Group 2: Project 4

Group Members: Brittany Castro, Cooper Harris, Lois Stetson, & Thanh Vo

Data Project Proposal: Cost of living in the US

Introduction

The goal of this project is to analyze the cost of living using a machine learning model. The project aims to develop and optimize a predictive model that can estimate the cost of living based on various features. This could help clients or employers with numerous things but mostly for aid in compensation planning for future employees based on location.

Data Source

https://www.kaggle.com/datasets/asaniczka/us-cost-of-living-dataset-3171-counties?rvi=1

Problem Statement

Problem Statement: Analyzing Cost of Living with Machine Learning

The cost of living is a critical factor influencing individuals' economic decisions and well-being. Our project focuses on leveraging machine learning to conduct a comprehensive analysis of the cost of living dataset. The goal is to develop and optimize a predictive model capable of estimating the cost of living based on various features. This endeavor involves data cleaning, normalization, and standardization processes to ensure accurate and reliable predictions. Additionally, the project aims to utilize data from Spark to enhance data processing capabilities, providing a robust foundation for meaningful predictive power. The ultimate objective is to optimize the model, document the iterative changes made, and showcase the overall performance,

thereby providing valuable insights into the cost of living dynamics for informed decision-making.

Project Overview:

Data Model Implementation :

Initialization, Training, and Evaluation:

- Develop a Python script that initializes, trains, and evaluates a machine learning model.
- Utilize popular machine learning libraries such as sklearn.

Data Cleaning, Normalization, and Standardization:

- Implement data cleaning processes to handle missing values and outliers.
- Normalize and standardize the features to ensure consistent scaling.

Utilization of Data from Spark:

- Incorporate data retrieval from Spark to enhance data processing capabilities.

Meaningful Predictive Power:

- Ensure that the model demonstrates meaningful predictive power, achieving at least 75% classification accuracy or 0.80 R-squared.

Data Model Optimization :

Optimization and Evaluation Documentation:

- Document the model optimization process, showcasing iterative changes made to enhance model performance.
- Record changes and their impacts in a CSV/Excel table or within the Python script.

Overall Model Performance Display:

- Print or display the overall model performance at the end of the script.
- Provide insights into the model's effectiveness and areas for improvement.

Proposed Timeline:

Week 1: Project Initialization

- Review the existing Jupyter notebook and understand the dataset.

- Set up the project structure and initialize a version control system.

Week 2: Data Preparation and Cleaning

- Implement data cleaning processes to handle missing values and outliers.
- Normalize and standardize features.

Week 3: Model Implementation

- Develop a Python script for initializing, training, and evaluating the machine learning model.
- Incorporate data retrieval from Spark.

Week 4: Model Optimization

- Document the model optimization process, including changes and their impacts.
- Display overall model performance.

Week 5: Finalization and Documentation

- Perform final testing and validation of the model.
- Document the entire process, including code comments and explanations.

Expected Deliverables:

Python script (Cost_of_Living.py) containing the implemented model and data processing steps.
Documentation highlighting the model optimization process, including a CSV/Excel table or in-code comments.
A comprehensive README file explaining the project structure, dependencies, and instructions for running the script.

Team Members and Responsibilities:

Data Cleaning and Processing:

- Responsibilities: Implement data cleaning processes and feature engineering.

Model Implementation:

- Responsibilities: Develop the Python script for initializing, training, and evaluating the model.

Data Retrieval and Integration:

- Responsibilities: Retrieve data from Spark and integrate it into the script.

Model Optimization and Documentation:

- Responsibilities: Document the model optimization process and overall model performance.
- Team Member: [Name]

Conclusion:

This project aims to create a robust and optimized machine learning model for cost of living analysis.

Introduction: The objective is to analyze the cost of living using a machine learning model. The project aims to develop and optimize a predictive model that estimates the cost of living based on various features. This assists in compensation planning for future employees, considering location factors.

Data Source: US Cost of Living Dataset

Problem Statement: Analyzing Cost of Living with Machine Learning The project focuses on leveraging machine learning for a comprehensive cost of living analysis. It involves data cleaning, normalization, and standardization processes. Utilizing data from Spark enhances data processing capabilities, aiming for meaningful predictive power. The goal is to optimize the model, document changes, and showcase overall performance, providing insights into cost of living dynamics.

Project Overview:

Data Model Implementation:

1. Initialization, Training, and Evaluation:
    - Develop a Python script using popular ML libraries like sklearn.
2. Data Cleaning, Normalization, and Standardization:
    - Implement processes to handle missing values and outliers.
    - Normalize and standardize features for consistent scaling.
3. Utilization of Data from Spark:
    - Incorporate data retrieval from Spark to enhance processing capabilities.
4. Meaningful Predictive Power:
    - Ensure the model achieves at least 75% classification accuracy or 0.80 R-squared.

Data Model Optimization:

1. Optimization and Evaluation Documentation:
    - Document the iterative model optimization process.
    - Record changes and impacts in a CSV/Excel table or within the Python script.
2. Overall Model Performance Display:

    - Print or display overall model performance.
    - Provide insights into effectiveness and areas for improvement.

Proposed Timeline:

- Week 1: Project Initialization
    - Review existing Jupyter notebook and understand the dataset.
    - Set up project structure and initialize version control.
- Week 2: Data Preparation and Cleaning
    - Implement data cleaning processes and handle outliers.
    - Normalize and standardize features.
- Week 3: Model Implementation
    - Develop Python script for initializing, training, and evaluating the model.
    - Incorporate data retrieval from Spark.
- Week 4: Model Optimization
    - Document model optimization process and changes.
    - Display overall model performance.
- Week 5: Finalization and Documentation
    - Perform final testing and validation.
    - Document the entire process with code comments and explanations.

