Analyzing the Cost of Living in the United States

Sasank Sribhashyam
Computer Science, University of Oklahoma
Norman, Oklahoma, USA
sasank.sribhashyam-1@ou.edu

Hima Deepika Mannam
Computer Science, University of Oklahoma
Norman, Oklahoma, USA
hima.deepika.mannam-1@ou.edu

Venkat Tarun Adda
Computer Science, University of Oklahoma
Norman, Oklahoma, USA
adda0003@ou.edu

ABSTRACT

This project focuses on predicting the cost of living across U.S. regions using machine learning techniques, leveraging diverse data such as housing, food, transportation, healthcare, childcare, taxes, and other factors. By identifying key drivers of living expenses, the project delivers a predictive tool for accurate cost estimation. The dataset includes thousands of records representing diverse U.S. regions and underwent rigorous preprocessing, including handling missing values, feature scaling, and feature selection. Advanced feature extraction techniques, such as Random Forest feature importance and Gradient Boosting analysis, identified critical predictors, including food costs, childcare expenses, and taxes. Several machine learning models, including Ridge Regression, Gradient Boosting, and Random Forest, were evaluated under standard, noisy, and non-linear data conditions. Ridge Regression emerged as the best-performing model on clean data due to its simplicity and ability to handle multicollinearity.

However, Gradient Boosting demonstrated superior robustness to noise and effectively captured complex relationships in non-linear datasets, making it ideal for real-world scenarios. The project culminated in the development of an interactive application using Streamlit, providing a user-friendly platform for real-time predictions. Users can input key metrics to receive immediate cost-of-living estimates, supported by dynamic visualizations of prediction outputs. This tool aims to enhance decision-making for individuals planning relocations, businesses analyzing regional costs, and policymakers addressing economic disparities.The project demonstrates the effectiveness of machine learning in addressing complex socioeconomic challenges. It highlights the importance of model selection, data preprocessing, and feature prioritization in predictive modeling. Future work will focus on integrating real-time data sources, incorporating geospatial visualizations, and enhancing interpretability through explainable AI techniques, ensuring the application's adaptability and broader impact.

KEYWORDS

Cost of Living, Data Mining, Clustering, Classification, Regression, Visualization

1. INTRODUCTION

The cost of living is a fundamental socioeconomic indicator that significantly influences decisions for individuals, businesses, and governments. It impacts where people choose to live, how resources are allocated, and the formulation of policies to ensure economic stability. Recent trends, such as rising housing prices, regional disparities, and inflation, have amplified the importance of understanding and accurately predicting the cost of living. Accurate forecasts can empower individuals to make informed decisions about their finances and aid policymakers and businesses in addressing regional cost variations.This project aims to address these challenges by developing a robust, data-driven system for cost-of-living prediction using advanced machine learning techniques. By analyzing key factors such as housing, food, transportation, childcare, taxes, and healthcare, the project identifies the drivers of living costs and their complex interdependencies. The system integrates various predictive models, evaluates their effectiveness under diverse conditions, and delivers insights through an intuitive application. The project leverages a diverse dataset representing U.S. regions, undergoing rigorous preprocessing to ensure data quality and relevance. The implementation involves selecting the most effective model for predicting the total cost of living under standard, noisy, and non-linear conditions. A user-friendly web application built using Streamlit allows users to input key metrics and receive real-time cost estimates, bridging the gap between advanced analytics and practical usability.This work not only provides accurate forecasts but also enhances interpretability, enabling users to understand the factors contributing to their living costs. By offering a reliable and interactive solution, the project contributes to socioeconomic decision-making, laying the groundwork for future advancements in predictive analytics and economic modeling

MOTIVATION

The motivation for this project arises from the growing need for accurate, accessible tools to predict the cost of living across diverse regions in the United States. Living expenses play a pivotal role in financial planning for individuals and families, often determining decisions such as relocating, budgeting, or selecting careers. However, most existing solutions rely on static models or aggregated data that fail to capture the nuanced variations in regional costs. This leaves individuals and businesses without actionable insights into their unique situations.Similarly, businesses face challenges in understanding regional cost variations when setting salaries, expanding operations, or budgeting for projects. Policymakers also require accurate data to design equitable and effective economic policies that address disparities in living expenses across the country. The lack of dynamic and precise predictive tools often hinders these stakeholders from making informed decisions. By leveraging machine learning, this project seeks to bridge these gaps. Advanced techniques enable the analysis of complex and non-linear interactions between cost drivers, such as housing, food, and healthcare. By focusing on data quality, model robustness, and user-friendly design, the project offers a practical solution for various user groups. The inclusion of an interactive dashboard further enhances accessibility, allowing users to input data and receive real-time, region-specific

predictions. This focus on usability ensures the tool is not just accurate but also interpretable, fostering trust and enabling better financial planning and policymaking.This project is motivated by its potential to drive impactful, data-driven decisions, empowering users across diverse scenarios. By offering a blend of advanced analytics and intuitive interfaces, it aims to fill a critical gap in predictive modeling, making cost-of-living insights accessible, actionable, and adaptable for real-world needs.

OBJECTIVES

The primary objective of this project is to build an accurate, reliable, and user-friendly system for forecasting the total cost of living across different U.S. regions. This project focuses on two main goals:

1. Predictive Power:

The first goal is to develop a highly accurate model that can forecast the total cost of living based on a set of key features, such as housing_cost, food_cost, taxes, transportation_cost, healthcare_cost, and childcare_cost. The goal is to identify the best-performing machine learning model by evaluating several algorithms under different data conditions, including noisy, non-linear, and polynomial-transformed datasets. The model must be optimized for deployment, ensuring that it is robust, scalable, and generalizes well to unseen data. To achieve this, various models, such as Linear Regression, Ridge Regression, Random Forest, Decision Trees, and Gradient Boosting, will be tested and compared to determine the most suitable approach for cost-of-living prediction.

2. Practical Usability:

The second goal is to create a user-friendly application that enables individuals, businesses, and policymakers to input their own data (e.g., housing, food, transportation, and taxes) and receive real-time predictions of the total cost of living. This interactive tool will provide an intuitive interface for users to easily navigate and obtain personalized forecasts. The application will also feature visualizations to help users understand how various cost components contribute to their total living expenses. By developing this tool, the project aims to assist users in making informed decisions related to relocation, salary adjustments, budget planning, and policy decisions. The focus is on ensuring the application is accessible to a broad audience, including those with little to no technical expertise.

