Exploratory Data Analysis on

Insurance Premium Prediction

by

Group 14

Ayush Chaudhari
ID: 202201517
Course: BTech(ICT)

Kishan Pansuriya
ID: 202201504
Course: BTech(ICT)

Mihir Bhavsar
ID: 20241079
Course: MTech(ICT)

Course Code: IT 462
Semester: Autumn 2024

Under the guidance of

Dr. Gopinath Panda

Dhirubhai Ambani Institute of Information and Communication Technology

December 2, 2024

Acknowledgment

We would like to express our heartfelt appreciation for your unwavering support and guidance throughout the course of our project titled ”Insurance.” Your expertise and mentorship have been invaluable in helping us achieve the objectives of this project, and we are genuinely thankful for the time and effort you invested to ensure its success.

It has been a privilege to work under your guidance. Your insights, encouragement, and generosity in sharing your knowledge have significantly contributed to the depth and quality of our work. Your constructive feedback and thoughtful suggestions have played a critical role in overcoming challenges and expanding our understanding of the subject matter.

Additionally, we are deeply grateful to the entire team at DAIICT for creating an environment that encourages collaboration and innovation. The resources and facilities provided have been essential in facilitating our research and analysis, enabling us to achieve meaningful and impactful results.

We also want to extend our thanks to our peers and colleagues for their constant support and camaraderie throughout this journey. Their invaluable input and motivation have been a continuous source of inspiration.

This project has been an incredible learning experience, and we are confident that the knowledge and skills we have gained will serve as a strong foundation for our future endeavors.

Once again, we sincerely thank you for your guidance and for believing in our potential. Your mentorship has been immeasurable, and we are truly grateful for the opportunity to work alongside you.

Sincerely,
Ayush Chaudhari - 202201517
Kishan Pansuriya - 20220104
Mihir Bhavsar - 2022011079

DECLARATION

We, the students of Group 14, with roll numbers 202201517, 202201504, and 202411079, hereby declare that the work presented in this report for the EDA project is entirely our own. It has not been submitted for any other academic degree, and all references used in this report have been duly cited.

We acknowledge that the dataset used in this project was sourced from the Group 14 website, and we confirm that we have adhered to the terms and conditions specified on the website for utilizing the dataset. We assure that the dataset is accurate and true to the best of our knowledge.

We further confirm that we have not sought external assistance in the completion of this project, other than the guidance provided by our mentor, Prof. Gopinath Panda. We also declare that there are no conflicts of interest related to this EDA project.

We sign this declaration and confirm the submission of this report on 2nd December 2024.

Ayush Chaudhari
ID: 202201517
Course: BTech(ICT)

Kishan Pansuriya
ID: 202201504
Course: BTech(IT)

Mihir Bhavsar
ID: 202411079
Course: MTech(ICT)

CERTIFICATE

This is to certify that Group 14, comprising Ayush Chaudhari, Mihir Bhavsar, and Kishan Pansuriya, has successfully completed an exploratory data analysis (EDA) project on the Insurance dataset, which was obtained from Kaggle.

The EDA project presented by Group 14 is their original work and has been completed under the guidance of the course instructor, Prof. Gopinath Panda, who has provided support and guidance throughout the project. The project is based on a thorough analysis of the Insurance dataset, and the results presented in the report are based on the data obtained from the dataset.

This certificate is issued to recognize the successful completion of the EDA project on the Insurance dataset, which demonstrates the analytical skills and knowledge of the students of Group 14 in the field of data analysis.

Signed,
Dr. Gopinath Panda,
IT 462 Course Instructor
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, Gujarat, INDIA.
December 2, 2024

Contents

List of Figures

List of Tables

Abstract

This exploratory data analysis (EDA) project focuses on the prediction of insurance premiums based on various demographic, financial, and lifestyle factors. The dataset utilized in this study consists of features such as age, gender, annual income, credit score, smoking status, and more, with the target variable being the insurance premium amount. The analysis aims to uncover patterns and relationships between these factors and how they influence insurance premiums. Through visualizations like scatter plots and histograms, as well as statistical summaries, the project examines the distribution and trends of the premium amounts across different categories. Furthermore, advanced modeling techniques, including regression and machine learning algorithms, are employed to predict premium amounts and explore the key drivers behind premium variations, offering insights into how these factors shape insurance pricing.

