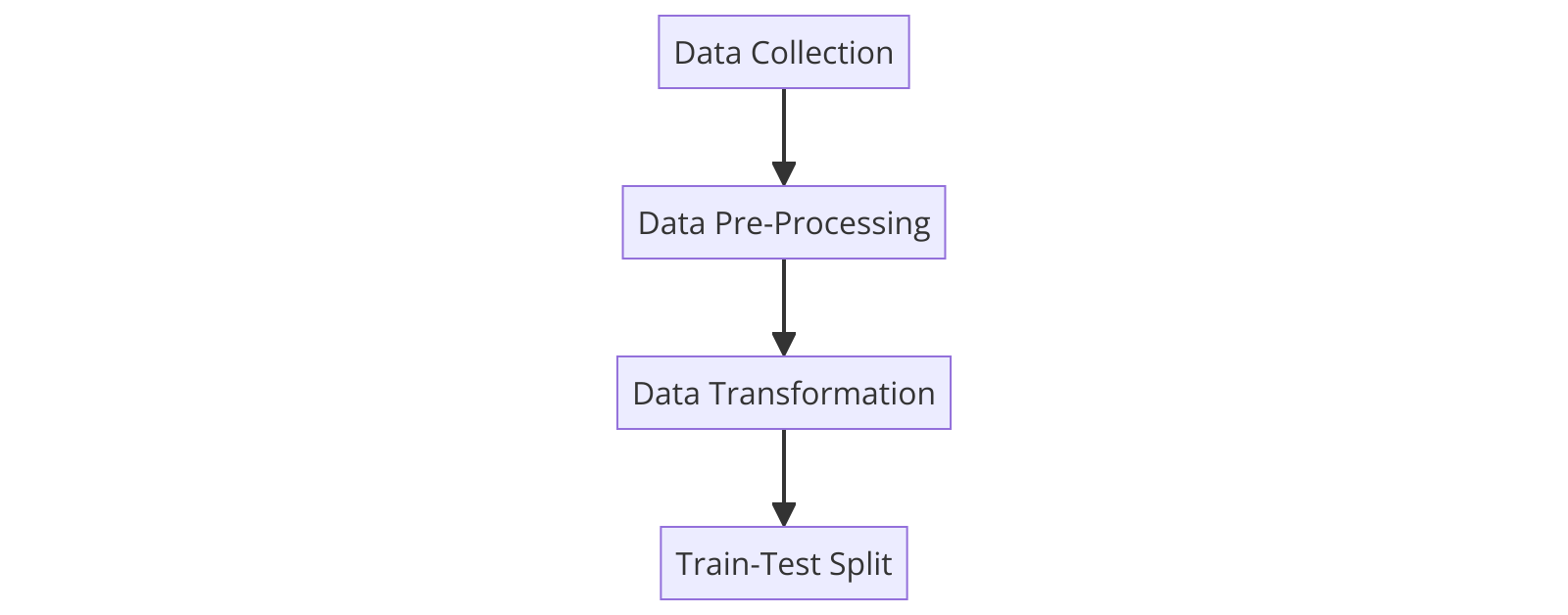
**3.1 Data Process**

In the project Residential Valuation Intelligence System aiming for house price detection, the data is collected using Rapid API to access Zillow's real estate listings. This method brings out key information of the property listing like prices, location, size, number of bedrooms and other vital features that influence home valuations. The dataset is assured to be always current and reflects the dynamics in the market, as Rapid API is being used for the extraction of data in real time. Given the sensitivity and commercial value of real estate data, our usage of the Zillow API through Rapid API adheres to the terms and conditions set forth by both Zillow and Rapid API. This ensures that the data used for predictive modeling is collected ethically and legally.

Once the data is retrieved from the API in a JSON format, it will go through a number of important pre-processing steps that are central to the cleaning of data, putting it into shape for further analysis and modelling. First and foremost, it involves converting JSON data into a manageable format of CSV, so as to have ease in the process of manipulation and evaluation. Further, in the cleaning stage, rigorous cleaning is performed on the data where it not only identifies and removes any null values but also corrects data entry errors, changes the data types and column names and also handles outliers that might affect the modeling result. Moreover, this will help standardize and make sure that the data types and the names of the columns are consistent throughout the dataset. The process is very important for the integration of data from different sources and easy merge of the data toward assisting a full analysis.

The Data Transformation part includes a few very crucial operations in which the raw data assumes the form of a regression analysis. We have implemented Feature engineering to generate new, potentially influential features that include price per square foot, age of property, and proximity to key amenities, which may significantly affect house prices. Alongside feature creation, normalization techniques are applied to scale numerical data appropriately. Then data is split into train, test, and validation datasets in the ratio of 80:20. These datasets are analysed to ensure that the data is unbiased. Finally, the resulting dataset is saved in CSV format to be used during the model development and evaluation phase.

**Figure 8**

*Data Process Diagram*

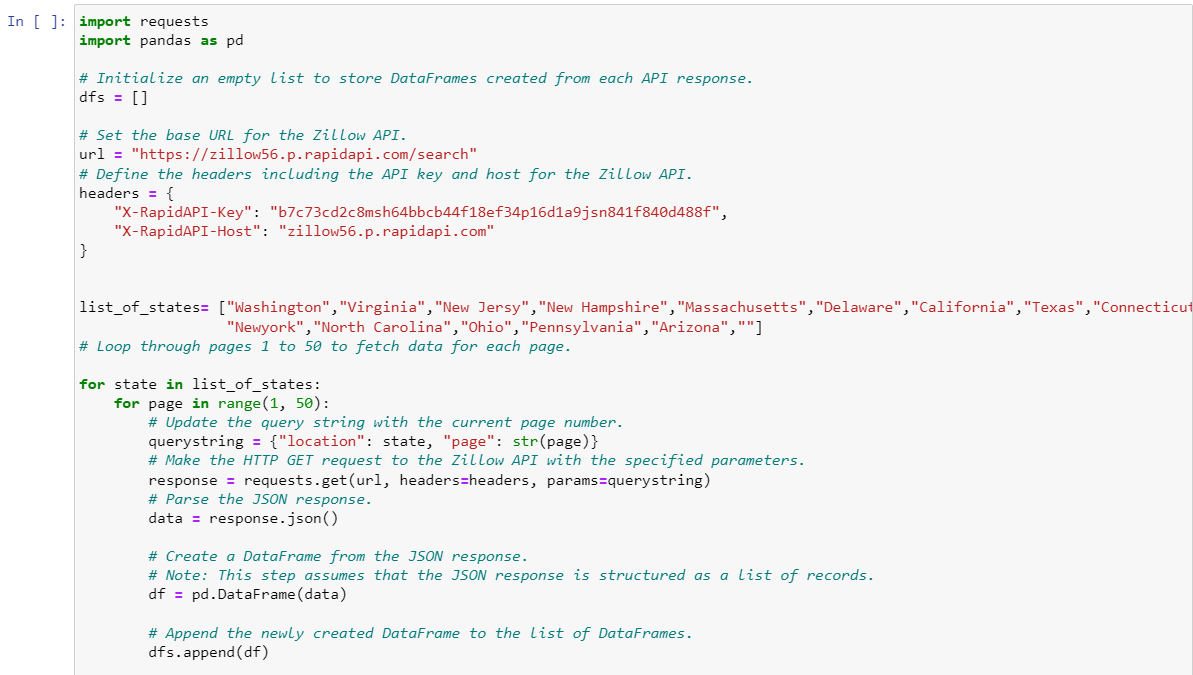
As mentioned in Figure 8, the data process involves four main steps. They are Data Collection followed by Data Pre-Processing, Data Transformation and at last Train Test Split.

**3.2 Data Collection**

The primary objective of our project is to develop a robust predictive model for housing prices. This model aims to utilize regression analysis to forecast prices based on various property characteristics and prevailing market conditions.

The data collected from Zillow will provide us with a multitude of property attributes that are crucial for determining house valuations, such as the number of bathrooms, home type, and square footage, as well as market trends indicated by variables like days on Zillow and price changes.

**Figure 9**

*Code snippet for scraping the data using Rapid API- Key*

From Figure 9, We gathered this data through a Python script interfaced with the Zillow API at Rapid API. For automated and systemized collection of the real estate data, the script employed the 'requests' and 'pandas' libraries.

First, the script called for an exhaustive sweep of diverse housing markets, which was specifically defined as a list of U.S states. For each of these states in the list, it programmatically queried the Zillow API with 50 pages of listings this represented vast records in its database.

This script, for each HTTP GET request, was made to include pertinent headers such as API key and host information to authenticate and direct the query to the correct endpoint. This header ensures that the requests conform to the parameter requirements set by the API for data retrieval.

**Figure 10**

*Dataset in JSON format*



From Figure 10, we can see that the raw dataset that is extracted from Zillow is in JSON format.

*Data Parameters and Quality Requirements*

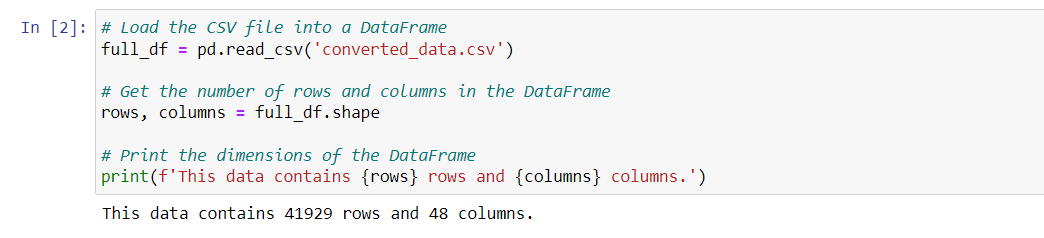
The data parameters we expect for effective prediction include:

1. Location: Zip code, neighbourhood, city, which can greatly influence house prices.
2. Size: Area in square feet or meters, number of bedrooms, bathrooms.
3. Age: Year built or age of the property.
4. Condition: Overall state of the property, any recent renovations or upgrades.
5. Features: Presence of a garage, garden etc.
6. Market trends: Current market conditions, historical price trends in the area.

The data quality requirements we expect for effective prediction include:

1. Accuracy: Data should accurately reflect real-world values, such as correct listing prices and should match the actual features of the home.
2. Completeness: All records should have values for key features and there should not be any missing values as they impact model performance.
3. Consistency: Data formats and values should be consistent across the dataset.
4. Timeliness: The dataset should include the most recent data possible to reflect the current market conditions.
5. Uniqueness: Each record should represent a unique property to avoid duplication, which can skew results.
6. Validity: Data should conform to logical constraints.

**Figure 11**

*Raw dataset shape after converting JSON format to CSV readable format.*

From Figure 11 we can see that the raw dataset has total 41929 rows, 48 columns in which many of them will be removed in cleaning process.

**Table 2**

*Data Collection Plan – Rapid API*

|  |  |
| --- | --- |
| **Description of Data Collection – Rapid API** |  |
| Purpose of Data Collection? | Data collection from Zillow via Rapid API provides historical property data. Collecting data allows for the extraction and analysis of relevant features like location, size, amenities that significantly influence property values, enabling more precise valuations. We can also access up-to-date data that helps in understanding market dynamics and trends, aiding investors, developers, and homeowners in making informed decisions based on current market conditions. |
| How will the Data help? | The data gathered will be utilized for training machine learning models to accurately predict house prices. |
| What should we do after collecting the data? | After collecting the data, it should be cleaned and preprocessed to ensure accuracy and relevance, removing outliers and handling missing values. And then the data can be analyzed to identify patterns and features that will inform the development of predictive models. |
| Raw Dataset Format | JSON format |
| Key Variables | Price, state, lotArea\_sqft, city, latitude and longitude, state, zipcode, streetAddress, homeType, bathrooms, bedrooms |
| Variable data types | Text and Numeric |
| Collection Method | Extracted data using web scraping from zillow |
| Data Collector | Shrinivas Karthik |
| Start Date | 03/28/2024 04/01/2024 |
| End date | 03/29/2024 04/02/2024 |
| Duration | 1 day 1 day |

**3.3 Data Pre-processing**

As we have a raw dataset extracted from Zillow it has different format, different types of outliers, the presence or absence of null values, the different number of columns and performs initial level cleaning on them. This helps us take the leverage of different types of data in separate files to normalize them so that they are equivalent before we merge them.