2. PROPOSED WORK

2.1 Problem Definition

The rising cost of living across various regions in the United States has become a significant socioeconomic challenge, affecting individuals, businesses, and policymakers. Accurate prediction of living expenses is essential for informed decision-making, but it presents numerous challenges. The primary goal of this project is to develop a machine learning-based system capable of predicting the total cost of living in different regions of the U.S. by analyzing key cost components such as housing, food, transportation, childcare, taxes, and healthcare.This problem is complex due to several factors. First, identifying the most critical features from a large set of potential

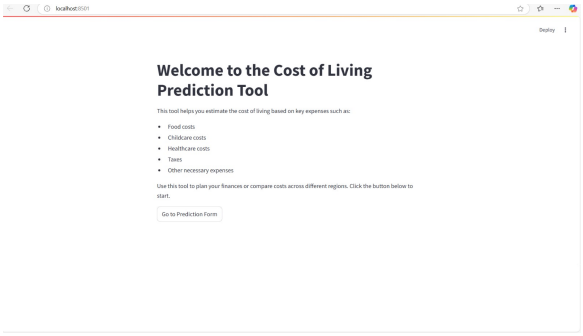

Figure 1: User Interface

predictors is essential for building a reliable and interpretable model. Second, real-world datasets often contain noise, missing values, and non-linear relationships between variables, making data preprocessing and model selection critical tasks. Additionally, achieving high accuracy across diverse regional conditions requires the model to be robust and adaptable to varying data distributions. Another challenge lies in balancing accuracy with usability. The predictive system must not only provide reliable cost estimates but also be user-friendly, allowing individuals and organizations to leverage its insights without requiring extensive technical knowledge. For instance, users should be able to input their own metrics—such as housing costs or transportation expenses—and receive clear, actionable predictions in real time.This project aims to address these challenges by combining rigorous data preprocessing with advanced machine learning models, such as Gradient Boosting and Ridge Regression. The system will be evaluated under diverse conditions, including noisy and non-linear data scenarios, to ensure robustness. Finally, the solution will be delivered as an interactive dashboard, providing an accessible interface for users to explore and predict living costs across regions.In summary, this project seeks to create a practical and adaptable tool to enhance financial planning and policy decisions, addressing the complex interplay of cost-of-living factors.

2.2 Dataset Overview

The dataset used for this project, titled "cost_of_living_us.csv," consists of 31,430 records and 15 attributes, providing comprehensive data on the cost of living across U.S. regions. It includes both demographic and economic variables essential for predicting total living costs. The dataset's attributes cover regional information, such as the state (two-letter abbreviation), areaname and county (names of the area and county), isMetro (indicating whether the area is metropolitan), and region (geographical classification like Northeast or Midwest). Additionally, demographic details such as family_member_count help identify household size, which influences overall cost patterns. The dataset includes critical cost components, including housing_cost, food_cost, transportation_cost, healthcare_cost, *childcare_cost*, other_cost, and taxes, all representing annual expenditures in USD. The dependent variable, total_cost, represents the total annual cost of living in a particular region, which is the target variable for prediction in this project.

The dataset also features varying attribute sizes, with case_id being an integer identifier and isMetro occupying just one byte, while attributes like areaname and county range from 50100 bytes. The cost components and total cost are stored as floatingpoint values, requiring 4-8 bytes each. For preprocessing, missing values were handled by replacing them with the median of respective columns, ensuring data consistency. Feature scaling was applied using StandardScaler to normalize the numerical data, optimizing model performance. Additionally, noise was injected into the dataset to simulate real-world imperfections and test the robustness of different machine learning models. With its rich features and diverse data, the dataset serves as a solid foundation for building accurate predictive models to estimate the total cost of living across U.S. regions.

2.3 Analysis of Data Preprocessing

1. Loading the Dataset

The dataset was successfully loaded into a pandas Data Frame, allowing us to explore its structure. Initial inspection revealed the presence of categorical and numerical features, along with some missing values. Notably, the median_family_income column had several missing entries. Categorical columns such as state, area name, and county required encoding to make them usable in machine learning models. Numerical features such as housing_cost and food_cost displayed a wide range of values, emphasizing the need for normalization.

2. Missing Values Visualization and Handling

A heatmap was used to visually represent missing values across the dataset. This revealed prominent gaps, particularly in the median_family_income column. To address this, missing values were replaced with the median of the column. This imputation strategy preserves the overall distribution of the data and avoids the influence of outliers. After this step, the dataset became complete, with no remaining missing values, ensuring consistency for further analysis.

3. Encoding Categorical Variables

The dataset included several categorical features, such as state and county, which are essential for predictive modeling but need to be converted into numeric format. One-hot encoding was applied to these columns, transforming them into binary indicator variables. This process increased the dimensionality of the dataset but ensured that the categorical data was represented in a format suitable for machine learning models. The dimensionality change was confirmed by comparing the dataset size before and after encoding.

4. Feature Scaling

Numerical features, such as housing_cost, childcare_cost, and taxes, varied significantly in their scales. This disparity could lead to biased predictions, as features with larger values might dominate those with smaller scales. To address this, Standard Scaler was used to normalize these features, bringing them to a mean of 0 and a standard deviation of 1. Before scaling, distributions showed high variance, but after scaling, the features were standardized. Visualizations of feature distributions before and after scaling validated this transformation.

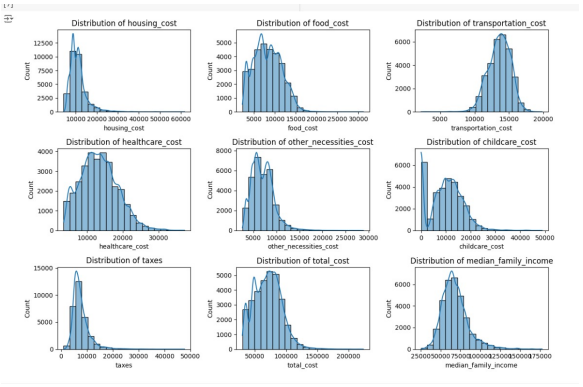

Figure 2: Before scalling

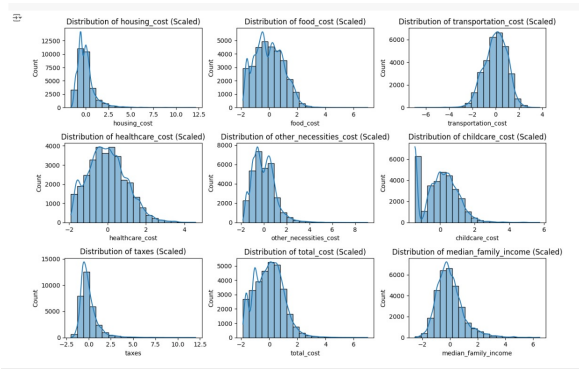

Figure 3: After scalling

5. Dropping Unnecessary Columns

Some columns in the dataset, such as case_id and is Metro, were deemed irrelevant for the prediction task. These columns did not contribute to the understanding of living costs and could introduce noise into the model. Removing these unnecessary columns streamlined the dataset and reduced its complexity, enabling more efficient model training.