Chapter 1. Introduction

1.1 Project Idea

Insurance is a vital aspect of financial planning and risk management, and determining premium amounts is crucial for achieving a balance between affordability for customers and profitability for insurance companies. This project focuses on using machine learning techniques to predict **Premium Amounts** based on customer demographics, policy features, and lifestyle indicators. The project leverages an insurance premium prediction dataset for analysis and model development.

1.2 Objective

Goal: To develop a machine learning model that accurately predicts **Premium Amounts**, providing insights into customer behavior and enabling data-driven decisions in premium pricing and policy management.

Objectives:

1. Perform exploratory data analysis to uncover trends and relationships in the dataset.
2. Engineer features and preprocess data for model training and evaluation.
3. Train machine learning models like Random Forest and Gradient Boosting to predict premiums, and evaluate their performance using Root Mean Squared Logarithmic Error (RMSLE).
4. Provide actionable insights to improve dynamic pricing strategies and customer retention.

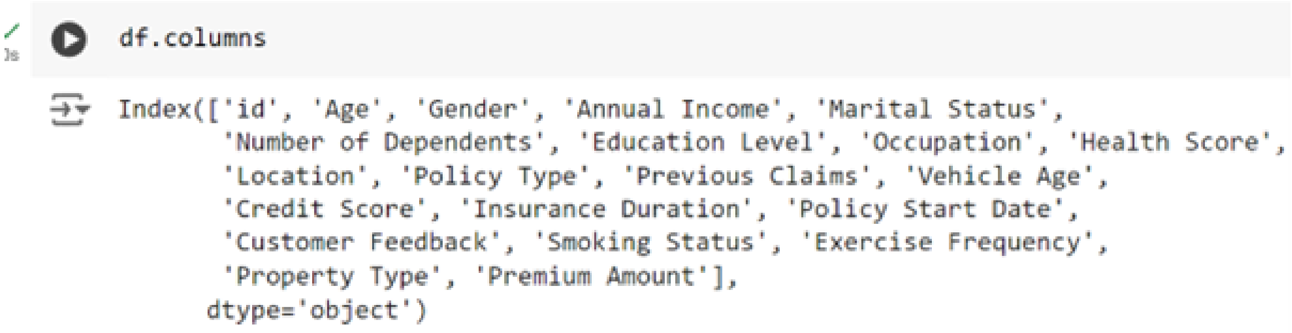
1.3 Dataset Description

The dataset provides customer demographic, lifestyle, and policy-related information for predicting insurance premium amounts. It contains the following features:

- **Demographics:** Age, Gender, Marital Status, Education Level, and Location (Urban, Rural, Suburban).
- **Financial Metrics:** Annual Income, Credit Score, and Number of Dependents.
- **Policy Features:** Insurance Duration, Property Type, Vehicle Age, and Policy Start Date.
- **Lifestyle Factors:** Health Score, Smoking Status, and Exercise Frequency.

- **Target Variable:** Premium Amount (continuous).

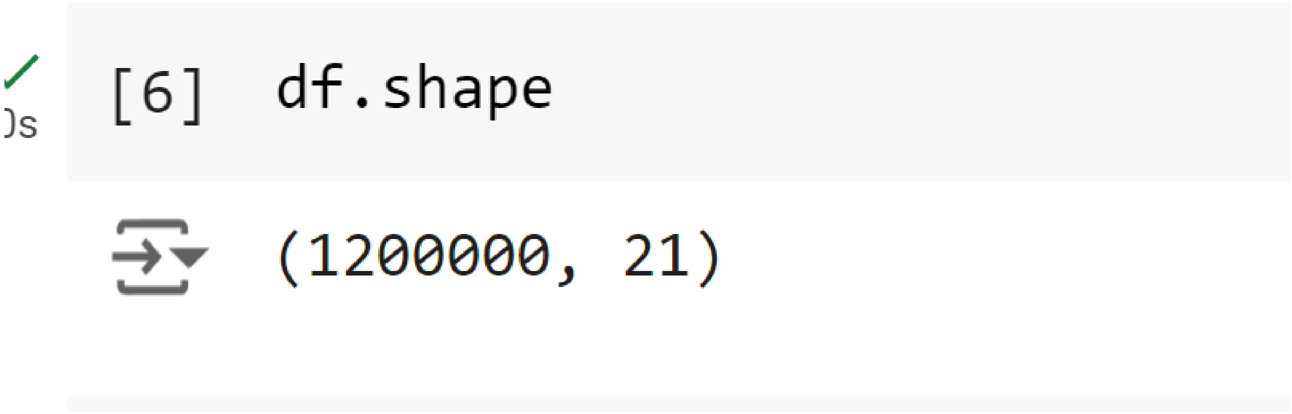
The dataset is synthetic and closely mirrors real-world insurance data, with missing values and categorical features, making it suitable for data preprocessing and machine learning model development.



```
df.columns
```

```
Index(['id', 'Age', 'Gender', 'Annual Income', 'Marital Status',
       'Number of Dependents', 'Education Level', 'Occupation', 'Health Score',
       'Location', 'Policy Type', 'Previous Claims', 'Vehicle Age',
       'Credit Score', 'Insurance Duration', 'Policy Start Date',
       'Customer Feedback', 'Smoking Status', 'Exercise Frequency',
       'Property Type', 'Premium Amount'],
      dtype='object')
```