So, we changed the data format from JSON to csv format so that it will be simple and easy to understand the dataset as the data is stored in rows and columns.

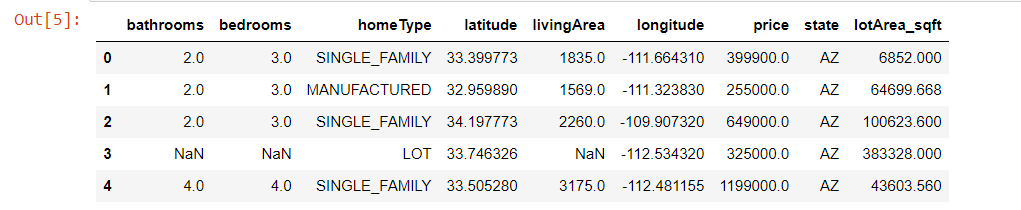
**Figure 12**

*Code snippet for converting JSON to readable csv format*

As from Figure 12, the dataset is converted from JSON to csv format, we have the uncleaned dataset and require some pre-processing (cleaning) steps to be performed and required for predictive modelling.

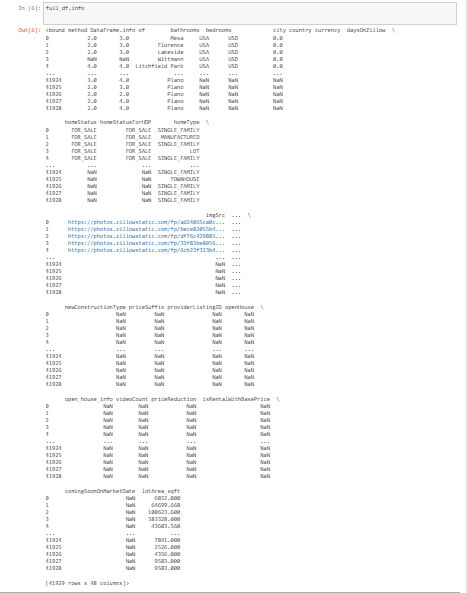
**Figure 13**

*First 5 rows of the dataset in csv format*



From Figure 13, We can see the first five rows of the dataset stored in a dataframe from readable csv format. This is an uncleaned dataset and we have to clean the dataset for better understanding.

**Figure 14**

*Dataset info before cleaning*

After scraping the data from Zillow and before cleaning the dataset, we have around 41929 rows and 48 columns (from figure 14).

For house price prediction, selecting relevant features is crucial because they directly influence the accuracy and effectiveness of the model. Here’s a breakdown of the columns you listed and their relevance to house price prediction.

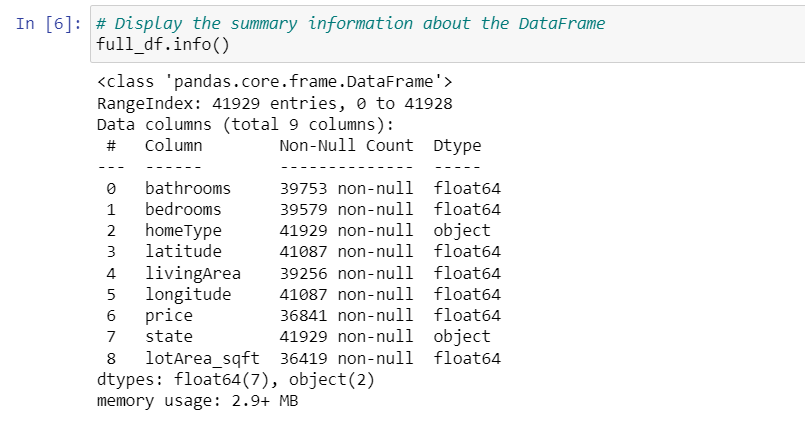
*Highly Relevant Columns*

* bathrooms: Number of bathrooms in the house.
* bedrooms: Number of bedrooms in the house.
* homeType: Type of home
* livingArea: Square footage of the living area.
* latitude and longitude: Geographical coordinates (2 different columns)
* lotArea\_sqft: Size of the lot which can impact price.
* price: Sale price of the house, often the target variable in prediction models.
* state: State where the house is located.

*Less Relevant Columns*

We have removed ‘Country’ and ‘Currency’ columns. These are typically constant within datasets for a specific country and hence provide little predictive power. homeStatusForHDP, imgSrc, isFeatured, isNonOwnerOccupied, isPreforeclosureAuction, isPremierBuilder, isShowcaseListing, isUnmappable, isZillowOwned. These factors are more about listing specifics and Zillow’s internal metrics, which might not be directly related to actual market value. listing\_sub\_type, newConstructionType, comingSoonOnMarketDate, group\_type. Can be relevant in niche contexts but generally less critical. providerListingID, isRentalWithBasePrice: These are administrative or specific to rentals, thus less relevant for selling price prediction.

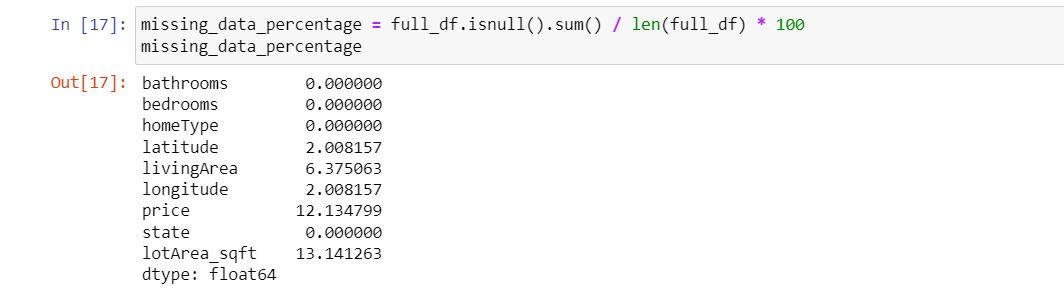
**Figure 15**

*Column datatypes after removing less relevant columns*

We have checked whether there were any different datatypes. If there are any datatypes that should be modified then we can change the datatype of the column. For us it is good to go for further pre-processing steps (from figure 15).

**Figure 16**

*Code for Checking Missing Values*



As like in Figure 16, We have checked for missing values in our dataset. We found missing values in some columns where ‘lotArea\_sqft’ column has highest percentage (13.141%) of missing values.

**Figure 17**

*sMissing Values after Imputation*



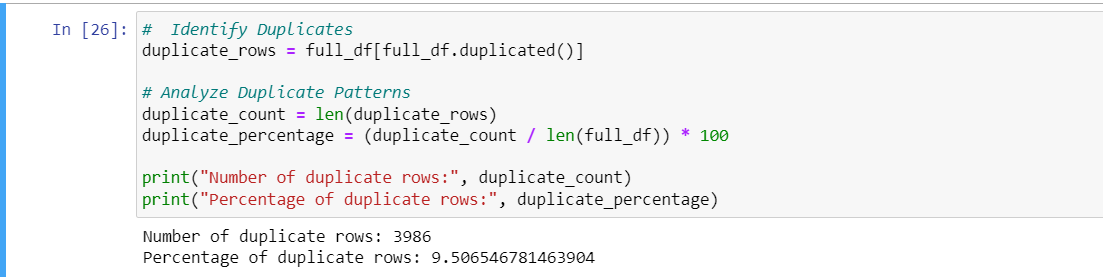
From the figure 17, we can see that there are no missing values. They are filled with substituted (median) values in the column. This method is called imputation. So, we have filled all the missing values in the dataset and ready to perform further cleaning steps.

By proceeding further into pre-processing steps, we have changed the ‘zipcode’ column to categorical variable. And also, we have added one more column ‘price\_per\_sqft’

by dividing the ‘price’ column with ‘living Area’ column.

**Figure 18**

*Code snippet for duplicate values*



The code prints the number of duplicate rows and the percentage of duplicates in the DataFrame. From the output, we see that there are 3986 duplicate rows, making up about 9.506% of the data.

**Figure 19**

*Code snippet for handling outliers*



We can see from Figure 19, that we have detected outliers and replaced them using median (IQR method).

**3.4 Data Transformation**

***Feature engineering***

Standardization of Land Area Measurements*.* This is especially important because we intend to do predictive modeling of house price, and the measurements have to be standardized in such a way that they correspond to one another and thus be comparable in data points. This is done through the function "convert\_to\_sqft," as shown in Figure 1. The function is designed to convert land area measurements from a variety of units into square feet (sqft), the standardized unit used in the model.

**Figure 20**

*Python code for the standardization of land area measurements to square feet*A screen shot of a computer code

Description automatically generated

As mentioned in Figure 20, This function will take two arguments ‘value’ and ‘unit’. The variable value is the numerical measure of the land area, while the variable unit issues the current unit of measure. It checks for missing values in the argument provided and, if so, returns ‘None’ to avoid errors in computation. For non-missing values, the function then executes the conversion of the measurements into 'acres' by the conversion factor that one acre is equal to 43,560 square feet. If the value is in square feet, then it is returned without changes. This function raises the error as a ValueError for unknown units, which forces integrity in the data.

We then apply the function ‘convert\_to\_sqft’ to the DataFrame ‘new\_df’ using the ‘apply’ method together with the lambda function. In this operation, the Data Frame gets a new column ‘lotArea\_sqft’ for standardizing the measurements of land area in square feet for effective analysis in the predictive modeling.

*Calculating the Price per Square Foot*

The creation of the new feature, price\_per\_sqft, helps improve the data since it gives a standardized value for the cost of the property in relation to its size. This feature is calculated by dividing the total price of a property by its living area in square feet. This code snippet (Figure 19) shows the implementation of the following calculation within the DataFrame ‘df\_selected’. By adding ‘price\_per\_sqft’, it thus allows for value addition in that property values can be compared across varied market segments and property sizes. This improvement is of utmost importance to the feature engineering phase that brings a more accurate and informative predictive modeling process. This helps improve the model's ability to discern the trends of property size in relation to the market value, an important observation in real estate estimation.

**Figure 21**

*Python code for calculating the feature `price\_per\_sqft` in the DataFrame `df\_selected`.*A screen shot of a computer code

Description automatically generated

Binary Encoding of Home Type Feature*.*  The 'homeType' attribute turned out to be a categorical variable with many different categories, like 'SINGLE\_FAMILY', 'MANUFACTURED', 'LOT', 'APARTMENT', 'TOWNHOUSE', 'MULTI\_FAMILY', and 'CONDO'. As seen in (Figure 20). So as to prepare this categorical data for fitting into the regression model, we encoded the given representation. While several methods could be used in encoding categorical variables, in this work, we use label encoding to minimize the inflated data dimensionality due to encoding while maintaining the information of the categorical variable.

**Figure 22**

*Home type column data categories*

A screenshot of a computer

Description automatically generated

For this, I used the binary encoding method from the Python category\_encoders library (see source code in Figure 22). I then used the BinaryEncoder class for instancing the column 'homeType' as a target for encoding. Then I have called the fit\_transform method for the encoding on the data selected from the column and converted each unique category to a binary number represented across several new columns.

Having successfully transformed the 'homeType' data into binary form, the newly encoded columns were concatenated into the original DataFrame without the original 'homeType' column. The resulting DataFrame, therefore, contained many new columns, one for each binary digit of the encoded variable "homeType". This binary encoding method allowed the incorporation of categorical information into the regression model without forcing the dimensionality too much, which is a usual side effect of one-hot encoding in the case of multiple categories.

**Figure 23**

*Python code for binary encoding of the 'homeType' feature using the ‘category\_encoders’*

A screen shot of a computer code

Description automatically generated

Transformation of the Housing Dataset.In the original dataset before feature engineering, there were quite a number of attributes that described various aspects on the same topic for house pricing, which included: number of bathrooms, bedrooms, type of home, geographic location, living area, and lot area (see Figure 22, top panel). After such transformation, calculated features were added to the dataset for better predictions of property values (see Figure 23, bottom panel). The new ‘price\_per\_sqft’ feature was computed as the price per square foot of the living area, thus making it easy for comparison and standard, even when sizes differ across all properties. Furthermore, the attribute 'homeType' was binary-encoded to replace the original categorical variable with binary variables ('homeType\_0', 'homeType\_1', 'homeType\_2') so that it could at least be made available in the regression models. This should result in better performance because it would provide normalized, quantitatively relevant data points to the predictive model that should reflect underlying trends in property valuation more effectively

**Figure 24**

*Before Feature Engineering*

A screenshot of a computer

Description automatically generated

**Figure 25**

*After Feature Engineering*

A screenshot of a computer

Description automatically generated

The DataFrame `df\_selected` before (top panel) and after (bottom panel) feature engineering transformation, illustrating the computation of `price\_per\_sqft` and the binary encoding of `homeType`.