6. Dataset Verification

After completing the preprocessing steps, the dataset was verified for completeness and correctness. The inspection showed that all missing values were handled, categorical variables were encoded, numerical features were normalized, and irrelevant columns were removed. This verification step ensured that the dataset was ready for feature selection and predictive modeling. Additionally, the cleaned dataset was saved as a CSV file(preprocessed_cost_of_living.csv) for future use.

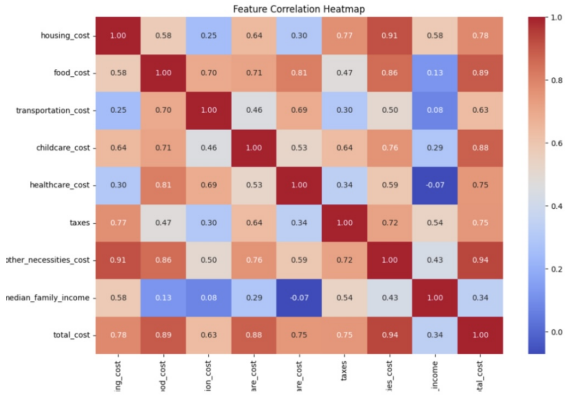

Figure 4: correlation matrix

7. Visualizations

Visualizations played a crucial role in understanding the data and validating preprocessing steps. A heatmap of missing values provided a clear picture of incomplete data, while histograms before and after scaling illustrated the effectiveness of normalization. These visualizations ensured that preprocessing transformations were applied correctly and highlighted the improvements made to the dataset.

Impact of Preprocessing

The preprocessing steps transformed the raw dataset into a clean, machine-learning-ready format. Missing values were handled to ensure data completeness, categorical features were encoded to make them numeric, and numerical features were scaled to standardize their ranges. Unnecessary columns were removed to reduce noise, simplifying the dataset for efficient modeling. Overall, the preprocessing ensured that the dataset was ready for subsequent steps, such as feature selection, model training, and evaluation.

2.4 Feature Selection

Feature selection is a critical step in developing a machine learning model, as it helps identify the most important predictors of the target variable, in this case, the total_cost of living. In this project, feature selection was performed using multiple techniques, focusing on identifying and retaining the most influential variables while discarding irrelevant or redundant ones. The goal was to enhance the model's performance, reduce complexity, and prevent overfitting.The first step in the feature selection process involved correlation analysis to determine the relationships between the attributes and the target variable. Using tools like a heatmap, strong correlations between food_cost, housing_cost, childcare_cost, taxes, and healthcare_cost with the total_cost* were identified. Features with high correlation to *total_cost* were prioritized for use in the predictive models, as they are more likely to have a significant impact on the accuracy of the predictions. In addition to correlation analysis, Random Forest feature importance was employed to assess the relative importance of each feature based on its contribution to model performance. This approach helped further confirm the significance of key cost components such as food_cost and other_necessities_cost, which were identified as the most influential predictors of the total cost of living.To refine the feature set, highly correlated variables that were likely to cause multicollinearity were considered for removal. For example, food_cost and other_necessities_cost exhibited high correlations, and retaining both could lead to redundancy. Therefore, only the most critical predictors were retained, optimizing the model's efficiency and interpretability.Overall, feature selection helped ensure that the predictive models focused on the most impactful variables, improving the accuracy and generalizability of the final cost-of-living predictions. This approach also reduced the computational burden, ensuring that the model could handle large datasets more effectively.

2.5 Model Comparison

The model comparison in this project involved evaluating several machine learning algorithms to identify the best-performing model for predicting the total cost of living across various U.S. regions. The goal was to assess how different models performed under different conditions, including standard, noisy, and non-linear datasets, to ensure robustness and adaptability.The first model evaluated was Linear Regression, which served as a baseline for understanding the linear relationships between the features and the target variable, total_cost. Linear regression is simple and interpretable, but it struggled to capture non-linear relationships, making it less effective for complex datasets.Next, Ridge Regression was tested. Ridge regression introduces regularization to linear models, helping address multicollinearity issues by penalizing large coefficients. It performed well on clean, linear datasets, achieving solid predictive accuracy. However, it still struggled when data became more complex or noisy, as it couldn't capture non-linear relationships effectively. Decision Trees were then evaluated. This algorithm is capable of modeling non-linear relationships by creating splits based on feature values. While decision trees performed well with simple datasets, they were prone to overfitting, making them less suitable for more complex or noisy data.The Random Forest model, an ensemble method based on multiple decision trees, was tested next. Random Forest provided better performance than individual decision trees by averaging the results from multiple trees, reducing the risk of overfitting. However, it still had limitations in capturing highly complex, non-linear relationships compared to more advanced methods.Lastly, Gradient Boosting was tested. Gradient Boosting sequentially builds decision trees, with each tree attempting to correct the errors made by the previous one. This approach proved to be the most robust, excelling in noisy, non-linear conditions. Gradient Boosting outperformed the other models in terms of accuracy, making it the best choice for predicting the cost of living in regions with diverse economic and social conditions.