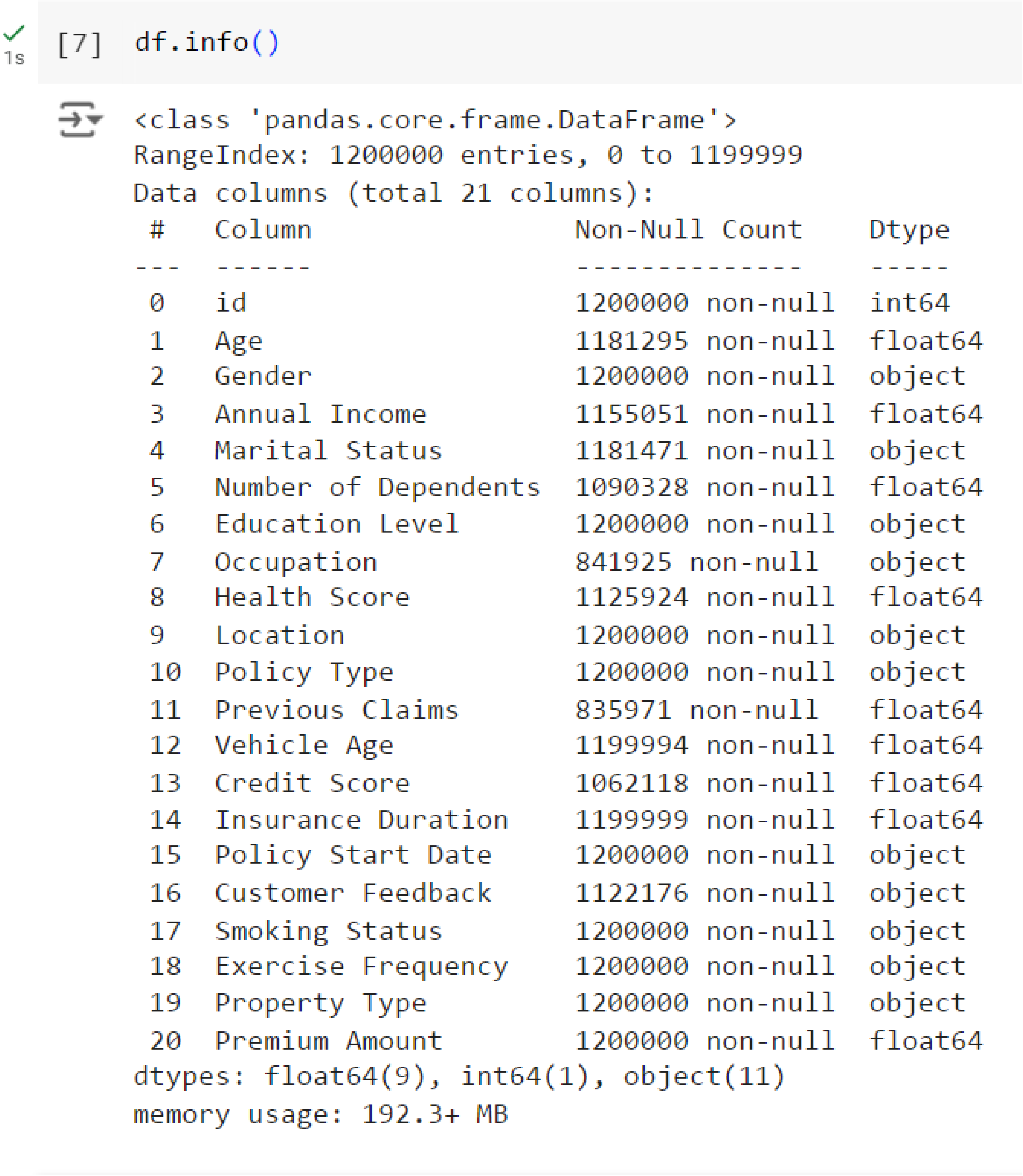
Figure 1.1: df.columns is an attribute in pandas that returns an index object containing the column labels (names) of a DataFrame.



```
[6] df.shape
```

```
(1200000, 21)
```