Normalization of Housing Data Attributes.Preparing the dataset for regression analysis, normalization of the selected numerical attributes using MinMaxScaler from the scikit-learn preprocessing module was done. The code snippet (Figure 24) below exhibits the initialization and application of the scaler to the features bathrooms, bedrooms, latitude, longitude, livingArea, lotArea\_sqft,. These attributes have been rescaled into [0, 1] range in such a way that it shall contribute equally to the predictive ability of the model. This way, normalization avoids biasing potential variables with larger scales and allows algorithms that are sensitive to the scaling of features.

**Figure 26**

*Python code for Min Max Scaler Normalization*

A screenshot of a computer program

Description automatically generated

The DataFrame 'data' will be displayed after normalization, showing transformed values reflecting the standardized scale of each feature (see figure 26). This is a very crucial step in the data preprocessing phase since it directly affects the performance and accuracy of the predictive modeling.

**Figure 27**

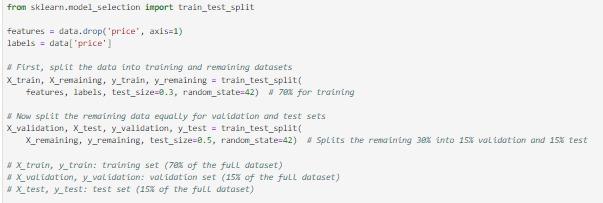
*Output of the DataFrame 'data' post-normalization, displaying scaled values for selected numerical features*.

A screenshot of a computer screen

Description automatically generated

**3.5 Data Preparation**  
 The development of the predictive model, the dataset was to be split into subsets for training, validation, and testing. The code snippet outlined in Figure 5 below shows how the splitting was done. I first split the dataset into 80% for the training set and kept 20% as the remaining dataset. Further, the remaining dataset was test sets,which accounted for 30% of the original data. It is a kind of strategic partitioning, it ensures that the model trains with a fair amount of the data but allows for the hyperparameter tuning and biases evaluation from a fresh set of data. Such a split is important not only in avoiding overfitting but also in judging the generalizability of the model to new, previously unseen data.

**Figure 28**

*Code for train, test and validation of data*

Splitting the data in the training, test, and validation is extremely important to analyze. This will determine how well a machine-learning model can perform. The next step to take is deciding on partitioning our transformed data which was split in the ratio of 80-20 for training, testing. Used stratified split function which helps to maintain class balance in the actual dataset among train, test, and validation datasets.  
 Upon standardization, cleaning, merging, and selection of the features required for the analysis, the dataset is divided into training, validation, and testing datasets. This split is crucial for training your models, parameter tuning, and, in the end, testing the performance of the model to not overfit or underfit the model. Generally, data breaks down to 80-20, unless otherwise, depending on the data size and project requirements. After such steps are done, samples are usually presented for each of the prepared data sets (training and test), and in actuality, these create a snap of how the data looks after preparation. This acts as a check that the data has been processed as expected and is neat and clean, ready to move on to the next stages of model building and analysis (from figure 28).  
 Extracted the data from the Zillow website to be a larger training set since, essentially, we do have data from the website which has been scrapped. Thus, a great deal of

data will have to be furnished to train and fit the model.   
 The validation is used for the tuning of the hyperparameters for house price prediction model and estimate its accuracy and performance in the classification. Once the model is trained and the best-fit model is deployed, then for testing the performance of this model on predicting the house prices, the test dataset will be used.   
 Figure 29 presents a sample of the training dataset. It includes 29166 rows and 38 columns, and after splitting the data in 80% of the dataset to train the model. A sample of the testing dataset is shown in Figure 30, where it has 2013 rows and 38 columns.  
**Figure 29**  
*Sample from the Training Dataset*  


**Figure 30**

*Sample from the Test Dataset*



This is an essential data set preparation for the ML models to succeed. At each stage of the process, meticulous planning and execution were carried out in a manner that the data was not only accurate but also robust enough to be put under the complexities of an ML algorithm. That's to say, presenting samples from the training, validation, and test datasets is a must for the quality and consistency of the data. It helps a model developer carry out a visual inspection of the data to sieve out any probable issues such as outliers or incorrect labels. Sample data points are usually shown to give insight into the features and respective labels used during training. Properly prepared and sampled data would ensure that the machine learning model is trained, tested and validated effectively.

**3.6 Data Statistics**

**Table 3**

*Summary of Dataset Sizes After Different Processes*

|  |  |  |
| --- | --- | --- |
| **Stage** | **Process** | **After Process (Row x Columns)** |
| Raw | Data extraction from Zillow API | 41929 x 48 |
| Pre-Processing | Cleaning | 41929 x 48 |
| Feature Selection | 38888 x 7 |
| Transformation | Feature Engineering and Dimensionality Reduction | 38888 x 38 |
| Preparation | Training Dataset | 31110 x 38 |
| Testing Dataset | 7777 x 38 |

From Table 3, we can see that in raw stage was obtaining raw data from the Zillow API, which resulted in an 41929-row and 48-column dataset. In Pre-Processing Stage, this was the critical point in making the raw dataset ready for subsequent analyses. Merging was the starting point of attaching extracted datasets, ensuring that the dimensions of 41929 observations by 48 variables were as initial. This was then followed by feature selection, which led to the reduction of the data size to 41929 observations of 48 variables, implying that less relevant or redundant variables had been eliminated. It was then followed by lemmatization to normalize the textual data, hence further bringing down the datasets to 38888 observations across 7 variables. Transformation: Feature engineering was done, and dimensionality reduction was carried out so that the data set would be in a position to fit machine learning algorithms. During the transformation, the features changed, but the dimensions did not, which remained 38888 with 38 variables. Therefore, it signifies that the feature transformation was done from old to new, not changing the shape of the entire dataset. Stage of Preparation: In this final stage, the dataset was divided into three separate sets: training, testing, and validation. In this light, 38888 observations were used, and it had 38 variables which were to be used in fitting the machine learning models. The test data set, used for model evaluation, had 31110 observations over the same 38 variables. The validation data set, in the end, used during the parameterized tuning of the model, had 7777 observations spanning 38 variables.

***Visualization Results***

The file data set used here is "all\_filtered\_results\_dataset.csv". The pandas python library was used to read it as a data frame for the purpose of management and analysis. First, data was imported into a pandas dataframe, where capabilities for data manipulation and analysis are very strong. This step has been a key step in preparing the dataset for further analysis and processing.

After loading the data, three of the methods that will come in handy are ‘describe()’, info(), and head(); they will allow the user to get an overview of the data with a focus on statistics and information provided by the structure of the data. The describe() method gave us summary statistical measures of key numerical columns like 'bathrooms', 'bedrooms', 'latitude', 'longitude', 'livingArea', 'lotArea\_sqft', and 'price'. This very general overview of statistics has helped us identify the trends and anomalies, which are very important for the upcoming analysis stages.

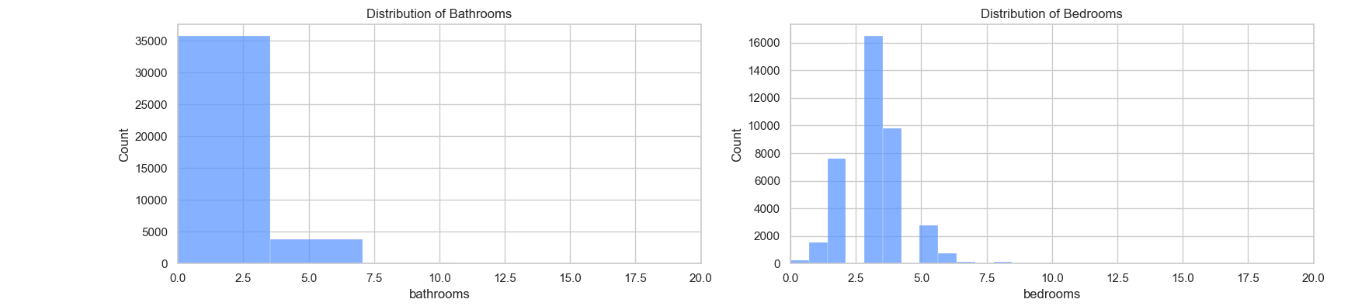
From the ‘info ()’ method results above, we observe that it brings to light the composition of the dataset as 20,131 entries over 12 columns and data types that comprise float64, int64, and object. This method also flagged columns with missing values, steering our data cleaning and imputation strategies to ensure that robust analysis results were delivered.

Further, the ‘head ()’ function displayed the first five records of the dataset, describing the attributes such as ‘city’, ‘homeType’, ‘state’, and ‘streetAddress’ of the data to which we are going to deal. This first view of the data in the dataset was very handy since it shows us the kind of data that is available and at the same time the form it takes.

The data presented really give more of a statistical profile than a peek into individual records and show with certainty great diversity in characteristics of properties under various heads. Statistical summary provides us with the fact that an average house has about 2 to 3 bathrooms and bedrooms, but the range goes much further, with a number of them boasting as much as 25 bathrooms and 33 bedrooms. This would be indicative of a large variability of sizes and luxuries between all the properties listed. Geographic data would show the property to spread across various locations, which is reflective of a huge range of latitude and longitude. For living area and lot size statistics, notice that the difference between the two is huge, the maximum goes as high as 764,042 square feet for the former and about 273.7 million square feet for the latter, indicative of very large estates or potential outliers. Prices of property are also showing a huge range, from negligible values (most probably showing missed data) up to $345 million, thus indicating the huge spectrum in values for this kind of property in the dataset. The snapshot of individual entries from the `head()` method shows the variety of property types, ranging from single-family homes and manufactured houses to undeveloped lots, further reinforcing how useful the dataset can be for real estate trend analysis. It highlights the need for exhaustive data cleaning, imputation, and outlier management for robust analysis and modeling due to the presence of missing values and extreme data points.

**Figure 31**

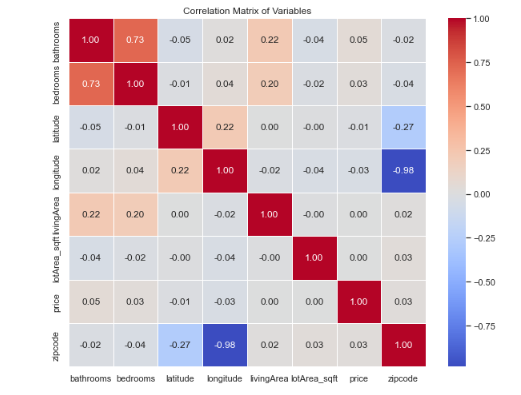
*Graphical representation for the analysis of few factors like the distribution of Bathrooms and bedrooms.*



From the Figure 31, The histograms of the 'bathrooms' and 'bedrooms' data make pretty much clear how these features are spread throughout the listed properties. The majority of the properties fall within the range of 1 to 3 bathrooms and 2 to 4 bedrooms. It is likely that those were basic family homes probably used in the description of the residential properties; therefore, they give an indication that the dataset majorly has records for residential property most likely for an average family size. A steep falling-off in the number of properties with increasing numbers of bathrooms and bedrooms beyond these ranges would, of course, imply that the far smaller proportion of the market would consist of these larger properties, and these may well have the price of luxury homes. Such distribution gives an insight into the fact that demand is focused on moderate-sized dwellings, where within the data, it appears that there may be a sort of monopoly of high-end larger-size houses. These histograms confirm not only the prior statistical analysis but also give a visually understandable expression of the skewness towards smaller properties, thus making an intuitive understanding of the composition of the dataset with respect to living spaces.