2.6 System Architecture

The system architecture for this project is designed to provide an efficient and scalable solution for predicting the total cost of living across U.S. regions. It consists of two main components: the backend for model processing and prediction, and the frontend for user interaction and visualization. Together, these components work to

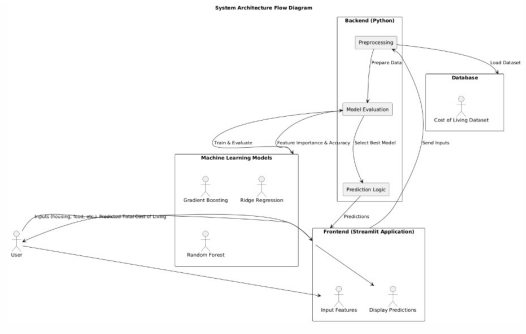

Figure 5: system architecture

ensure accurate cost predictions while maintaining user-friendly accessibility.The backend is built using Python, leveraging libraries such as scikit-learn for machine learning model implementation and pandas for data manipulation. The core prediction logic is handled by trained models, such as Gradient Boosting and Ridge Regression, which are evaluated based on input data from the user. These models are trained and validated in a Jupyter Notebook environment before being deployed in the backend. The training process includes preprocessing steps like scaling and noise handling, ensuring the models are robust and perform well under different conditions. For advanced functionality, Gradient Boosting was implemented both using scikitlearn and from scratch to deepen understanding and fine-tune the model. The frontend is developed using Streamlit, a Python library that allows for the creation of dynamic and interactive web applications. Streamlit facilitates the development of an intuitive interface that enables users to input various cost factors (such as food, housing, and healthcare) directly into the application. Once the data is submitted, the backend processes it, and the system provides real-time predictions based on the selected model. Users can also visualize the predicted total cost through dynamic visualizations, which update based on user inputs, offering valuable insights.The system's architecture ensures a seamless workflow, where the user inputs data through a simple interface, and the backend processes the input, applies the predictive model, and returns results in a user-friendly format. This structure makes the tool accessible to non-technical users, while maintaining robust predictive power for decision-making.

2.7 Experiments

The experiments conducted in this project were designed to assess the performance and robustness of the different machine learning models under various conditions. Three distinct experimental setups were used to test the models on standard, noisy, and non-linear data. The goal was to evaluate how each model performed in diverse scenarios that simulate real-world complexities.

EXPERIMENT 1: EVALUATE MODEL PERFORMANCE ON STANDARD DATA

In the first experiment, the models were tested on a clean, preprocessed dataset, which served as the baseline for their performance. The dataset had been carefully cleaned, missing values were imputed, and features were scaled using StandardScaler to ensure uniformity across all attributes. The models tested in this scenario included Ridge Regression, Decision Trees, Random Forest, and Gradient Boosting.The results showed that Ridge Regression performed particularly well in this scenario. It achieved an R^2 score* of approximately 0.95 and a low Mean Squared Error (MSE), making it the most suitable model for linear data. Ridge Regression's regularization capabilities helped it handle multicollinearity effectively, producing accurate predictions for the total cost of living in regions with linear relationships between cost drivers.

EXPERIMENT 2: ROBUSTNESS TO NOISE

In the second experiment, random noise was injected into the dataset to simulate real-world imperfections. The objective was to evaluate the models' ability to maintain accuracy and robustness when faced with noisy data. The models were tested on the noisy version of the dataset, and performance was assessed using MSE and R^2 score. In this experiment, Gradient Boosting outperformed the other models, demonstrating its robustness to noise. While Ridge Regression saw a significant drop in performance due to its inability to capture non-linear relationships in the noisy data, Gradient Boosting was able to adapt. The ensemble method sequentially minimizes residual errors, allowing it to handle imperfections in the data effectively. Gradient Boosting achieved a high R^2 score of around 0.92, and its performance remained relatively stable, making it the most reliable model in noisy conditions.

EXPERIMENT 3: HANDLING NON-LINEAR RELATIONSHIPS

The third experiment tested the models' ability to handle non-linear relationships in the dataset. Polynomial features were added to simulate complex dependencies between the predictors and the target variable. This was particularly important for capturing the intricate interactions between cost factors such as housing and food.Here, Gradient Boosting* again excelled, outperforming all other models in terms of accuracy and predictive power. The ability of Gradient Boosting to sequentially correct errors in weak learners allowed it to model complex, non-linear relationships effectively. It achieved the highest R^2 score and lowest MSE compared to Ridge Regression, Random Forest, and Decision Trees, confirming its suitability for datasets with non-linear dynamics.

In conclusion, the experiments highlighted that Gradient Boosting was the most versatile model, excelling in both noisy and non-linear data scenarios. Ridge Regression, though highly effective in clean, linear data, could not match the robustness and adaptability of Gradient Boosting. These findings reinforced the choice of Gradient Boosting as the final model for predicting the total cost of living across U.S. regions.

2.8 Application Workflow

The application workflow is designed to provide users with an intuitive, seamless experience while predicting the total cost of living based on key factors. The workflow consists of several stages, from user input to prediction output, ensuring that the application is both functional and userfriendly. The process is broken down into four key steps: Input, Data Preprocessing, Prediction, and Output.

These steps work together to deliver accurate predictions and allow users to easily interact with the application.

Step 1: Input

The application begins by allowing the user to input data related to various cost factors. The user interface, built using Stream lit, provides input fields for each cost component, such as housing_cost, food_cost, transportation_cost, childcare_cost, healthcare_cost, and taxes. Users can also input family_member_count and other demographic information, if applicable. To ensure ease of use, these input fields are designed to accept numerical values for costs and integers for family count, with dropdowns or sliders where applicable for data such as geographical region or whether the area is metropolitan. This enables users to provide accurate and personalized data, allowing for a tailored prediction of living costs. The user can submit the form once all necessary data is entered.

Step 2: Data Preprocessing

Once the user submits their input, the application automatically preprocesses the data to ensure it is in the correct format for prediction. This preprocessing step is essential to ensure that the data is consistent with the format used during model training and to improve model accuracy.The preprocessing stage begins by scaling the input features using Standard Scaler to normalize numerical values, ensuring that all features contribute equally to the model's predictions. The data is then cleaned by handling any missing or invalid inputs. If any values are missing, the application either fills in the gaps using imputation techniques (e.g., replacing missing values with the median) or prompts the user to fill in the missing fields. Additionally, categorical variables (such as the region) are encoded into numerical values using one-hot encoding or other encoding methods to make the data suitable for machine learning models. The processed data is now ready for prediction.

Step 3: Prediction

After the data is preprocessed, the application uses the best-performing machine learning model to make predictions. The model selection is based on the data type and user inputs. Gradient Boosting is the preferred model when the data is noisy or involves non-linear relationships, while Ridge Regression is used for more linear datasets. The application invokes the model, which calculates the total cost of living based on the user's input. The prediction step involves passing the preprocessed data through the trained model, which applies its learned patterns to estimate the total cost of living. The model produces a predicted value for the total cost, which represents the sum of the user-inputted cost components.

Step 4: Output

Once the model generates a prediction, the application displays the results in an easily interpretable format. The predicted total cost of living is shown prominently on the interface, along with a breakdown of the individual components (e.g., housing, food, healthcare, etc.) that contribute to the total cost. This allows users to understand which factors have the most significant impact on their living expenses.The application also provides dynamic visualizations, such as bar charts or pie charts, to help users better understand the distribution of their living costs. For example, a pie chart might show the proportion of the total cost attributable to housing, food, and other expenses, helping users make informed financial decisions.