Expected Deliverables:

1. Python script (`Cost_of_Living.py`) with the implemented model and data processing steps.
2. Documentation highlighting the model optimization process, including a CSV/Excel table or in-code comments.
3. Comprehensive README file explaining project structure, dependencies, and instructions for running the script.

Team Members and Responsibilities:

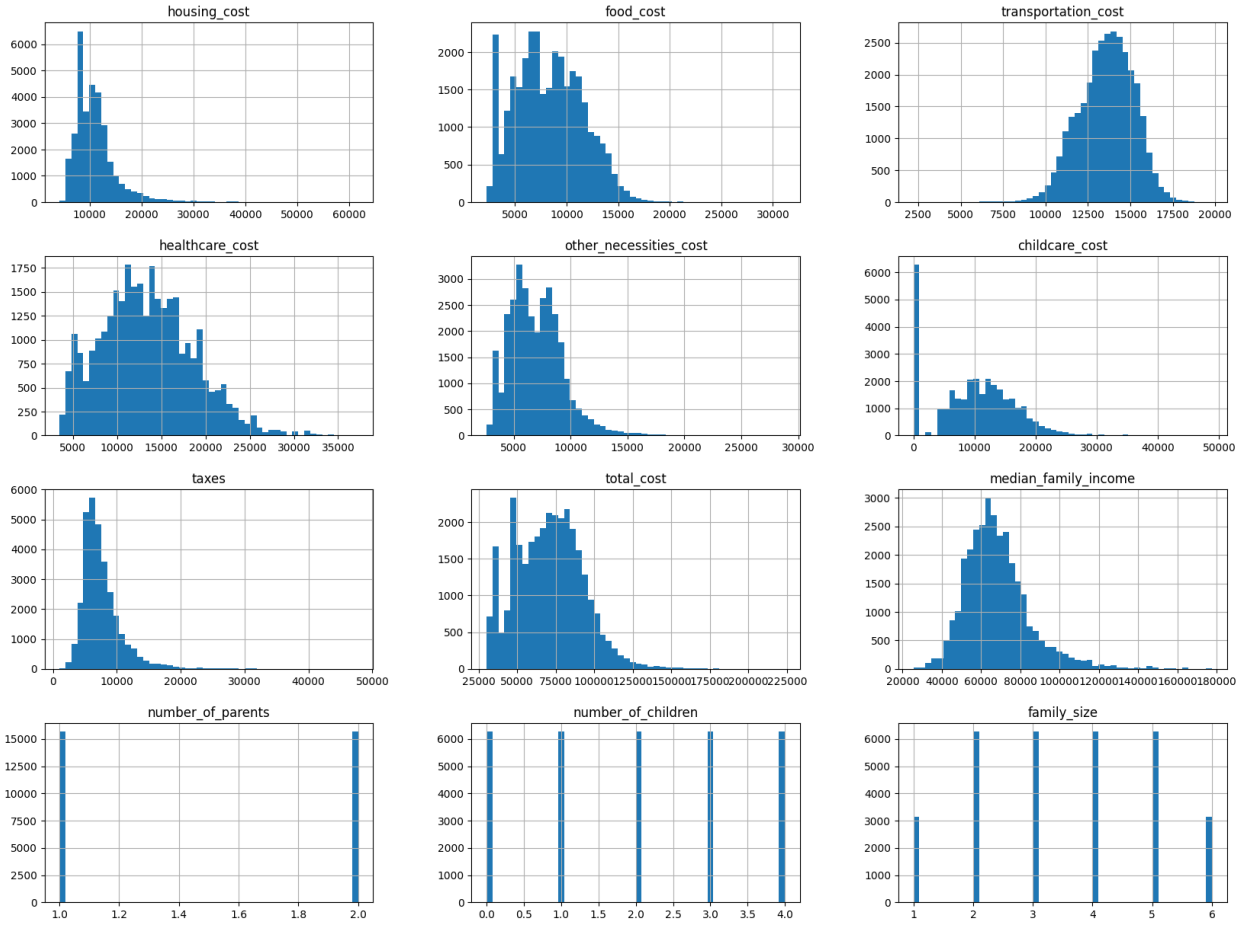
1. Data Cleaning and Processing:
    - Responsibilities: Implement data cleaning processes and feature engineering.
2. Model Implementation:
    - Responsibilities: Develop Python script for initializing, training, and evaluating the model.
3. Data Retrieval and Integration:
    - Responsibilities: Retrieve data from Spark and integrate it into the script.
4. Model Optimization and Documentation:

- Responsibilities: Document the model optimization process and overall model performance.