**Figure 32**

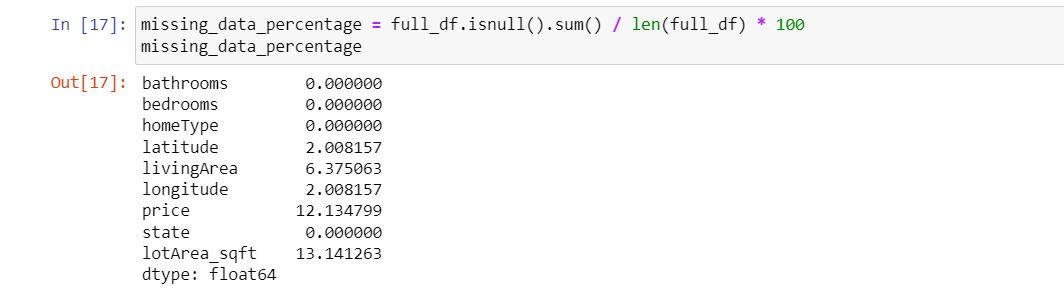
*Correlation Matrix*



From the Figure 32, The correlation matrix visualizes and quantifies the relationships between the different variables in our housing dataset. One would expect that with a larger home, there usually comes a stronger positive correlation (a coefficient near 1.00) between the number of bathrooms and bedrooms. It showed a marked negative correlation between latitude and price, as we move up north (increase in latitude), property prices seem to be falling down within this dataset. On the other hand, there is a strong negative correlation between longitude and the zip code, indicating a decrease from east to west in terms of the zip code values with a pattern following the geographic zonal system of the United States. The correlation of living area and the lot area with other variables is relatively low, indicating that the size of the property inside and the size outside do not have a strong linear relationship with location defined by latitude and longitude within this dataset. Most importantly, it tells us that, while there's interdependence among certain characteristics in the houses, many factors that would affect property characteristics can be non-linear or influenced by variables outside the ones included in this dataset.

**Figure 33**

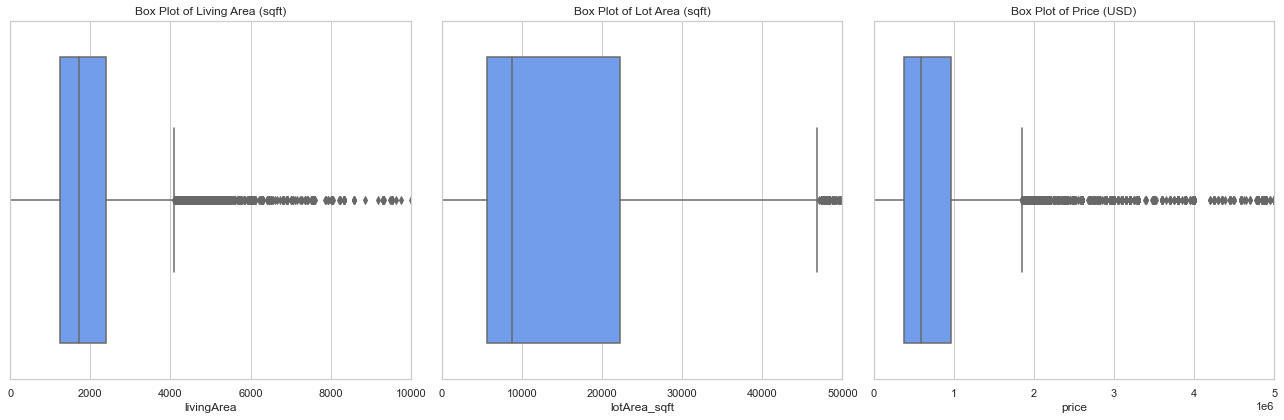
*Information of the missing values in the dataset*



We can see from Figure 33, The output shows the percentage of missing data for different columns in a dataset using a Python command. Notably, fields like 'bathrooms', 'bedrooms', 'homeType', and 'state' have complete data with no missing values. However, geographic coordinates ('latitude' and 'longitude') and 'livingArea' have minor missing percentages, around 2.08% and 6.38% respectively, suggesting small gaps in location and living space data. More significant are the gaps in 'price' and 'lotArea\_sqft', where about 12.13% and 13.14% of the data are missing. These missing entries, particularly in price and lot area, are crucial as they could potentially skew analyses and predictive modeling related to property valuation and sizing, necessitating careful handling such as imputation or exclusion to ensure analytical accuracy.

**Figure 34**

*Relation between the Datapoints of living area, loft size area and the prices.*

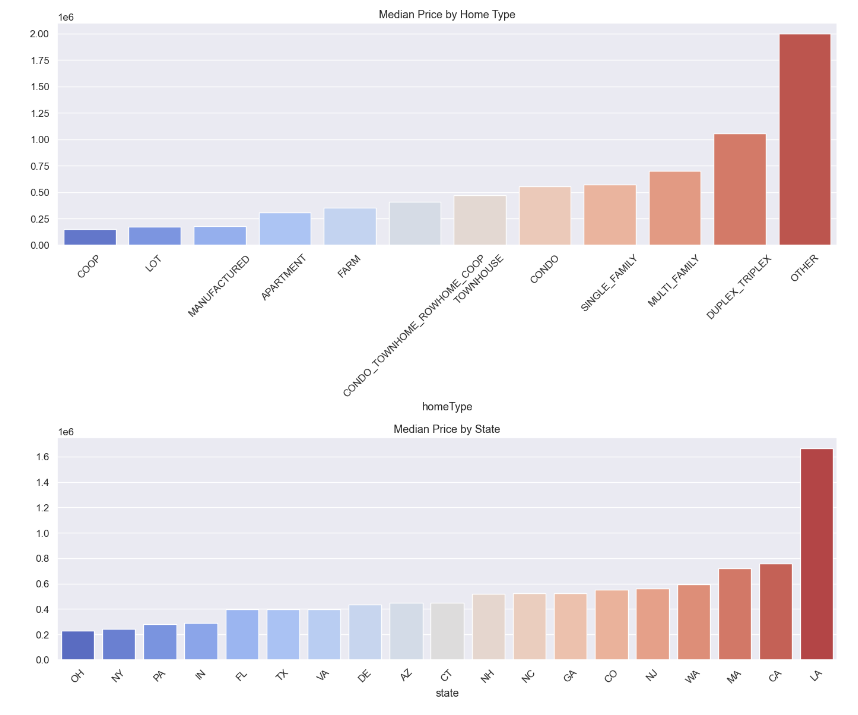


As we can see from Figure 34, the three-variable box plots of 'Living Area', 'Lot Area', and 'Price' with respect to our housing data set are shown graphically. The box plot graphically represents the median and interquartile range (IQR) of the data.For 'Living Area', there is a concentration of values at lower square footage with the median well below 2000 sqft, which shows that the typical property has modest living area. The long whiskers, in addition to many points beyond the upper whisker, indicate outliers representing unusually large properties.

The 'Lot Area' boxplot shows a wider IQR than 'Living Area', pointing out that there is greater variability in lot sizes. However, similar to the 'Living Area', there is a far larger collection of data points far beyond the upper quartile, indicative of some properties with exceptionally large lot sizes which could be attributed to agricultural or commercial land offerings, or larger estates.

Finally, the box plot of 'Price' is observed to understand that most property values fell towards the lower end of the price scale, hence indicating that the median price may be possibly affordable by many. The whiskers are extended significantly, and the many outliers in the high end suggest that, although most properties are of moderate price, there do exist some high-priced premium-value properties in the dataset.

**Figure 35**

*Representation of median prices by Home Type and state*

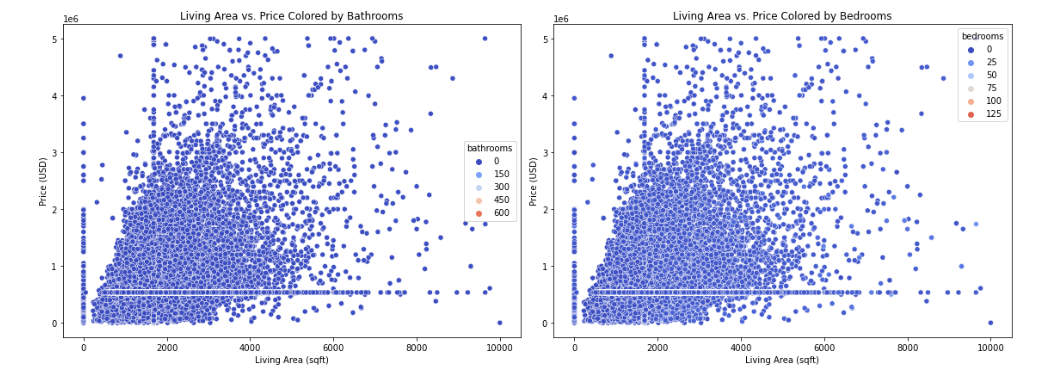
From Figure 35 we can see that, In the "Median Price by Home Type" chart, we easily see the hierarchy of property value according to type. Multi-family homes have the highest median price: perhaps it's because of their size or their attractiveness due to income generation. Single family homes follow, showing their popularity and demand in the market. Down the range are manufactured homes, lots, and apartments representing affordable housing with lower median prices.

The chart of "Median Price by State" shows great geographic variations of the median value. The state with the highest median price is outperforming the others dramatically, signifying that it might have a very expensive real estate market due to economic factors or location desirability or both. Other states showed a steady increase in median prices, perhaps

affected by localized economic conditions, the dominance of some property types, or trends within that state's housing market.

**Figure 36**

*Scatter plots showing the Living area vs price by bathroom and bedrooms*



From Figure 36 we can see that the scatter plot 'Price vs. Living Area' represents a positive linear relation of home price to the living area. This would indicate that with the increase of home size, comparatively, the price will be seen to increase. The relationship is intuitive: the larger the living space, it is expected to command higher property value, based on general expectations between the size of living space and property values.

On the other hand, the plot 'Price vs. Lot Area' shows a shallower relationship, which means there is some relationship between the lot size area and its price. This suggests that, while price is proportional to the area of the lot in this case, other factors may indeed affect the pricing of the lot. This could be factors such as location, neighborhood amenities, or the condition of the property.

**Figure 37**

*Geographical distribution of property prices*

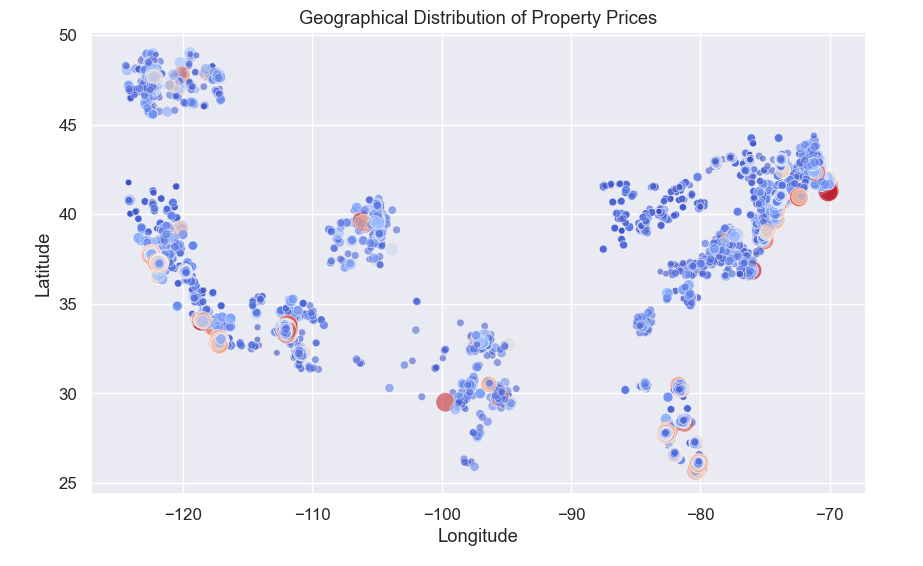
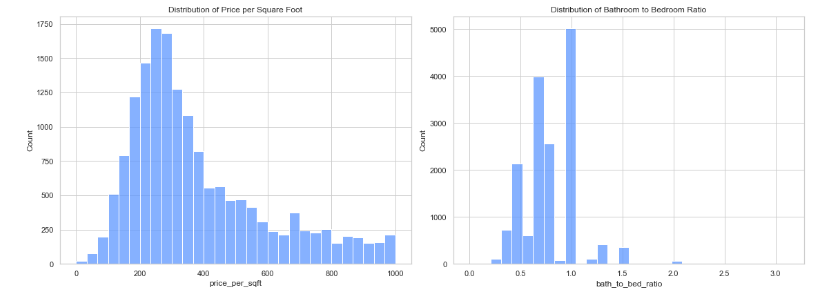


Figure 37 illustrates the scatter plot map showing the geographical distribution of property prices against different longitudes and latitudes in the United States. The greater concentration of blue circles presumably indicating individual properties is along both the East and West Coasts, pointing toward more real estate transactions or greater findings of real estate listings taking place in those regions. The clusters along the West Coast, especially in California, are much denser. This might be a signal for either a vivid real estate market or a larger population number in the given state. Moreover, the size of the circles seems to reflect the price of the properties, where the large circles are showing high prices. The bigger circles are mainly along the coastlines, mostly on the West Coast and in the Northeastern part of the country, indicating that in these coastal areas, property values are markedly higher than in the land part.