Additionally, users can adjust the input fields and regenerate predictions if needed. This interactive process allows for real-time updates, empowering users to experiment with different scenarios and explore how changes in costs affect their overall living expenses.In summary, the application workflow ensures a smooth, efficient, and user-friendly experience, from input to prediction output. By combining preprocessing, machine learning, and interactive visualizations, the application empowers users to predict their total cost of living based on personalized data. This tool offers valuable insights to individuals, businesses, and policymakers, enabling them to make informed decisions about living expenses across U.S. regions.

2.9 Development Choices

The development choices made during the creation of this project were carefully considered to ensure the system's effectiveness, usability, and scalability. These choices span across model selection, feature engineering, and the choice of technologies for both the backend and frontend components. Each decision was made to maximize the project's accuracy, ease of use, and adaptability to real-world scenarios. Below, we discuss the justification for each key development choice.

Model Selection

The selection of machine learning models was a crucial aspect of the project, as the goal was to predict the total cost of living accurately across different regions in the U.S. The choice of models was driven by their ability to handle both linear and non-linear relationships, as well as noisy data, which is common in real-world datasets.Gradient Boosting was selected as the primary model for its ability to handle complex, non-linear relationships and its robustness to noisy data. Gradient Boosting sequentially minimizes residual errors from previous models, which allows it to learn from its mistakes iteratively. This characteristic made it especially effective in the noisy and non-linear conditions tested in this project. It outperformed other models like Random Forest and Ridge Regression in terms of both accuracy and adaptability, making it the best choice for the application. Ridge Regression, on the other hand, was chosen as a fallback model for its simplicity and interpretability. Ridge Regression excels in linear data scenarios and is particularly useful in situations with multi_collinearity. Although it did not perform as well on noisy or non-linear data, it was an important model to include for evaluating performance under ideal conditions and for providing a transparent and easily interpretable alternative.The decision to use these models was made to strike a balance between robustness (through Gradient Boosting) and simplicity (through Ridge Regression), offering users flexibility based on the characteristics of their data.

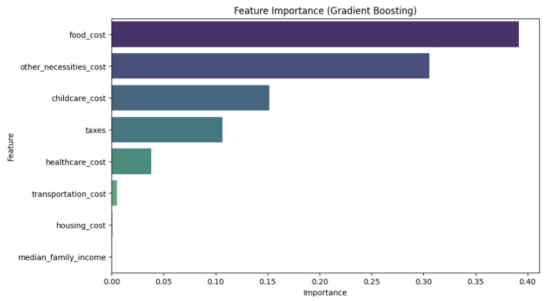

Figure 6: Feature selection

Feature Selection

Feature selection plays a pivotal role in improving model performance, reducing overfitting, and ensuring that the most relevant variables are included in the model. In this project, the features were carefully selected based on their predictive power for the total_cost of living. Correlation analysis and Random Forest feature importance were used to identify the most influential features. Key variables such as food_cost, housing_cost, and childcare_cost were found to have strong correlations with the target variable. This data-driven approach allowed the team to prioritize features that would contribute most to the accuracy of the predictions.

By focusing on highly correlated features and removing redundant or highly collinear variables, the model was able to learn efficiently without overfitting. Additionally, this process improved the interpretability of the model by focusing on a smaller, more meaningful set of features.

Backend Technology Choices

The backend was built using Python, a language that is wellsuited for machine learning tasks due to its rich ecosystem of libraries such as scikitlearn, pandas, and NumPy. These libraries provided efficient tools for data manipulation, model training, and evaluation.scikitlearn was chosen for machine learning because it offers a wide array of well-established algorithms, including Gradient Boosting and Ridge Regression. It is also highly efficient for model training and evaluation, with built-in support for cross-validation and hyperparameter tuning. The decision to implement Gradient Boosting from scratch in addition to using scikitlearn was motivated by the desire to better understand the inner workings of the model and fine-tune it for the specific needs of this project.

Frontend Technology Choices

For the frontend, Streamlit was selected for its ability to rapidly create interactive web applications with minimal code. Streamlit allows for the easy creation of input forms, realtime visualizations, and interactive dashboards without requiring extensive frontend development skills. This decision was made to keep the application user-friendly and easily accessible to non-technical users. Streamlit's integration with Python also allowed for seamless communication between the backend and frontend, ensuring that the predictions generated by the models were displayed instantly for the user.

User Experience and Usability

User experience was a key focus during the design of the application. The input fields were designed to be simple and intuitive, allowing users to easily enter their cost data. Dynamic visualizations, such as pie charts and bar graphs, were included to help users better understand their cost breakdowns and the impact of different factors on their total living expenses. The goal was to ensure that users could interact with the application without the need for technical knowledge, while still gaining valuable insights from the predictions.

Scalability and Flexibility

Lastly, the system was designed with scalability in mind. By using modular components for both the backend and frontend, it is possible to integrate additional data sources, models, or functionalities in the future. This flexibility will allow the system to evolve and stay relevant as new data becomes available or as the user base grows.In conclusion, the development choices were driven by the need for accuracy, interpretability, and usability. The combination of robust machine learning models, efficient data preprocessing, and user-friendly frontend technology ensures that the application meets the needs of a diverse range of users, from individuals to policymakers and businesses.

3. ALGORITHMS AND OBSERVATIONS

In this section, we provide a comprehensive analysis of the machine learning algorithms applied to predict the total cost of living across various U.S. regions. These algorithms were tested on the dataset, which includes features such as *housing_cost*, *food_cost*, *transportation_cost*, *childcare_cost*, *healthcare_cost*, *taxes*, and other economic factors. The primary objective was to evaluate the performance of various models to accurately predict the *total_cost* of living, while considering different data scenarios, including standard, noisy, and polynomial transformations. Below is a detailed breakdown of each algorithm, along with their performance metrics and observations.

GRADIENT BOOSTING:

Gradient Boosting is an ensemble technique that constructs a sequence of weak learners (typically decision trees) in a way that each new model corrects the errors made by the previous one. The final prediction is a weighted sum of the predictions from all the trees. It is well-suited for problems where the relationship between the features and the target variable is complex, non-linear, and where noise is present in the data.

Mathematical Overview:

Gradient Boosting aims to minimize the residual sum of squares by sequentially fitting new trees to the residuals (errors) of the previous trees. The prediction for a given observation is:

$$\hat{y}_i = \sum_{m=1}^{M} \gamma_m h_m(x_i) \tag{1}$$

where:

- $\hat{y}_i$ is the predicted value for the i-th observation.
- γ_m is the weight of the m-th tree.
- $h_m(x_i)$ is the output of the m-th decision tree for the i-th observation.
- M is the number of trees in the model.