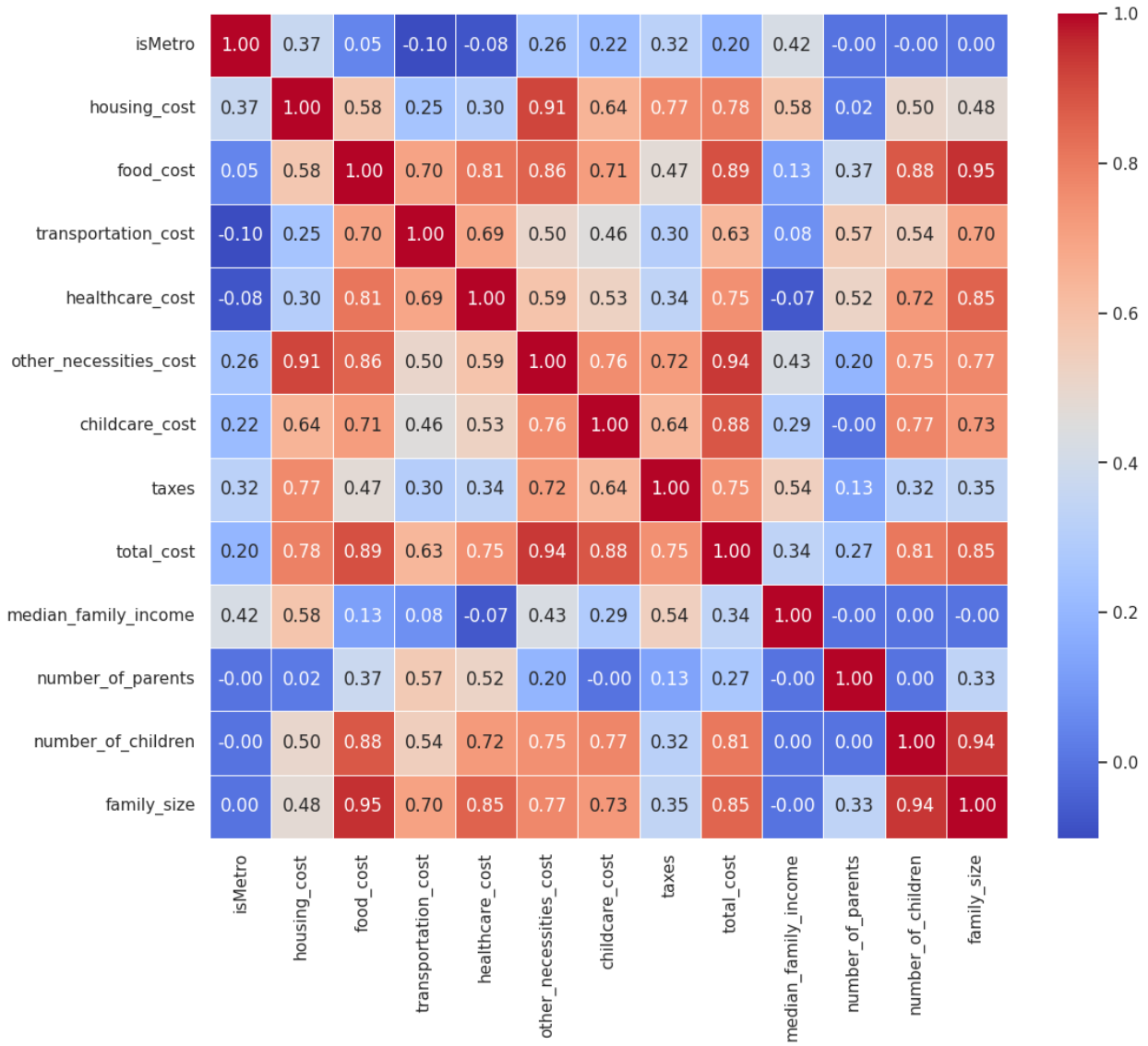
Conclusion: This project aims to create a robust machine learning model for cost of living analysis. The proposed timeline and responsibilities ensure a systematic approach to achieve the project objectives.

Our initial cleaning, we started by dropping unnecessary columns, looking for missing values and dropping those rows with missing values, and also separating out the family member count column to create three new columns one to represent parents in the home, children in the home, and an overall family size amount column. Then we moved into exploring the dataset.

Histogram Insights: • Peaks and patterns in histograms can indicate central tendencies and variations in the data. • Skewness or symmetry in distributions may be observed. • Outliers or unusual patterns might be identified. Key Takeaways: • Histograms are valuable for initial data exploration and understanding the spread of variables. • They help identify data patterns, potential outliers, and insights into the data's central tendencies. Utilizing histograms is an essential step in the exploratory data analysis process, offering a visual overview of the dataset's distribution characteristics.



Correlation Matrix Using sns in python, we created a correlation matrix In the correlation matrix, The colors indicate the strength and direction of correlations, with warmer colors (reds) representing positive correlations and cooler colors (blues) representing negative correlations. The numerical annotations provide precise correlation values for each pair of variables. This visualization is crucial for identifying patterns and dependencies in the data.



Correlation Plot Presentation Explanation: • Objective: The objective of this visualization is to explore the relationships and distributions between selected features in the dataset.

• Pair Plot: The pair plot provides a quick overview of the pairwise relationships between features. Each scatterplot shows how two variables interact with each other.

• Diagonal Plots (KDE): The diagonal plots represent the distribution of individual features using KDE. This helps in understanding the shape and spread of each variable's values.

• Off-Diagonal Plots: These scatterplots show the relationships between pairs of features. Patterns, trends, or clusters in these plots can indicate potential correlations or associations. Insights:

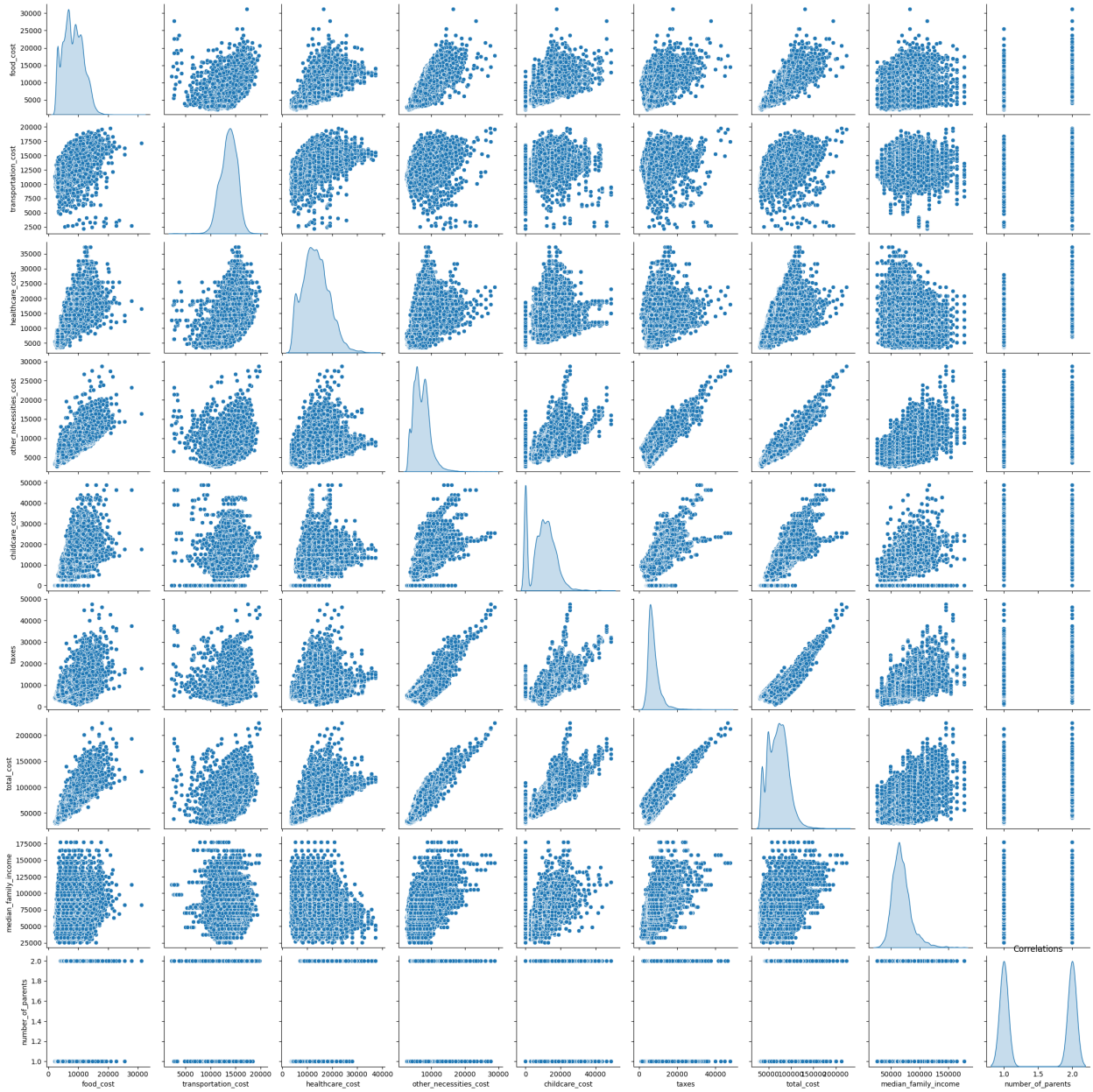
• Correlation Assessment: Look for patterns in the off-diagonal scatterplots to identify potential correlations between variables.

• Distribution Understanding: Diagonal KDE plots provide insights into the distribution of each feature, helping to understand the data's characteristics.

Key Points:

• Title Clarity: The title 'Correlations' suggests a focus on relationships between variables.

• Visualizing Data Patterns: Pair plots are effective for visualizing multivariate patterns and can aid in feature selection and understanding data distributions.



Machine Learning Model

In [108…

```
#Seperate into Train and Test DataFrames

df = df.sample(frac=1, random_state=2)
train_df = df[:25000]
train_df = train_df.reset_index(drop=True)
test_df = df[25000:]
test_df = test_df.reset_index(drop=True)
```

In [109…

```
train_df
```

Out[109…

	housing_cost	food_cost	transportation_cost	healthcare_cost	other_necessities_cost	childcare_cost	taxes	total_cost	m
0	8203.75632	5858.48772	12332.94960	9746.69976	5095.30560	0.00000	5705.29284	46942.4916	
1	5664.00000	3039.91200	11556.09672	5302.79004	3153.77052	0.00000	5463.14832	34179.7176	
2	10452.00000	10359.47388	15339.26640	24921.84960	7540.81788	13162.81080	10267.82808	92044.0488	
3	9672.00000	8756.77512	12316.96920	12859.02360	6677.47188	8847.39624	4589.69124	63719.3256	
4	6516.00000	5573.17200	13611.82920	9901.37988	4380.38376	0.00000	6304.81644	46287.5796	
…	…	…	…	…	…	…	…	…	
24995	8697.91188	8874.71412	15134.80560	16420.99080	6367.25532	13677.74280	8574.69504	77748.1176	
24996	16632.00000	3963.29772	8806.32720	6328.54032	7462.48824	0.00000	9472.09788	52664.7540	
24997	9876.00000	10139.83884	14389.33920	24000.32760	7252.52784	9999.14580	7825.91376	83483.0916	
24998	6036.00000	3154.03824	11137.67868	5898.69000	3329.91360	0.00000	5398.60140	34954.9224	
24999	14004.00000	12548.04600	14912.44320	13108.06320	9620.85276	13981.54200	9067.35132	87242.2968	

25000 rows × 12 columns

In [110…

```
test_df
```

Out[110…

	housing_cost	food_cost	transportation_cost	healthcare_cost	other_necessities_cost	childcare_cost	taxes	total_cost	m
0	9240.000	8424.98220	14642.58720	19499.00640	6400.71972	15842.90520	9384.11724	83434.3188	
1	11424.000	9534.32148	14107.73280	16239.61560	7594.02684	18368.77560	10016.06256	87284.5368	