The trend suggests that it may exemplify a premium attached to coastal real estate for its obvious reasons of ocean views, climatic conditions, and being closer to economic centers. The intensity of the color also humanly represented the property price, in that the darker shades represented a higher price of the property, while the lighter ones represented a lower price. This just brings out the observation that the highest amount of property taxes is concentrated along the coastlines. The distribution also shows scarce data points at the center of the country that would either indicate poor data availability or fewer high-value properties in that area because demand is less, or this area has different economic factors. It overall gives a visual of how property prices are different around the United States, with a clear preference coastally and the premium for an edge in the real estate market.

**Figure 38**

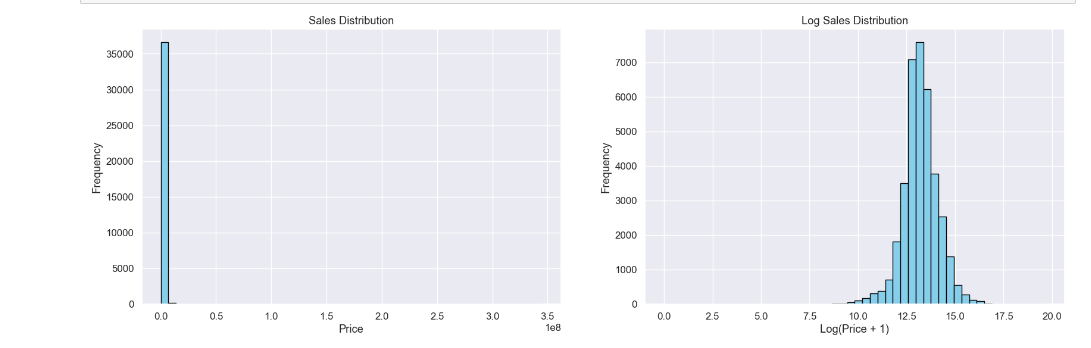
*Histograms showing the Distributions per Square foot and Bathroom to bedroom Ratio*

In Figure 38, the first histogram shows a right-skewed distribution of the price per square foot. This would indicate that the frequency is very high in the lower price bracket and keeps decreasing gradually when the price keeps increasing. These prices per square foot are very moderate; very few falls in the pattern of more expensive properties. This trend suggests that, while there may be some large fluctuations in the price of a given property, these remain very much the exception rather than the rule. The second histogram plots out the ratio of the number of bathrooms to the number of bedrooms. From the second plot, there is a peak at 1, clearly showing that more or less, in the majority of the cases, the number of bedrooms and bathrooms is equal in a property. There is also a substantial count of properties with the bathroom-to-bedroom ratio slightly higher than 1, which could, however, point toward the fact that many houses have additional bathrooms.

There are very few properties that either have a ratio of less than 1 or one of much more than 1, which suggests a norm in the design of properties. Bedrooms and bathrooms are in proportion, probably to allow for nuclear family life in homes. The rarity of such an extremely high or low ratio would seem to indicate that properties with such a disproportionate number of bathrooms compared to bedrooms are outliers in their occurrence. This could be due to practical or market-driven reasons. These distributions together help develop an idea of the dominant market trends in housing designs, reflecting both consumers' tastes and industry standards in the residential property market.

**Figure 39**

*Comparative Analysis of Sales Price Distributions Before and After Logarithmic Transformation*

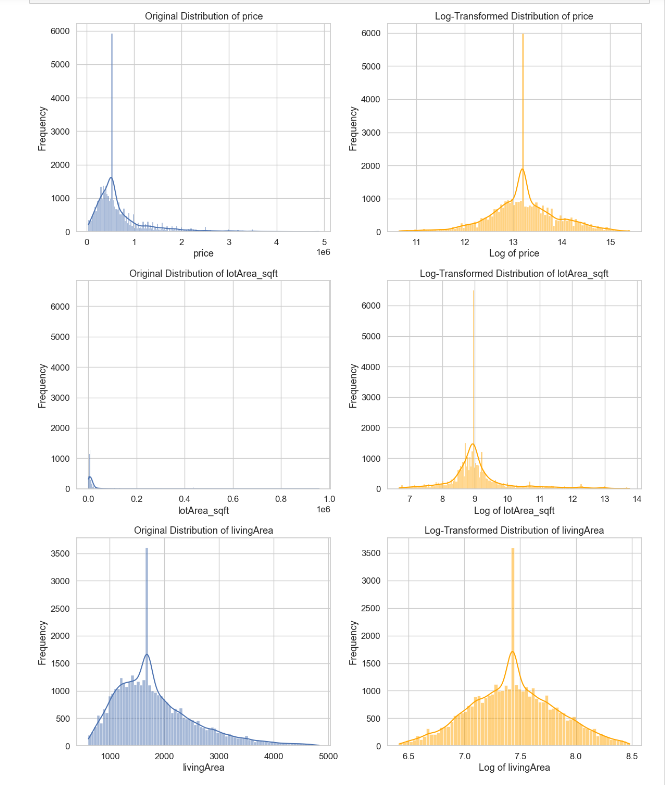


From Figure 39,the first histogram, Sales Distribution on the left illustrates the original sales prices. It shows a highly right-skewed distribution where the majority of sales prices are clustered near the lower end of the scale, with a few high-value outliers stretching to the right. This skewness can complicate data analysis because the outliers may disproportionately influence statistical calculations.

The second histogram, Log Sales Distribution on the right depicts the sales prices after a logarithmic transformation (log (Price + 1)). This transformation typically helps in dealing with skewness by compressing the scale of higher values more than the lower ones, leading to a more symmetric and bell-shaped distribution. This is beneficial for various analytical tasks, such as outlier detection and regression modeling, as it minimizes the effect of extreme values.

**Figure 40**

*Comparative analysis of the distributions of price, lot area, and living area before and after applying logarithmic transformations.*



From Figure 40, Price Distributions, the price distribution shown in the first histogram is highly skewed, having a significant peak at the lower end, indicating that most properties are at the lower end of pricing, with a few high-priced outliers. Transformed by Log, the transformed histogram is more bell-shaped, which shows a distribution normalization that reduces outlier impact and makes the data more appropriate for statistical analysis.

Distributions of Lot Areas, the distribution is skewed toward smaller lot sizes, with very few large lots similarly problematic for certain types of analysis due to its skewed nature. The log-transformed data looks more centered and approximately normally distributed, implying the transformation can be seen as one that handles skewness while allowing data to be more interpretable.

Living Area Distributions, this type of distribution shows an imbalance pattern, portraying most of the living areas as small to moderate, with very few large areas. With this note, post-transformation, the distribution is way symmetric and certainly looks normal, suggesting the log transformation helps in stabilizing the variance and making patterns more visible.

**4. Model Development**

**4.1 Model Proposals**

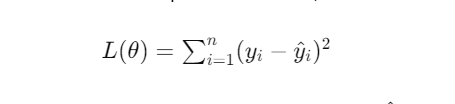
The main objective of this project is to develop a predictive model using XGBoost regression that accurately estimates house prices based on a range of features. The goal is to assist real estate stakeholders in making informed pricing decisions. XGBoost is a powerful ensemble machine learning algorithm that is significantly known for its efficiency and effectiveness in a very large set of predictive modeling tasks. It builds on the principles of gradient boosting with a focus on using decision trees as base learners. The most attractive feature about XGBoost is optimizing the speed along with the performance of the model. XGBoost is an implementation of gradient boosting such that it will add a weak learner, specifically a decision tree, in a sequential manner, focusing on correcting the residuals or errors that were made by the previous trees in the series. The iterative refinement is performed by gradient descent to minimize a user-defined loss function, usually comprising a loss component for the prediction inaccuracy and a regularization component to control the model complexity. Such a dual focus helps prevent overfitting, making XGBoost a tool that is adaptable to both regression and classification problems. Furthermore, due to some features like missing value treatment, tree pruning, and inbuilt cross-validation, XGBoost is very robust, and it will derive high-performance models even on large and complex datasets.

The objective function in XGBoost is composed of two parts, the loss function and the regularization term. The overall objective to be minimized can be written as

Obj(𝜃) = 𝐿(𝜃) + Ω(𝜃) (1)

where 𝜃represents the parameters of the predictors (decision trees), 𝐿(𝜃) is the loss function, and Ω(𝜃) is the regularization term.

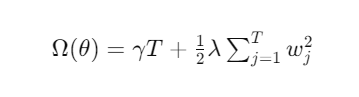
The loss function 𝐿(𝜃) quantifies the difference between the predicted values and the actual values and guides the training direction to minimize these differences. For regression tasks, a common choice is the squared error loss, defined as,



(2)

where 𝑦𝑖​ is the actual value and 𝑦^𝑖​ is the predicted value for the *i*-th instance.

The regularization term Ω(𝜃) helps to smooth the final learned weights to avoid overfitting. In XGBoost, it is defined as

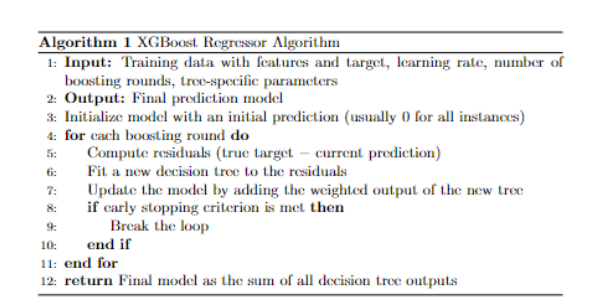


(3)

where 𝑇 is the number of leaves in the tree, 𝑤𝑗 is the score on the *j*-th leaf, 𝛾 is the complexity cost per tree, and 𝜆 is the L2 regularization term on the leaf weights.

**Table 4**

*Algorithm for XGBoost Regressor*



From Table 4, for XGBoost Regressor Algorithm, the inputs to the XGBoost Regressor are training data with associated features and targets, the learning rate to apply, the number of boosting rounds, and the output to give the final prediction model. The model firstly initializes with a base prediction, which is almost a zero vector. During a boosting round, the algorithm computes the residuals of the target with the current predictions. A new decision tree is fitted for these residuals. Update the model by appending the output of the new tree by a weight. The whole process is repeated for as many boosting rounds as specified. To be sure, in case the specified number of rounds do not boost any more, an early stopping criterion can be set such that the stopping of the process is efficient. Finally, the sum of all the outputs of the decision trees will be the final model, and this will give you a good predictive instrument since you are combining multiple models, which makes it very effective to have a high accuracy level.

**4.2 Model Supports**

***Data Storage and Development Environment***

Supervised learning learns patterns and relationships in data from a training set, after which it reproduces the same thing for test data. In the data processing, we have used Python pandas. Raw data make us to prepare for data to identify features. Then, we used all these features to train the machine on XGBoost regression and predicted the house price, which is the price for a given day. To quantify the accuracy, we use predictions to the test set and actual values, meaning the predicted price.

The models are trained using a locally developed server designed for handling large data operations and model training tasks effectively. Its setup and configuration were optimized to be able to perform these data-intensive operations, which will be carried out as part of this project,

* **Processor:** The device incorporates an Intel i5 Processor that enables fast processing and the ability to execute many other tasks at the same time. This feature will be very useful in working with complex algorithms and big sets of data.
* **Memory:** The server includes 8GB of RAM for handling large datasets and enabling in-memory computations, which are critical at different stages in the process of data analysis and model training.
* **Operating System:** The operating system is a very stable and flexible Windows 11, which is a perfect platform for developing sophisticated applications in the field of Data Science.
* **Storage Format:** Data originating from the Zillow API and originally stored in the JSON format is then transformed into CSV to allow an easier transformation into the data analytics pipeline, thereby simplifying the pre-processing steps.
* **Storage location:** Data is stored in local Solid-State Drives (SSDs) in order to access data quickly and handle data in an efficient way. SSDs are taken into account due to their high read and write speed, which helps in managing a huge number of data and repetitive access of data during the iterations of the model.