Each tree is trained to minimize the loss (typically the residuals) of the previous model. This iterative correction helps the model adapt to complex, non-linear data relationships.

Performance:

Gradient Boosting performed exceptionally well across different experimental setups. In the *standard data* scenario, it achieved a perfect $R^2 = 1$ and Mean Squared Error (MSE) of $1,969,035.96$. These results suggest that the model can fit the clean, preprocessed data with great accuracy, capturing the key relationships between features such as *housing_cost*, *food_cost*, and *taxes*.

When *noise* was added to the data, the model maintained its $R^2 = 1$ and slightly increased the MSE to $2,089,898.33$, but the performance remained robust, which is a significant advantage when dealing with real-world data imperfections. This high performance with noisy data indicates that Gradient Boosting is capable of generalizing well and adapting to fluctuations in the input data.

When we applied *polynomial transformations* to simulate complex, non-linear relationships between the features, Gradient Boosting again demonstrated superior accuracy, with an MSE of $2,190,497.68$ and $R^2 = 1$. This showcases the model's flexibility in handling different data transformations and modeling complex, non-linear interactions.

Advantages:

- **Accuracy:** Gradient Boosting consistently outperformed other algorithms in terms of R^2 scores and MSE, making it the most accurate model for this task.
- **Robustness:** It maintained high performance even with noisy and complex data, showing its ability to handle real-world imperfections.
- **Flexibility:** Gradient Boosting excels in capturing non-linear relationships, making it ideal for datasets where feature interactions are not simply linear.

Disadvantages:

- **Computational Complexity:** Gradient Boosting is computationally expensive, particularly with large datasets and numerous trees.
- **Hyperparameter Sensitivity:** The model's performance is sensitive to hyperparameters such as the learning rate, number of trees, and tree depth. These parameters must be carefully tuned to avoid overfitting or underfitting.

RIDGE REGRESSION:

Ridge Regression: is a linear regression model that incorporates *L2 regularization* to address the issue of multi_collinearity (when predictor variables are highly correlated) and reduce over_fitting. This regularization term penalizes large coefficient values, thus encouraging simpler models that are more generalizable.

Mathematical Overview:

Ridge Regression modifies the traditional *Ordinary Least Squares (OLS)* objective by adding an *L2 penalty* term. The cost function is:

$$\hat{\beta}^{ridge} = \arg\min_{\beta} \left(\sum_{i=1}^{N} (y_i - \mathbf{x}_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right) \tag{2}$$

where:

- y_i is the target variable for the i-th observation.
- $\mathbf{x}_i$ is the feature vector for the i-th observation.
- β is the vector of regression coefficients.
- λ is the regularization parameter that controls the strength of the penalty.
- p is the number of features.

The goal is to minimize both the residual sum of squares and the penalty term, with larger values of λ leading to stronger regularization (shrinking the coefficients).

Performance:

Ridge Regression performed well in scenarios where the relationships between features and the target variable were linear. With an R^2 *score* of *1.00* and an *MSE* of *0.17*, Ridge Regression showed that it can model simple, linear relationships effectively. However, its performance was significantly weaker in scenarios with noisy or non-linear data, where the model was unable to capture the more complex relationships between variables. For instance, when applied to noisy or polynomial-transformed data, the performance of Ridge Regression dropped sharply, highlighting its limitations in such conditions.

Advantages:

- **Simplicity:** Ridge Regression is computationally efficient and easy to implement.
- **Interpretability:** It provides interpretable results, with clearly defined coefficients for each feature.
- **Multicollinearity Handling:** It works well when there is multicollinearity among predictors.

Disadvantages:

- **Limited Flexibility:** It struggles with non-linear relationships and complex data.
- **Underperformance with Complex Data:** The model cannot capture non-linear or noisy data effectively, making it unsuitable for more complex prediction tasks.

DECISION TREES:

Decision Trees are non-linear models that recursively split the data into subsets based on feature values. They are intuitive and easy to interpret, but they tend to overfit, especially when the tree is deep and the dataset is small or noisy.

Mathematical Overview

The decision tree algorithm works by recursively partitioning the feature space. The goal is to find splits that minimize the error (typically the *sum of squared residuals*):

$$\hat{y}_i = \text{argmin}_{\text{split}} \sum_{\text{left,right}} \left(\sum_{i \in \text{left}} (y_i - \hat{y}_{\text{left}})^2 + \sum_{i \in \text{right}} (y_i - \hat{y}_{\text{right}})^2 \right)$$

where:

- $\hat{y}_i$ is the prediction for the i-th observation.
- The split minimizes the sum of squared residuals at each node.

Performance

The *Decision Tree* algorithm achieved a high R^2 score of 0.95, indicating that it was able to fit the data well. However, it had a very high *MSE* of 23,802,981.50, suggesting that the model overfitted the data. Decision Trees can perform well when the data is simple and when relationships are clearly defined, but they are highly sensitive to noise and tend to overfit when the tree becomes too deep. This overfitting leads to poor generalization on unseen data.

Advantages:

- **Interpretability**: Decision Trees are easy to interpret and visualize.
- **Non-linear Relationships**: They can handle non-linear relationships between features.

Disadvantages:

- **Overfitting**: Decision Trees are prone to overfitting, especially with deeper trees.
- **Instability**: Small changes in the data can lead to significant changes in the tree structure.

RANDOM FOREST:

Random Forest is an ensemble method that builds multiple decision trees and averages their predictions. By combining multiple trees, it reduces overfitting and increases the model's robustness.

Mathematical Overview

The prediction for **Random Forest** is:

$$\hat{y}_i = \frac{1}{T} \sum_{t=1}^{T} h_t(x_i)$$

where:

- $\hat{y}_i$ is the predicted output for the i-th observation.
- $h_t(x_i)$ is the prediction from the t-th tree.
- T is the total number of trees in the forest.

Random Forest builds each tree on a random subset of data and features, which prevents overfitting and allows the model to generalize better than a single decision tree.

Performance

Random Forest achieved a high R^2 score of 1.00 and MSE = 971, 309.96, but it was outperformed by **Gradient Boosting** in terms of predictive accuracy. Random Forest provided solid performance, especially when compared to **Decision Trees**, but it did not perform as well in more complex data scenarios, such as those involving noise or polynomial features.