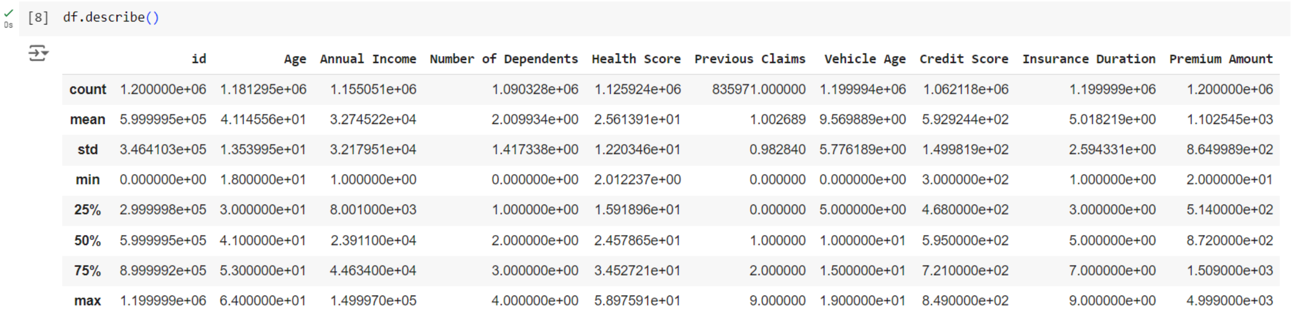
Figure 1.2: df.shape: Returns a tuple representing the number of rows and columns in the DataFrame.



```
[7] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1200000 entries, 0 to 1199999
Data columns (total 21 columns):
 #   Column                Non-Null Count    Dtype  
---  ------                --------------    -----  
 0   id                    1200000 non-null  int64  
 1   Age                   1181295 non-null  float64
 2   Gender                1200000 non-null  object 
 3   Annual Income         1155051 non-null  float64
 4   Marital Status        1181471 non-null  object 
 5   Number of Dependents  1090328 non-null  float64
 6   Education Level       1200000 non-null  object 
 7   Occupation            841925 non-null   object 
 8   Health Score          1125924 non-null  float64
 9   Location              1200000 non-null  object 
 10  Policy Type           1200000 non-null  object 
 11  Previous Claims       835971 non-null   float64
 12  Vehicle Age           1199994 non-null  float64
 13  Credit Score          1062118 non-null  float64
 14  Insurance Duration    1199999 non-null  float64
 15  Policy Start Date     1200000 non-null  object 
 16  Customer Feedback     1122176 non-null  object 
 17  Smoking Status        1200000 non-null  object 
 18  Exercise Frequency    1200000 non-null  object 
 19  Property Type         1200000 non-null  object 
 20  Premium Amount        1200000 non-null  float64
dtypes: float64(9), int64(1), object(11)
memory usage: 192.3+ MB
```

Figure 1.3: df.info(): Provides a summary of the DataFrame, including the data types, non-null values, and memory usage



```
[8] df.describe()
```

	id	Age	Annual Income	Number of Dependents	Health Score	Previous Claims	Vehicle Age	Credit Score	Insurance Duration	Premium Amount
count	1.200000e+06	1.181295e+06	1.155051e+06	1.090328e+06	1.125924e+06	835971.000000	1.199994e+06	1.062118e+06	1.199999e+06	1.200000e+06
mean	5.999995e+05	4.114556e+01	3.274522e+04	2.009934e+00	2.561391e+01	1.002689	9.569889e+00	5.929244e+02	5.018219e+00	1.102545e+03
std	3.464103e+05	1.353995e+01	3.217951e+04	1.417338e+00	1.220346e+01	0.982840	5.776189e+00	1.499819e+02	2.594331e+00	8.649989e+02
min	0.000000e+00	1.800000e+01	1.000000e+00	0.000000e+00	2.012237e+00	0.000000	0.000000e+00	3.000000e+02	1.000000e+00	2.000000e+01
25%	2.999998e+05	3.000000e+01	8.001000e+03	1.000000e+00	1.591896e+01	0.000000	5.000000e+00	4.680000e+02	3.000000e+00	5.140000e+02
50%	5.999995e+05	4.100000e+01	2.391100e+04	2.000000e+00	2.457865e+01	1.000000	1.000000e+01	5.950000e+02	5.000000e+00	8.720000e+02
75%	8.999992e+05	5.300000e+01	4.463400e+04	3.000000e+00	3.452721e+01	2.000000	1.500000e+01	7.210000e+02	7.000000e+00	1.509000e+03
max	1.199999e+06	6.400000e+01	1.499970e+05	4.000000e+00	5.897591e+01	9.000000	1.900000e+01	8.490000e+02	9.000000e+00	4.999000e+03