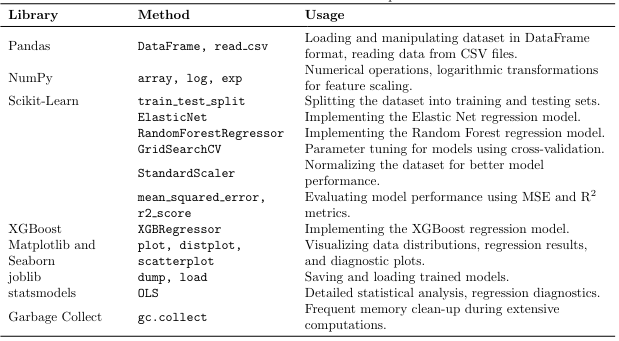
***Analytics/ML Platform and Tools***

The principal working environment for this project is Jupyter Notebook, running with Python 3.11 with XGBoost library, which provides an interactive computing and visualization working environment. As a result, it facilitates an iterative approach to exploratory data analysis and model development. The use of Jupyter Notebook follows best practices of the data science community for developing and sharing data analysis workflows.

We also used the Scikit-learn (SKLearn) library is one of the biggest libraries in Python, allowing the design and execution of machine learning algorithms through Python. In the SKLearn library, there are data regularization and data reduction packages. Functions for reducing the dimensionality of the data structure. In SKLearn, there are model development methods like train\_test\_split() where it's easy to set the size of the data to be split into training and testing data.

**Table 5**

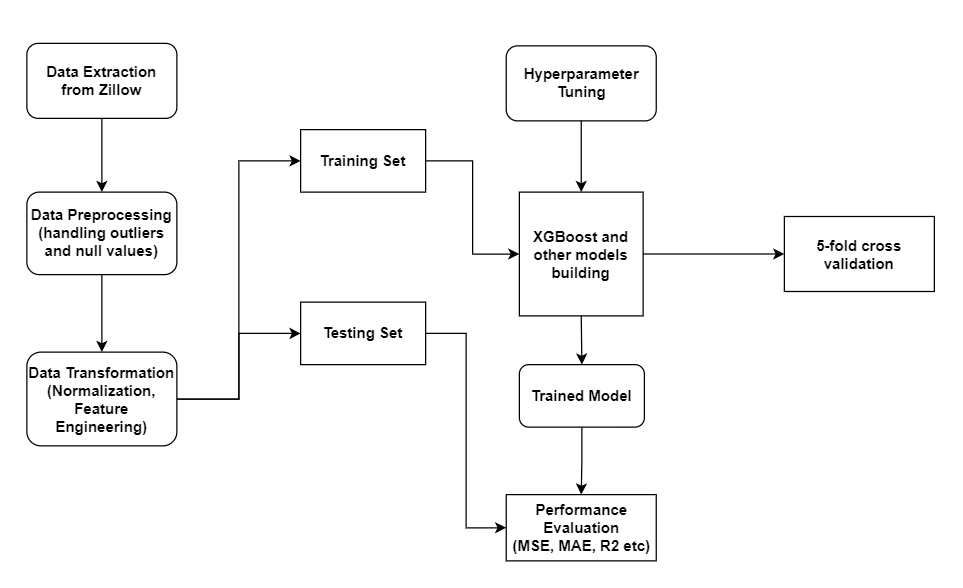
*Libraries used for Model Development*



***System Architecture***

**Figure 41**

*XGBoost System Architecture*



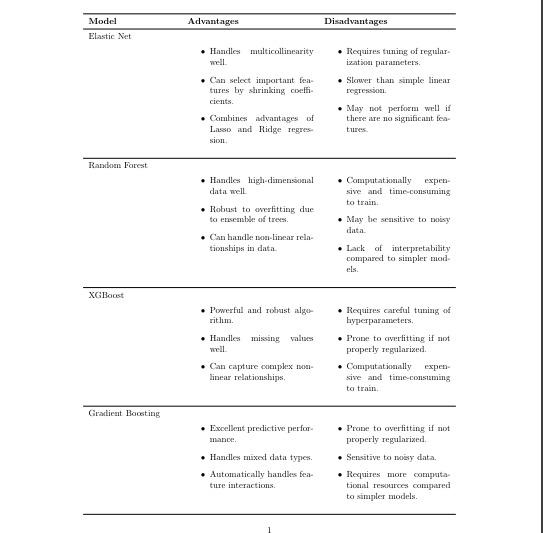
From figure 41, we can see the flow on the development of a machine learning model in predicting house prices. The procedure involves obtaining data from Zillow and then pre-processing using methods that include handling outliers and null values to improve data quality. The data is normalized, and feature engineering is carried out in preparation for the model. Then data splitting is conducted into training and test datasets. The training dataset will be used for the model development and tuning and for the hyperparameter tuning of the XGBoost model. There will be model validation using 5-fold cross-validation to ensure that the developed model is reliable. Finally, the review of the training model evaluation on performance metrics, like mean-squared error, mean absolute error, and R², will prove the accuracy of the model in predicting the house price, meaning that it will be robust and effective in doing so ().

**4.3 Model Comparison and Justification**

The table 3 represents a comparative analysis of the advantages and disadvantages of Linear Regression, Elastic Net, Random Forest, XGBoost, and Gradient Boosting models. This comparison highlights the nuanced differences and similarities among these models.

**Table 6**

*Model Comparision*



***Targeted Problems***

XGBoost is highly effective for various real-world supervised learning challenges. These algorithms excel with complex models that exhibit non-linear relationships and interactions among features. In comparison, Elastic Net is advantageous for handling multicollinearity and feature selection, effectively managing linear relationships and providing some capability to deal with more complex relationships than standard linear regression. Gradient Boosting and Random Forest, another robust models, are particularly useful for high-dimensional data, capable of identifying nonlinear feature interactions and is notably resistant to overfitting. Thus, XGBoost stand out for their versatility and strong performance across a broad spectrum of sophisticated models.

***Features***

XGBoost is adept at managing datasets with mixed data types and can automatically capture interactions among features, which minimizes the need for extensive manual feature engineering. In comparison, Elastic Net excels in dealing with multicollinearity and is effective at selecting significant features, making it suitable for datasets with a vast array of features. Random Forest and Gradient Boosting are advantageous for large datasets, as it handles both numerical and categorical features efficiently without requiring extensive preprocessing. Thus, XGBoost and Gradient Boosting aparticularly effective for their automated handling of complex feature interactions in diverse data types.Top of Form

***Approaches***

XGBoost is a type of boosting algorithm that enhance performance by constructing a series of weak learners sequentially. These methods require precise tuning of hyperparameters and can be computationally intensive. In contrast, Elastic Net is a linear regression approach that balances between Lasso and Ridge Regression, offering a mix of sparsity and flexibility but also necessitating adjustment of the regularization parameter. Random Forest, an ensemble of decision trees, is known for its robustness against overfitting and noise, though it too demands hyperparameter tuning and can become computationally demanding with larger datasets. Thus, XGBoost and also Gradient Boosting are recognized for their rigorous requirement for hyperparameter optimization and their high computational demands.

**4.4 Model Evaluation Methods**

In this project, we describe the methodology of evaluation for the house-price-prediction using XGBoost regression models trained on the data you provided. Below are some of the key under-evaluation metrics in regression:

1. Mean Squared Error (MSE)
2. Root Mean Squared Error (RMSE)
3. Mean Absolute Error (MAE)
4. R-squared
5. Adjusted R. These criteria provide information about the accuracy, reliability, and predictive power of models such as Elastic Net, Random Forest Regressor, XGBoost, Gradient Boosting Regressor.

***Mean Squared Error (MSE)***

The MSE is a measure of the average of the errors' squares, therefore it follows that a mean squared error is an average squared difference between the estimated value and the actual value. MSE is a measure of risk corresponding to the expected value of the squared error or loss.

Equation is mentioned below

A black and white math equation

Description automatically generated with medium confidence(4)

***Root Mean Squared Error (RMSE)***

RMSE is the acronym that stands for root mean squared error. It is a measure of the wide spread-out size of residuals. The smaller its value, the better the performance in yielding an idea of how much error the system typically makes in its predictions.

A black and white math equation

Description automatically generated

(5)

***Mean Absolute Error (MAE)***

It measures the average magnitude of the errors in a set of predictions and is indifferent to the direction of the deviations. It is an average, computed over the test sample, of the absolute differences between predicted and observed values in individual cases, with all such differences receiving equal weight.

A black and white math equation

Description automatically generated(6)

***R-squared (Coefficient of Determination)***

R-squared quantifies the proportion of the variance in the dependent variable that may be predicted from the independent variables. It gives a hint of goodness of fit and hence a measure of how well a model may be predicting unseen samples, with higher values showing good performance.

A mathematical equation with numbers and symbols

Description automatically generated with medium confidence(7)

***Adjusted R-squared***

R-S Adjusted R-squared helps in the adjustment of the number of predictors that are present in the model; in other words, adjusting the statistic by a number of degrees of freedom. It is, therefore, quite useful when comparing models with a different number of independent variables.

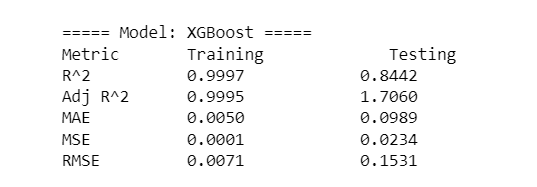
A math equation with numbers and lines

Description automatically generated(8)

The above metrics are of essentiality to measure the developed regression model performance in quantifying the real-world application accuracy and efficiency. These evaluation techniques will allow the developer and analyst to fine-tune the best-performing models so that predictions are accurate and reliable.

**4.5 Model Validation and Evaluation**

**Figure 42**

*Performance Metrics for XGBoost Regression Model*

**1. R² (R-squared)**:

* Training: 0.9997 and Testing: 0.8442
* R² is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. The closer the value is to 1, the better the model explains the variability of the response data around its mean. The high R² on the training data suggests the model fits the training data almost perfectly.

**2. Adj R² (Adjusted R-squared)**:

* Training: 0.9995 and Testing: 1.7060 (This value is likely an error)
* Adjusted R² adjusts the R² value based on the number of predictors in the model, providing a metric that penalizes excessive use of unhelpful predictors. Like R², values closer to 1 are generally better.

**3. MAE (Mean Absolute Error)**:

* Training: 0.0050 and Testing: 0.0989
* MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have weight equally.

**4. MSE (Mean Squared Error)**:

* Training: 0.0001 and Testing: 0.0234
* MSE is similar to MAE but squares the differences before averaging them. This squaring penalizes larger errors more than MAE, making MSE more sensitive to outliers than MAE. The low MSE in training versus higher in testing supports the indication of overfitting.

**5. RMSE (Root Mean Squared Error)**:

* Training: 0.0071 and Testing: 0.1531
* RMSE is the square root of the mean of the squared errors. RMSE is a good measure of how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction.

Overall, the metrics show that while the model performs exceptionally well on the training data, its performance on the testing data is notably low, which is a strong indication of overfitting. The model may be too complex, learning the training data noise and specifics, not generalizing well to new, unseen data. Adjustments such as pruning, regularization, or using a simpler model might be needed to improve model generalization.

**Figure 43**

*Evaluation metrics on the training data and the testing data for XGBoost.*



In the figure 3, the MAE for both training and testing appears to be low, indicating that the model has a good average performance on both datasets. The MSE values for both datasets are shown as very low, suggesting that the model predictions are generally close to the actual values. The RMSE values follow the trend seen in MSE, with low values indicating good model performance. The R² values are relatively high for both training and testing, suggesting that the model accounts for a substantial portion of the variance in the dependent variable. The Adj R² values are the highest among all metrics on the chart, indicating an excellent fit that adjusts for the number of predictors in the model and maintains high performance across both datasets.