In [103…
```
y_train = train_df['housing_cost'].to_numpy()
y_test = test_df['housing_cost'].to_numpy()

y_train.shape, y_test.shape
```

Out[103… ((25000,), (6420,))

In [129…
```
# Get error of baseline model


average_housing_cost = train_df['housing_cost'].mean()
baseline_model_test_predictions = [average_housing_cost] * len(test_df)

mean_absolute_error(baseline_model_test_predictions, y_test)
```

Out[129… 2838.302903254783

In [132…
```
train_df[['housing_cost' , 'median_family_income', 'food_cost']].corr()
```

Out[132…

	housing_cost	median_family_income	food_cost
housing_cost	1.000000	0.583345	0.578886
median_family_income	0.583345	1.000000	0.128180
food_cost	0.578886	0.128180	1.000000

In [134…
```
x_train_3 = train_df[['housing_cost' , 'median_family_income', 'food_cost']].to_numpy()

x_train_3.shape
```

Out[134… (25000, 3)

In [137…
```
x_test_3 = test_df[['housing_cost' , 'median_family_income', 'food_cost']].to_numpy()

x_test_3.shape
```

Out[137… (6420, 3)

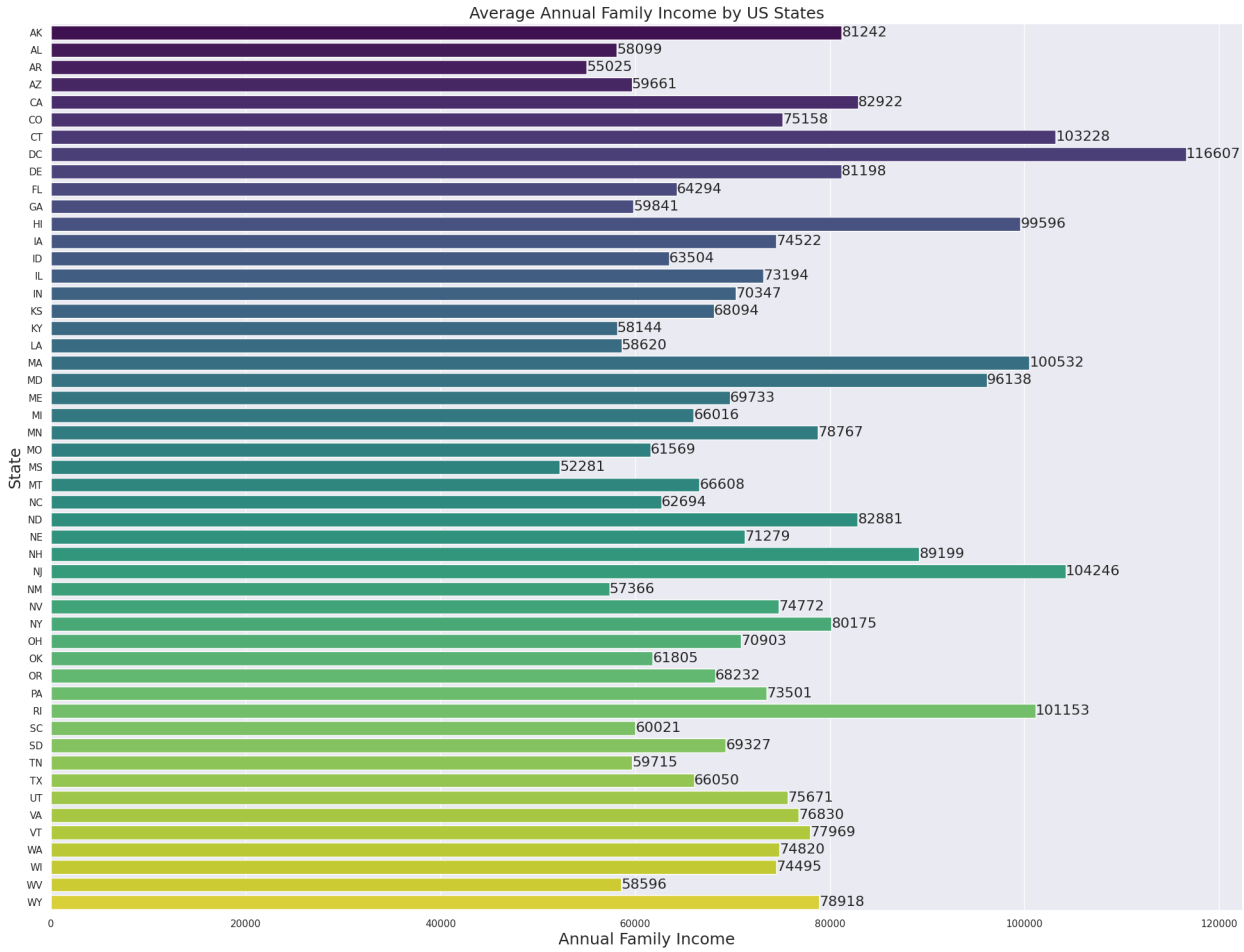
In [138…
```
forest_base = RandomForestRegressor(n_estimators=50, max_depth=5).fit(X_train_3, y_train)
forest_base_test_predictions = forest_base.predict(X_test_3)
mean_absolute_error(y_test, forest_base_test_predictions)
```

```
train_df.corr()
```

```
<ipython-input-38-a7a80d20a44e>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it wi
  train_df.corr()
```