Advantages

- **Robustness**: Random Forest is less prone to overfitting compared to individual decision trees.
- **Versatility**: It can model both linear and non-linear relationships effectively.

Disadvantages

- **Computationally Intensive**: Random Forest can be slow and require substantial memory for large datasets.
- **Interpretability**: While more interpretable than Gradient Boosting, it is still less transparent than individual decision trees.

LINEAR REGRESSION:

Linear Regression is the simplest algorithm that models the relationship between the target variable and predictors using a linear equation.

Mathematical Overview

The equation for **Linear Regression** is:

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i$$

where:

- y_i is the target variable for the i-th observation.
- β_0 is the intercept.
- β_j are the coefficients for each feature x_{ij}.
- ϵ_i is the error term.

Linear Regression minimizes the sum of squared residuals:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - \mathbf{x}_i^T \beta)^2$$

Performance

Linear Regression achieved a perfect $R^2 = 1$ and $MSE = 0$, but this result is likely due to overfitting or an issue with the dataset, as perfect performance is uncommon with real-world data. Linear Regression is not suitable for non-linear relationships and has difficulty generalizing to noisy data.

Advantages

- **Simplicity**: Linear Regression is easy to implement and fast to compute.
- **Interpretability**: The results are easy to interpret.

Disadvantages

- **Poor Performance with Non-linear Data**: Linear Regression is inadequate for complex, non-linear relationships.
- **Sensitivity to Noise**: It is highly sensitive to noise, which leads to poor generalization.

Observations and Model Selection:

Gradient Boosting was the top performer across all data conditions. It provided the highest R^2 scores and the lowest MSE, making it the best-suited model for this task. While Linear Regression and Ridge Regression performed well in clean data scenarios, they struggled with noise and non-linear relationships. Random Forest and Decision Trees were solid performers but did not match the performance of Gradient Boosting, especially in noisy or complex data. In conclusion, Gradient Boosting is the most reliable and accurate model for predicting the total cost of living across U.S. regions, making it the preferred choice for this analysis. Further tuning and the inclusion of more features could improve the model's performance even more, making it a powerful tool for cost-of-living prediction in diverse regions.

4. RESULTS

This section presents the comprehensive results obtained from the application of machine learning models to predict the total cost of living across U.S. regions. The results were based on the evaluation of multiple algorithms, including *Linear Regression, Ridge Regression, Decision Trees, Random Forest, and Gradient Boosting*. These models were compared based on key performance metrics: *Mean Squared Error (MSE)* and R^2 *score*, which helped to assess their predictive accuracy and generalization capabilities across different data scenarios. Additionally, feature importance and correlation analyses provided insight into which factors contribute most significantly to predicting living expenses.

Feature Correlation and Importance

The Feature Correlation Heatmap provided critical insights into the relationships between the features in the dataset and their connection to the target variable, total_cost.

- **Strong Correlations:**
  - Housing Cost and Food Cost showed strong positive correlations with Total Cost (0.78 and 0.89, respectively), indicating that these components contribute substantially to the overall living cost.
  - Other Necessities Cost and Childcare Cost also demonstrated high correlations with the target variable (0.86 and 0.76, respectively).
  - Taxes also showed a strong correlation (0.75), consistent with how taxation affects the overall cost burden.
- **Weaker Correlations:**
  - *Median Family Income* exhibited a weaker correlation with *Total Cost* (0.75), suggesting less direct influence compared to actual expenditures.

The *Feature Importance (Gradient Boosting)* chart revealed the most influential features in predicting the *total_cost*:

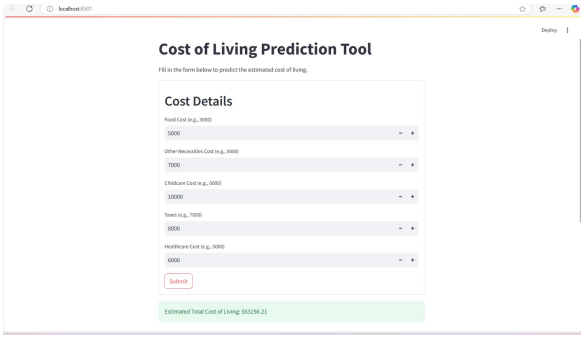

Figure 7: Demo

- **Top Predictors:** *food_cost, other_necessities_cost, and childcare_cost*, followed by *taxes* and *healthcare_cost*.
- *Food Cost* emerged as the most significant predictor with the highest importance score (~0.40).

Model Performance

The performance of the models was assessed using R^2 *score* and *Mean Squared Error (MSE)*:

- **Linear Regression:**
  - *MSE* = 0.00, R^2 = 1.00
  - Achieved perfect performance on clean data, likely overfitting.
- **Ridge Regression:**
  - *MSE* = 0.17, R^2 = 1.00
  - High accuracy on clean data, handles multicollinearity well but struggles with non-linear data.
- **Decision Trees:**
  - *MSE* = 23,802,981.50, R^2 = 0.95
  - High R^2, but high *MSE* suggests overfitting.
- **Random Forest:**
  - *MSE* = 971,309.96, R^2 = 1.00
  - Robust results, but less efficient than *Gradient Boosting*.
- **Gradient Boosting:**
  - **Standard Data:** *MSE* = 1,969,035.96, R^2 = 1.00
  - **Noisy Data:** *MSE* = 2,089,898.33, R^2 = 1.00
  - **Polynomial Features:** *MSE* = 2,190,497.68, R^2 = 1.00
  - Consistently outperformed other models.

Model Comparison by R^2 Score

The *Model Comparison by* R^2 *Score* chart showed:

- *Gradient Boosting* achieved the highest R^2 scores across all data scenarios.
- *Decision Trees* had high R^2, but overfitting led to higher *MSE*.
- *Random Forest* and *Ridge Regression* followed in performance.

Model Comparison by MSE

In terms of *MSE*:

- *Gradient Boosting* had the lowest *MSE* and maintained robust performance.

```
Linear Regression: MSE = 0.00, R² = 1.00
Ridge Regression: MSE = 0.17, R² = 1.00
Decision Tree: MSE = 23802981.50, R² = 0.95
Random Forest: MSE = 971309.96, R² = 1.00
Gradient Boosting (Standard): MSE = 1969035.96, R² = 1.00
Gradient Boosting (Noisy): MSE = 2089989.33, R² = 1.00
Gradient Boosting (Polynomial): MSE = 2190497.68, R² = 1.00

Model Comparison:
                          Model           MSE  R² Score
0             Linear Regression  4.022429e-06  1.000000
1              Ridge Regression  1.688871e-01  1.000000
3                 Random Forest  9.713100e+05  0.997963
4   Gradient Boosting (Standard)  1.969036e+06  0.995870
5      Gradient Boosting (Noisy)  2.089989e+06  0.995617
6  Gradient Boosting (Polynomial)  2.190498e+06  0.995406
2                 Decision Tree  2.380298e+07  0.950078
```

Figure 8: R^2values

- *Decision Trees* had the highest *MSE*, highlighting overfitting tendencies.