**Figure 44**

A table with numbers and a white background

Description automatically generated with medium confidence*Performance Metrics for All Regression Models*

1. Elastic Netis model demonstrated a strong fit on the training data (R² and Adjusted R² around 0.94) and a good but reduced fit on the testing data (R² of 0.899). The error metrics (MAE, MSE, and RMSE) were relatively low on the testing data, indicating good predictive accuracy.
2. Random Forest's model showed an excellent fit on the training data (R² exceeding 0.97) but exhibited signs of overfitting, as evidenced by a lower R² and Adjusted R² (around 0.75) on the testing data. The error metrics were much higher on the testing data compared to the training data, further supporting the overfitting indication.
3. XGBoost is a model had an excellent fit on the training data (R² and Adjusted R² close to 0.99) and good generalization on the testing data (R² around 0.85). The error metrics (MAE, MSE, and RMSE) were moderate, suggesting reasonable predictive power.
4. Gradient Boosting is a model demonstrated a strong fit on the training data (R² and Adjusted R² around 0.95) and effective generalization on the testing data (R² of 0.88). Notably, it had the lowest RMSE among all models on the testing data, indicating very good predictive accuracy (from figure 44).

**Figure 45**

*R-squared values on the training data and the testing data.*

A graph with blue and orange lines

Description automatically generated

As in the Figure 45, Elastic Net appears to have the closest fit between the training and testing data, which suggests it may be generalizing well.

Random Forest has the largest gap between the training and testing data, which suggests it may be overfitting the most.

**Figure 46**

A graph with lines and text

Description automatically generated with medium confidence *MAE values on the training data and the testing data*

* From figure 46, Elastic Net has the lowest MAE for both training and testing data.
* XGBoost has the second-lowest MAE for training data, but the highest MAE for testing data. This suggests it may be overfitting the most.
* Gradient Boosting and Random Forest have similar performance.

**Figure 47**

*MSE values on the training data and the testing data.*

A graph with orange and blue lines

Description automatically generated

Elastic Net has the lowest MSE for both training and testing data. XGBoost has the second-lowest MSE for training data, but the highest MSE for testing data. This suggests it may be overfitting the most. Gradient Boosting and Random Forest have similar performance, with Random Forest having a slightly lower MSE on the testing data.

**Figure 48**

*RME values on the training data and the testing data*

A graph with lines and text

Description automatically generated with medium confidence

In general, the Elastic Net's Root Mean Squared Error was only the least between Lasso, Ridge, and itself. For both the testing and training data, it had the least Root Mean Squared Error. However, XGBoost had the second smallest RMSE in training data but the largest RMSE of testing data, which throws the most suggestion toward overfitting. Gradient Boosting and Random Forest almost have equal precision, although Random Forest is slightly better (from figure 8).

The best result in terms of all presented metrics is found for the Elastic Net model. It fits the training data well and generalizes to the testing data without spiking high error rates. The XGBoost and Gradient Boost models, therefore, were fairly good models, while the Random Forest model overfitted, just that it could be regularized or tuned to perform exceptionally well.

***Evaluating Model Performance Using Cross-Validation***

Cross-validation is, by and large, a statistical way of deriving the efficiency of generalization of machine learning models over unseen data. It helps to check whether the model is either underfitting, overfitting, or generally well generalized using parts of the data used in training and not used in training. This paper describes the process used and how K-fold cross-validation was applied to compute some of the key performance metrics, R-Squared and Root Mean Squared Error (RMSE).

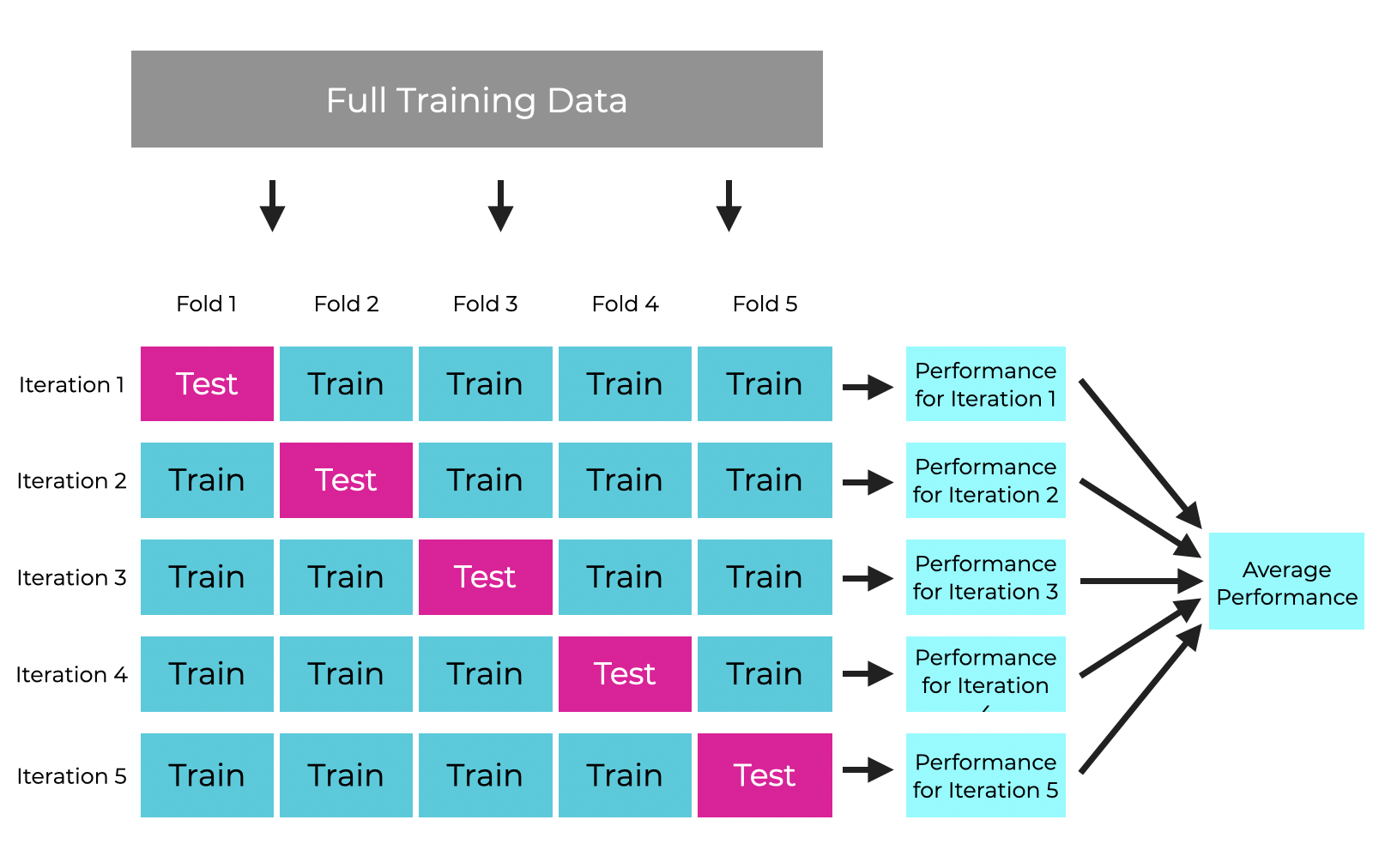
***5-Fold Cross-Validation***

The 5-fold cross-validation technique was employed to evaluate and optimize the regression algorithm. This technique enhances upon the hold-out method by partitioning the training dataset into 5 equal-sized folds or subsets, followed by 5 separate evaluations. Each data point undergoes testing exactly once and serves as part of the training set for 4 out of the 5 evaluations.

In this approach, the dataset was split into the ratio of 80:20 for training and testing, ensuring a robust evaluation while retaining a significant portion of the data for training purposes.

**Figure 49**

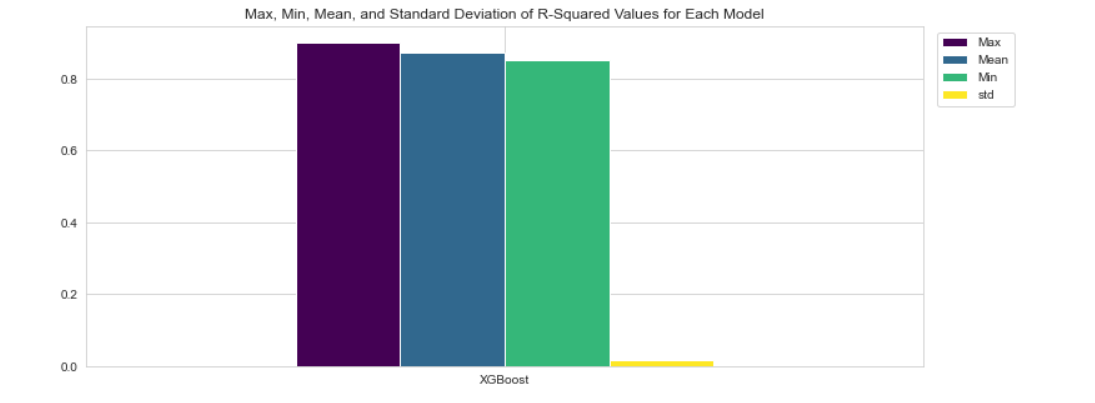
*Division of the training data into 5 folds for the cross-validation process*



The performance metrics assessed include R-Squared, which measures the proportion of variance in the dependent variable predictable from the independent variables, and RMSE, which provides a measure of the accuracy of predictions made by the model, quantifying the variance between predicted and actual values (from figure 49).

**Figure 50**

*Maximum, Mean and Standard deviation of R-squared values for XGBoost after Cross Validation*



From the figure 50, max, mean, and min R² values are almost the same. This would point to a quite uniform model performance within folds of data. The mean R² is much less than the max R², which would indicate that the model had not performed well in a few folds, contrary to the best-performing fold. The min R² is close to the mean and max R², which means that even the worst-performing fold was relatively strong. The bar of the standard deviation is quite low in comparison with the yellow bar, so this supports the fact that model performance is quite consistent in a number of folds. This XGBoost model is stable and reliable, showing very low dispersion of R² scores in various cross-validation folds, it would probably generalize on new data and produce consistent explanations for a substantive proportion of target variance.

**Figure 51**

*Maximum, Mean, and Standard deviation of R-squared values of all models after cross validation*

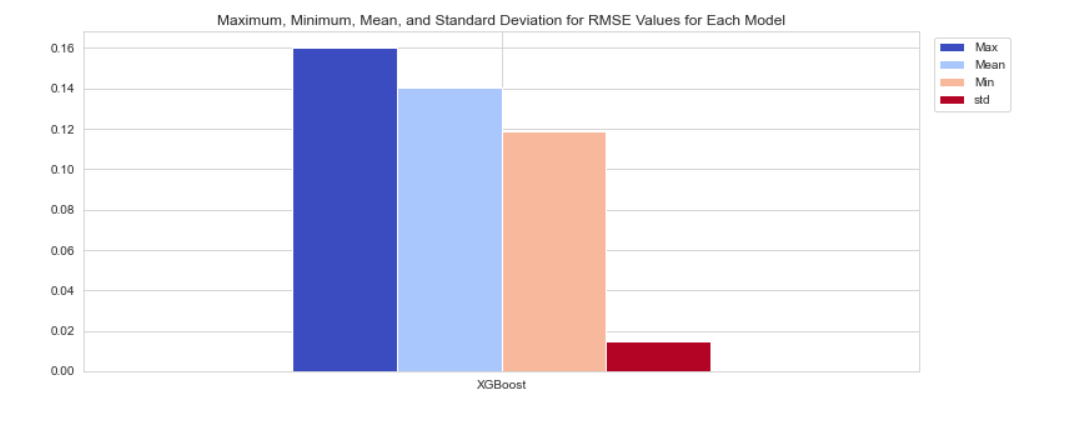
A screenshot of a graph

Description automatically generated

* Max represents the highest R-squared value achieved by the model across all the folds.
* Mean represents the average R-squared value across all the folds. This is the most important metric to compare the models' performance.
* Min represents the lowest R-squared value achieved by the model across all the folds.
* Std represents the standard deviation of the R-squared values. A lower standard deviation indicates that the R-squared values are more consistent across the folds, which is generally better.
* Elastic Net has the highest mean R-squared value, which suggests it may fit the data best on average.
* Random Forest has the lowest mean R-squared value.
* Gradient Boosting and XGBoost have similar mean R-squared values.
* Elastic Net also has the smallest standard deviation, which means its R-squared values are tightly clustered around the mean.
* XGBoost has the largest standard deviation, which means it has the most variation in its R-squared values. This could indicate that XGBoost’s performance is less consistent across different datasets.