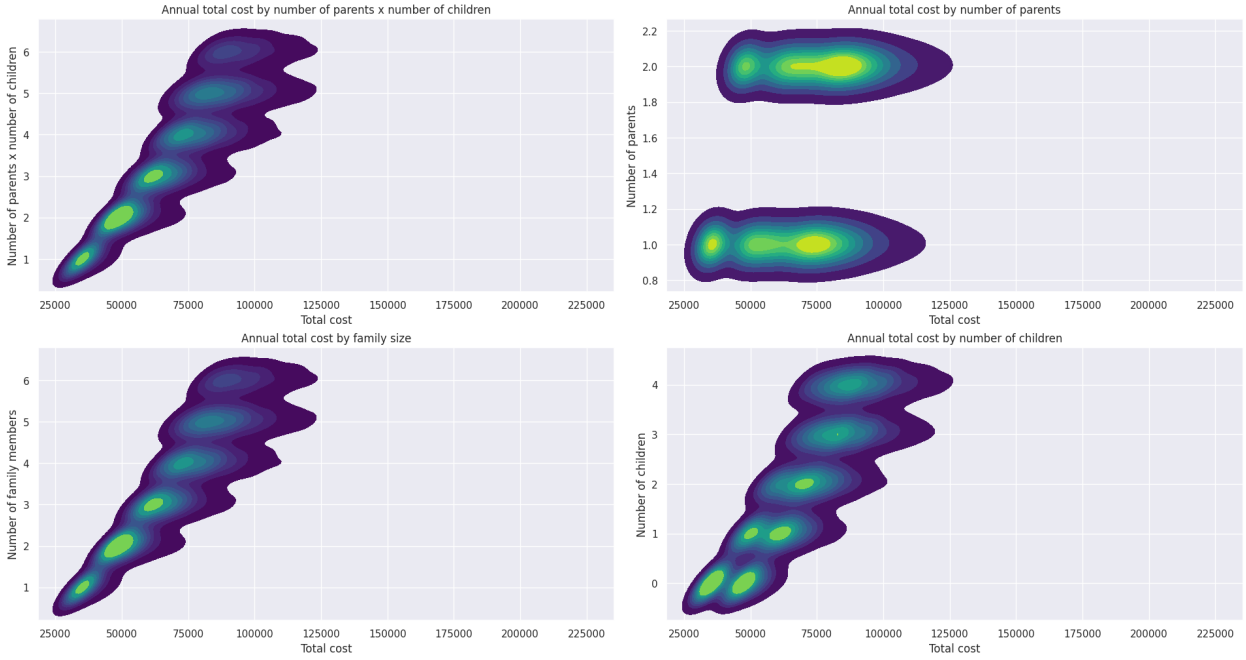
	housing_cost	food_cost	transportation_cost	healthcare_cost	other_necessities_cost	childcare_cost	taxes	total_cos
housing_cost	1.000000	0.579418	0.252193	0.300147	0.915794	0.647399	0.768197	0.7870
food_cost	0.579418	1.000000	0.693363	0.812770	0.857983	0.714347	0.477231	0.8940
transportation_cost	0.252193	0.693363	1.000000	0.684671	0.500636	0.452246	0.296469	0.6299
healthcare_cost	0.300147	0.812770	0.684671	1.000000	0.589704	0.527517	0.336031	0.7473
other_necessities_cost	0.915794	0.857983	0.500636	0.589704	1.000000	0.760058	0.719341	0.9366
childcare_cost	0.647399	0.714347	0.452246	0.527517	0.760058	1.000000	0.639605	0.8812
taxes	0.768197	0.477231	0.296469	0.336031	0.719341	0.639605	1.000000	0.7509
total_cost	0.787099	0.894022	0.629923	0.747383	0.936649	0.881268	0.750916	1.0000
median_family_income	0.583269	0.128799	0.078398	-0.068985	0.431081	0.291760	0.537242	0.3404
number_of_parents	0.017502	0.369124	0.561555	0.511806	0.192935	-0.001324	0.135053	0.2635
number_of_children	0.499221	0.876251	0.539097	0.722961	0.746455	0.771861	0.319214	0.8086
family_size	0.477206	0.950418	0.696219	0.853243	0.769130	0.728361	0.346427	0.8514

Analysis & Visualizations

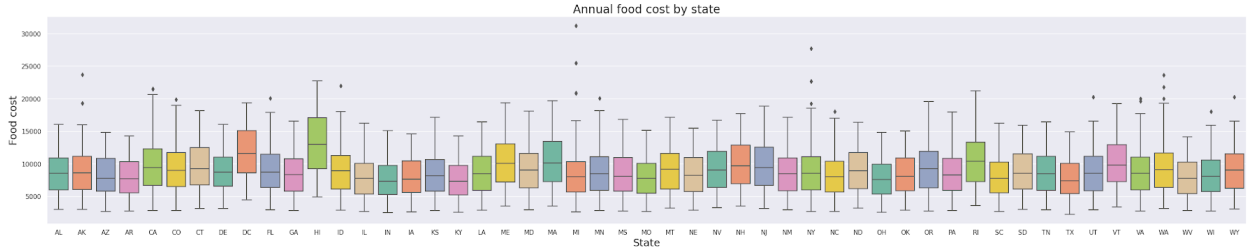


This shows a horizontal bar plot is created to visualize the average annual family income across different US states. The resulting visualization is a horizontal bar plot that effectively communicates the average annual family income for each US state. The annotations provide specific values, and the color palette enhances the overall aesthetics of the plot. Overall, DC has the highest Average Annual Income and