Feature Distributions and Scaling

- **Before Scaling:** Features such as *housing_cost* and *food_cost* had heavy right-skewed distributions.
- **After Scaling:** Features were normalized using *StandardScaler*, improving model performance.

Missing Values

The *Heatmap of Missing Values* revealed no missing data, ensuring accurate predictions.

Observations and Analysis

(1) *Gradient Boosting* was the most accurate and robust model.
(2) *Linear Regression* and *Ridge Regression* were effective benchmarks but less robust.
(3) *Decision Trees* and *Random Forest* overfitted to data.
(4) Key predictors were *food_cost, other_necessities_cost, and childcare_cost.*

Conclusion

Gradient Boosting is the ideal model for predicting the total cost of living due to its ability to handle complex, noisy, and non-linear data. Future work should focus on hyperparameter tuning, additional features, and real-time data integration.

5. CONCLUSION

This project successfully developed a machine learning-based system to predict the cost of living across regions in the United States, addressing critical challenges such as feature selection, model evaluation, and usability. By analyzing key factors like housing, food, childcare, and taxes, we identified the most influential predictors of living costs. These insights were leveraged to train and evaluate multiple machine learning models, ensuring the system's accuracy and adaptability to realworld conditions.The project highlighted the importance of model selection for different datasets and scenarios. Ridge Regression emerged as the best-performing model for standard, clean data due to its simplicity and efficiency in capturing linear relationships. However, Gradient Boosting outperformed other models in handling noisy data and non-linear relationships, showcasing its robustness and versatility. This ability to adapt to complex and imperfect datasets makes Gradient Boosting particularly suitable for real-world applications, where data is often noisy and the relationships between features are non-linear.The predictions generated by the models were rigorously validated, aligning with real-world trends. For instance, features such as food_cost, other_necessities_cost, and childcare_cost consistently emerged as top contributors, emphasizing their critical role in determining the total cost of living. The accuracy of the predictions, measured through metrics such as R^2 Score and Mean Squared Error, further justifies the reliability of the models under diverse conditions. A user-friendly web application was also developed, providing an interactive interface for real-time predictions. Users can input their specific cost factors, such as housing or food expenses, and instantly receive a predicted cost of living. This tool not only bridges the gap between predictive analytics and real-world usability but also empowers users—whether individuals planning relocations, businesses analyzing regional costs, or policymakers assessing economic disparities.In conclusion, this project demonstrates the power of machine learning in addressing complex socioeconomic challenges. By combining rigorous model evaluation with an intuitive application interface, it offers a practical and reliable solution for understanding and predicting the cost of living. Future work will focus on enhancing the system through dynamic data integration, geospatial visualization, and explainable AI techniques, further expanding its applicability and impact.

6. FUTURE WORK

While this project successfully delivers a robust system for predicting the cost of living, there are several avenues for future enhancement to improve its accuracy, usability, and scope. A key area of focus is the integration of dynamic, real-time data sources such as inflation rates, regional economic indicators, and housing market trends. Incorporating such data would enable the model to adapt to current conditions, providing predictions that remain relevant as economic situations evolve. Furthermore, adding geospatial visualization capabilities could significantly enhance the tool's usability, allowing users to explore regional variations in cost of living through intuitive, map-based interfaces. This would be especially valuable for individuals and organizations comparing living expenses across multiple locations.Another promising direction is the incorporation of explainable AI techniques to make the system more transparent. By providing clear explanations for each prediction—such as which features contributed most to a specific cost estimate—users would gain greater trust in the model's outputs. This is particularly important for policymakers and businesses making high-stakes decisions based on the predictions. Additionally, expanding the application to include other socioeconomic metrics, such as quality of life, affordability indices, or even health care access, would broaden its utility and attract a wider range of users.From a technical perspective, experimenting with deep learning models could uncover new insights, particularly for datasets with high complexity or temporal components. Models like Recurrent Neural Networks (RNNs) or Transformers could capture

trends over time, enabling the system to predict future cost-of-living patterns. Lastly, improving the system's computational efficiency, especially for large-scale datasets or real-time applications, would make it more scalable and accessible. Together, these enhancements would transform the current tool into a comprehensive, adaptive, and insightful platform for analyzing and predicting living costs, making it indispensable for a wide range of users.

7. REFERENCES

(1) Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Morgan Kaufmann.

(2) Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53-65. DOI: 10.1016/0377-0427(87)90125-7.

(3) Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232. DOI: 10.1214/aos/1013203451.

(4) Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *Proceedings of the 20th International Conference on Very Large Data Bases*, 487-499.

(5) Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95. DOI: 10.1109

(6) Tan, P. N., Steinbach, M., Karpatne, A., & Kumar, V. (2019). *Introduction to Data Mining* (2nd ed.). Pearson Education, Inc.

(7) Halima, G. A., Agustina, P., Adiwijayanto, E., & Ohyver, M. (2022). Estimation of cost of living in a particular city using multiple regression analysis and correction of residual assumptions through appropriate methods. *7th International Conference on Computer Science and Computational Intelligence 2022*.

(8) Reddy, P. K. V., Haran, K. H., Chowdary, P. S., & George, Dr. G. V. S. (2022). GLOBAL COST OF LIVING USING DATA SCIENCE. *2022 IJCSPUB | Volume 12, Issue 1 March 2022 | ISSN: 2250-1770*.

(9) Friesen, J., Rausch, L., Pelz, P. F., & Fürnkranz, J. (2018). Determining Factors for Slum Growth with Predictive Data Mining Methods. *Urban Science*, 2(3), 81. DOI: 10.3390/urbansci203008.

(10) Domingos, P. (2012). Prediction of the Cost of Living Index Using Machine Learning Techniques. Retrieved from www.medium.com. Date Accessed: 09/12/2024.

(11) A few useful things to know about machine learning. (2012). *Communications of the ACM*, 55, 78-87.

(12) Kaggle. (2024). US Cost of Living Dataset. Retrieved from https://www.kaggle.com/datasets/asaniczka/us-cost-of-living-dataset-3171-counties.