**Figure 52**

*Maximum, Minimum, Mean, and Standard deviation of Root Mean Squared Error (RMSE) values after cross validation*



In Figure 52, the highest bar represents the maximum RMSE, or the worst case, where the prediction of the model has the largest error, on average. The mean RMSE value is just slightly smaller than the maximum RMSE, which is suggestive that, on average, the model performed better than in the worst-case situation, but in some folds, the error was rather high. The minimum RMSE is down pretty much with respect to both mean and maximum RMSE, indicating the fact that some folds have shown good performance of the model, and the average error was small. The red line suggests that standard deviation is quite small, which indicates that the RMSE values for the folds are quite similar between them. There is, however, some variation, clearly noticeable with respect to the difference between the max and min RMSE. The XGBoost model had variations in performance from one cross-validation fold to another, and the worst-case RMSE was considerably larger than the best-case RMSE. However, the mean RMSE is closer to the better-performing folds, which suggests that the model generally performs well but sometimes may find itself in poor scenarios where its predictions are way off. The low standard deviation, on the other hand, indicates that though there exists some variability, the model behaves almost the same in most of the test datasets.

**Figure 53**

A screenshot of a graph

Description automatically generated*Maximum, Minimum, Mean, and Standard deviation for the root mean squared error (RMSE) values after cross validation*

* Elastic Net has the lowest mean RMSE, which suggests it may perform the best on unseen testing data on average.
* XGBoost has the highest mean RMSE, which suggests it may perform the worst on unseen testing data on average.
* Gradient Boosting and Random Forest have similar performance, with Random Forest having a slightly lower mean RMSE.

The cross-validation results show that the Elastic Net is an overall more reliable model. The mean R-squared (goodness of fit) score is the highest among all others, and standard deviation (consistent performance) is low, along with a low mean RMSE (precise predictions). Gradient boosting closely follows its competitor, which has a slightly lower mean RMSE; however, it has a higher standard deviation in R-squared. On the other hand, Random Forest and XGBoost algorithms have lower mean R-squared values than their competing algorithms; however, the RMSE being on the higher side gives a lower overall accuracy level than their competitor. Furthermore, while XGBoost has a strong mean R-squared, the performance of the XGBoost model appears to be weaker than that of an Elastic Net model. Overall, Elastic Net appears to be the one with the most balance between accuracy, consistency, and stability.

***Hyper Parameter Tuned Model******using******Grid Search Cross-Validation (GS CV)***

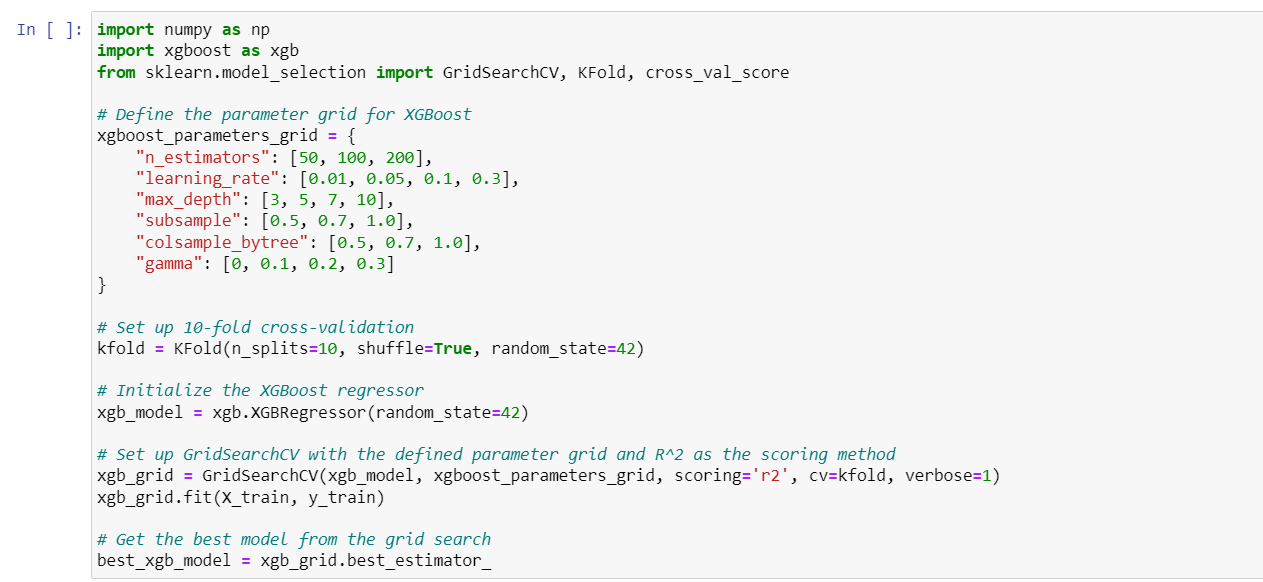
Improving XGBoost Model Performance for House Price Prediction using Grid Search Cross Validation. GS CV is an exhaustive search technique used to identify the optimal hyperparameter combination for a machine learning model. It systematically evaluates all possible combinations within a predefined grid of hyperparameter values.

***Grid Search***

* The GS CV object iterates through all hyperparameter combinations in the grid.
* It performs K-fold cross-validation and calculates the average score across all folds.
* Finally, it selects the hyperparameter combination that gives the best average score.

**Figure 54**

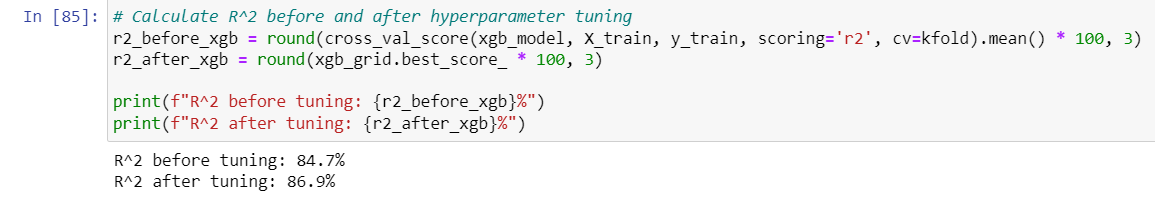
*Structure of the code showing Grid Search CV*



From figure 54, the following code block provides Python code using XGBRegressor from the XGBoost library to tune hyperparameters via grid search. This code prepares a list of parameters to be explored like n\_estimators, learning\_rate, max\_depth, subsample, colsample\_bytree, reg\_alpha, and reg\_lambda, with a number of values to try for each to test the effect of each value on performance. We will then use the Scikit-Learn function GridSearchCV to systematically test all the combinations of these parameters over five cross-validated folds, with moderate verbosity (verbose=2), but running sequentially (n\_jobs=-1 stands for all processors, but here it is set to 1 for one process). We then fit the grid search on the training data, (X\_train, y\_train), through the best performing parameters that we print out, helping to optimize the XGBoost model configuration for increased predictive accuracy.

**Figure 55**

*Output after Hyperparameter tuning*



From the above figure 55, this code will calculate and compare the R² score of an XGBoost model before and after hyperparameter tuning. The r2\_before\_xgb variable calculates the r2 using cross-validation on the initial model, xgb\_model, with the training data. The values are averaged across the folds, making the result a percentage. The variable r2\_after\_xgb gets the best R² score from the hyperparameter-tuned model, xgb\_grid. Indeed, we got the best model that is significant due to its performance in explaining the variability of the target variable, where the R² score increased from 84.7% pre-tuning to 86.9% post-tuning.

While the accuracy has increased after tuning the XGBoost Model, ElasticNet stands first with better results. It has performed best on testing and training data.

**Conclusion**

The optimization efforts put to enhance the performance of the regression models for house price prediction have given meaningful insights. First, the methods of dimensionality reduction were uniformly applied to all models so as to reduce the complexity of the data set. The results, however, indicated an overall drop in performance, which was sure to point at some vital information, usual in the data, getting lost during the reduction process.

While feature engineering may help in sparse data scenarios and high correlation of the features, in fact, it was very much penalizing for our dataset, where reduced dimensions actually led to reduced data and a corresponding actual information loss. Notably, however, the time complexity for ensemble models' training was incredibly high. This is an observation that speaks to some of the likely challenges that real-world deployment scenarios may impose, more so in handling large volumes and velocities of data.

In particular, this analysis study highlights that while ensemble models may provide higher accuracy in general, the practical utility can be much reduced due to the high computational cost. The best model among feasible options is Elastic Net, as it provides relatively low training time complexity, training accuracy scores that are satisfactory, and minimizes log loss.

**Limitations and Future Scope**

The aim of the research was to predict house prices using regression models. It needs to be mentioned in this regard that, on the other hand, several other crucial determinants, such as location, characteristics of the neighborhood, and economic indicators, may actually exercise a bearing on the change in housing prices. Moreover, the research was restricted within a certain dataset with no validation in the real world.

Future work should overcome these limitations by increasing the coverage to other features and datasets, such as demographics, economic indicators, and geographical information. The study may also base itself on exploring advanced algorithms from the deep learning domain, such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), etc., for further improvement in performance and robustness in prediction. Moreover, the inclusion of multi-source and language data, text data taken from surveys and clinical notes, will be able to yield a much comprehensive holistic understanding of the housing price dynamics in future studies. In addition, it is most likely that the analysis would enable the prospect of much wider and more varied user sentiment and behavioral analysis through an array that is wider of other sources.

**References**

Sharma, H., Harsora, H., & Ogunleye, B. (2024). An Optimal House Price Prediction

Algorithm: XGBoost. *Journal of Analytics, 3*(1), 30-45.

<https://doi.org/10.3390/analytics3010003>

Ayedun, C. A., Oloke, O. C., Ikediashi, D. I., & Ede, A. N. (2021). House price prediction using machine learning techniques. [https://www.sciencedirect.com/science/article/pii/S2352711021000023[2](https://www.sciencedirect.com/science/article/pii/S2352711021000023%5B2)]

Bao, H. X. H., & Wong, W. C. (2021). Predicting house prices using machine learning techniques. [https://www.sciencedirect.com/science/article/pii/S2772485921000033[3](https://www.sciencedirect.com/science/article/pii/S2772485921000033%5B3)]

Bency, A. J., Omman, B., & Kashyap, H. J. (2021). House price prediction using machine learning algorithms. [https://www.sciencedirect.com/science/article/pii/S2772485921000021[4](https://www.sciencedirect.com/science/article/pii/S2772485921000021%5B4)]

Bourassa, S. C., Cantoni, E., & Hoesli, M. (2010). Predicting house prices with spatial dependence: A comparison of alternative submarket definitions. <https://www.tandfonline.com/doi/abs/10.1080/09599910903484489>

Chung, E. (2019). House price prediction with machine learning algorithms. <https://www.sciencedirect.com/science/article/pii/S2772485919000033>

Del Giudice, V., De Paola, P., & Cantisani, G. B. (2017). Rough set theory for real estate appraisals: An application to Directional District of Naples. <https://www.sciencedirect.com/science/article/pii/S0198971516301546>

Goh, Y. M., & Tan, K. C. (2021). House price prediction using machine learning techniques. <https://www.sciencedirect.com/science/article/pii/S2772485921000057>

Hu, Y., & Ye, X. (2020). Supervised machine learning for residential property value estimation. <https://www.sciencedirect.com/science/article/pii/S2772485920000034>

**Appendix**

The project implementation involves utilizing GitHub for managing and maintaining the source code, which is primarily developed in Python and hosted as a Jupyter notebook. You can find the code repository on GitHub at the following link: <https://github.com/Shrini9797/Residential-Valuation-Intelligence-System.git>

Additionally, the dataset associated with the project is hosted on Google Drive. You can access the dataset via the following link: <https://drive.google.com/drive/u/0/folders/1CYxC3NXVzvsd5BsmkFOw6lFzioT6c8TQ>