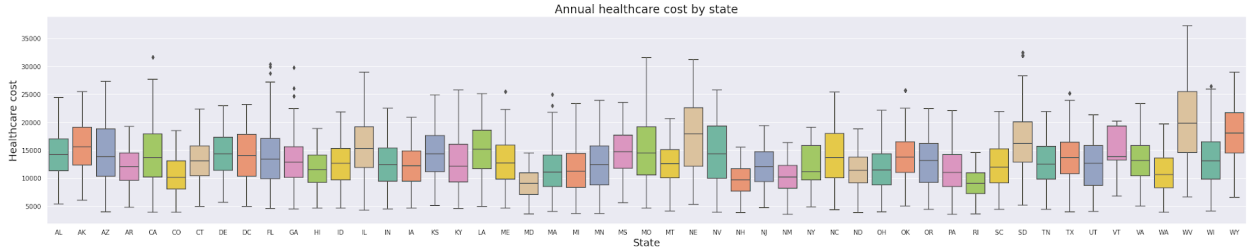
Mississippi has the lowest Average Annual Income.

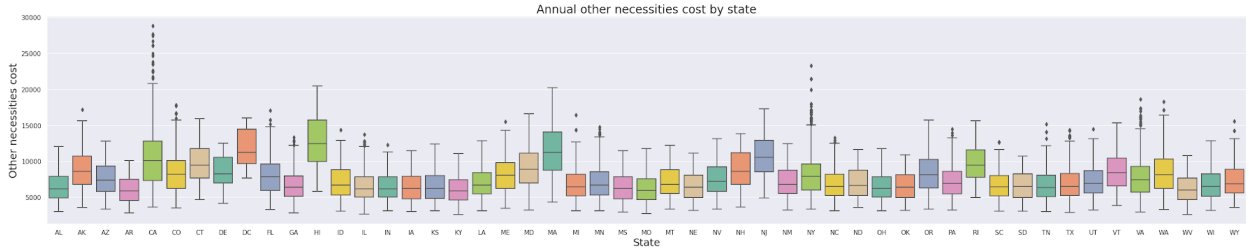


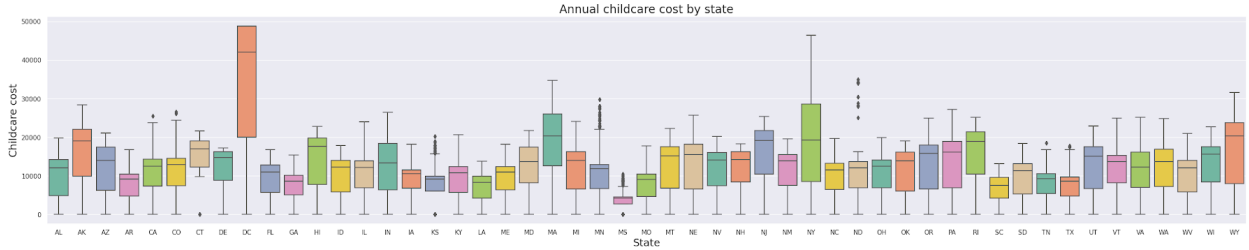
Using Python's Matplotlib and Seaborn libraries to create a set of four area plots. The plots visualize the distribution of the 'total_cost' variable with respect to different features ('family_size', 'number_of_parents', 'family_size', 'number_of_children'). The goal is to visually represent the distribution of 'total_cost' concerning different features, providing insights into the relationship between these variables. The color palette and the filled KDE plots enhance the visualization aesthetics.

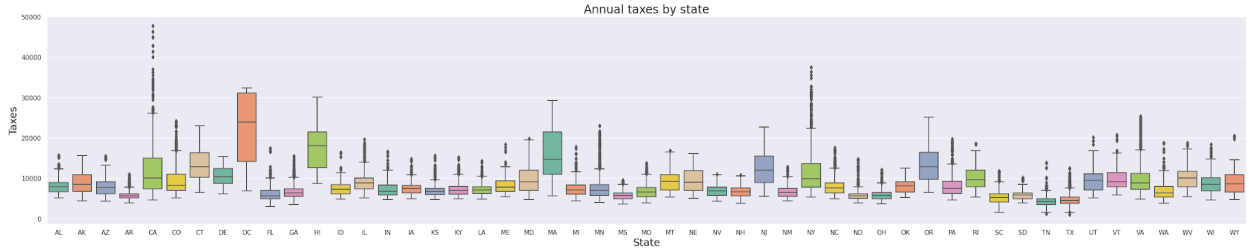

Annual food cost by state
Food cost
State

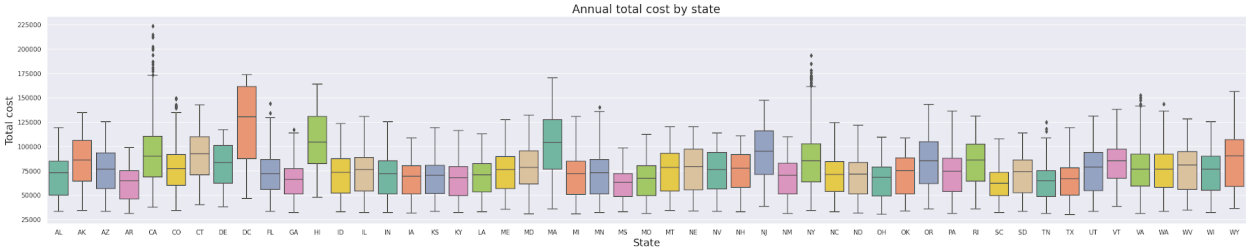

Annual transportation cost by state
Transportation cost
State


Annual healthcare cost by state
Healthcare cost
State


Annual other necessities cost by state
Other necessities cost
State


Annual childcare cost by state
Childcare cost
State


Annual taxes by state
Taxes
State


Annual total cost by state
Total cost
State

A set of boxplots is created to visually represent the distribution of annual values for different features across different states. The visualizations provide insights into the distribution of annual values for various features, allowing for a comparative analysis across different states. Each subplot represents a different feature, and the boxplots within show the spread of values within each state.

Our comprehensive analysis of the US Family Budget Dataset, a valuable resource offering detailed insights into the intricacies of the cost of living across diverse US counties. Derived from the Family Budget Calculator by the Economic Policy Institute (EPI), this dataset meticulously estimates the cost of living for ten distinct family types, ranging from single adults to families with up to four children, across all 1877 counties and metro areas in the United States.

Our exploration goes beyond the surface, encompassing the development and exploration of a sophisticated machine learning model. This model represents a significant enhancement, providing predictive insights into the dynamics of cost of living based on various features.

Aligned with our objectives, we engage in tasks that delve into the comparison of family budgets with federal poverty lines, uncovering the economic challenges faced by diverse family types. Our aim is to unravel the intricacies of the cost of living, offering valuable insights into real-world scenarios.

One notable focus is the examination of the affordability of essential commodities such as housing, food, transportation, healthcare, and childcare across different counties. We are eager to share how family income correlates with the overall cost of living and whether specific counties impose higher costs on larger families.

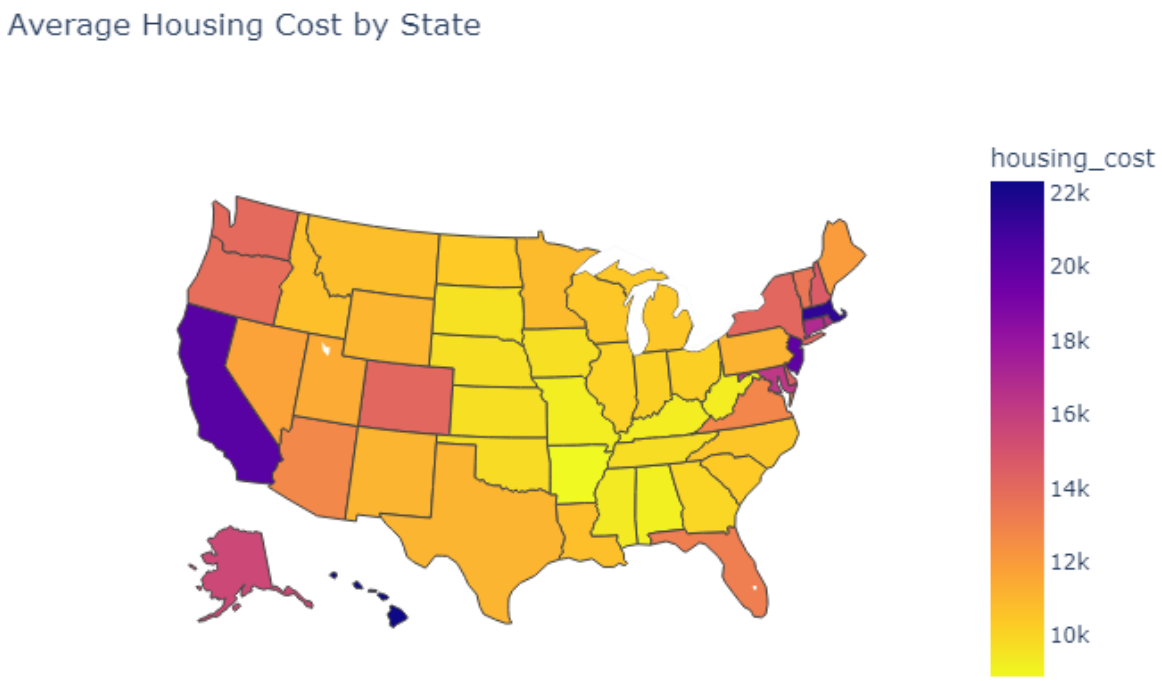
Our visualizations promise to be a captivating journey across states and major cities, providing a nuanced portrayal of the diverse landscape of living costs. In addition, we will address the crucial question of whether certain counties are affordable for families of varying sizes and compositions.

As we embark on this informative expedition through the economic landscape of US counties, we aim not only to present data but also to unfold compelling narratives reflecting the living standards and economic security experienced by families across the nation.

Moreover, our project is anchored by a clear problem statement and objective: to analyze the cost of living through the lens of a machine learning model. This involves the implementation of data cleaning, normalization, and standardization processes, leading to the development and optimization of a predictive model. Our model aims to

achieve a meaningful level of predictive power, showcasing its efficacy in estimating the cost of living based on various features.

Fasten your seatbelts for an enlightening voyage into the intersection of data and real-world insights. Let's delve into the rich tapestry of the economic landscape and unravel the stories that shape the lives of families across the nation.



The provided choropleth map visualizes the average housing cost by state in the United States. The color intensity on the map represents the variation in housing costs across different states. The color scale enhances the visual distinction between states with higher and lower average housing costs.
Key Findings:

- The map reveals geographical patterns in housing costs, with states exhibiting diverse cost distributions.
- Darker shades indicate higher average housing costs, while lighter shades represent lower costs.
- States with high housing costs are prominently highlighted, offering insights into regions where housing affordability may be a significant concern.

- Hovering over a state provides specific information on its average housing cost, allowing for a detailed exploration of individual states.

Overall, this visualization aids in understanding the spatial distribution of housing costs across the U.S., enabling stakeholders/clients to identify areas with distinct affordability challenges and formulate targeted strategies based on regional variations.