Figure 1.4: df.describe(): Generates descriptive statistics (e.g., mean, std, min, max) for numerical columns in the DataFrame.

1.4 Data Collection

The dataset used in this analysis is sourced from a Kaggle competition titled **Insurance Premium Prediction**. It provides data on various features of customers and their corresponding insurance premium amounts. The dataset consists of two primary files:

- **train.csv**: This file contains the training data, including various customer demographics, financial information, policy features, and lifestyle factors, with the target variable *Premium Amount* (the insurance premium) being provided.
- **test.csv**: This file contains the test data, where the goal is to predict the *Premium Amount* for each customer based on the same features as in the training data.

The dataset includes several customer-specific features such as age, gender, marital status, education level, annual income, credit score, and lifestyle factors, all of which are used to predict the insurance premium.

This data is made publicly available through the Kaggle platform, allowing individuals to use it for predictive modeling and analysis. By utilizing the information provided in these datasets, the project aims to explore relationships between the features and the insurance premium and develop an effective predictive model.

1.5 Packages Required

The following Python libraries were used for data analysis, modeling, and visualization:

- **NumPy** (`import numpy as np`):
  - A core library for numerical computing in Python.
  - Provides support for large multi-dimensional arrays and matrices along with mathematical functions to operate on them efficiently.
- **Pandas** (`import pandas as pd`):
  - An essential library for data manipulation and analysis.
  - Offers data structures such as DataFrames for handling and analyzing structured data with ease.
- **Matplotlib** (`import matplotlib.pyplot as plt`):
  - A powerful library for creating static, animated, and interactive visualizations.
  - Supports a variety of plot types, including line plots, scatter plots, histograms, and more.
- **Seaborn** (`import seaborn as sns`):
  - Built on top of Matplotlib, it simplifies the creation of statistical plots.
  - Provides advanced visualization techniques like heatmaps, pair plots, and box plots, ideal for exploring relationships in data.

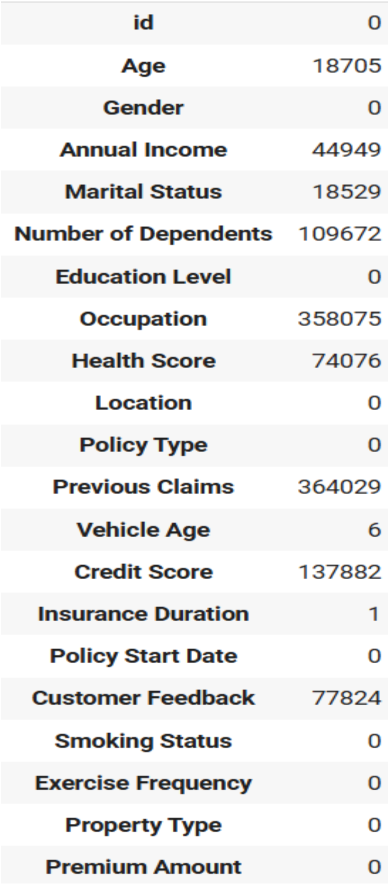
- **Plotly** (`import plotly.express as px`):
  - A library for creating interactive and dynamic visualizations.
  - Allows for the creation of highly customizable plots, such as interactive scatter plots and line charts, useful for data exploration.
- **Scikit-learn** (`import from sklearn.model_selection, StandardScaler, LinearRegression, DecisionTreeRegressor, RandomForestRegressor, mean_absolute_error, mean_squared_erro r2_score`):
  - A comprehensive library for machine learning tasks.
  - Provides tools for model training, data preprocessing, evaluation, and optimization.
    * `train_test_split`: For splitting datasets into training and test sets.
    * `StandardScaler`: For feature scaling, removing mean and scaling to unit variance.
    * `LinearRegression`, `DecisionTreeRegressor`, `RandomForestRegressor`: Algorithms for regression tasks.
    * `mean_absolute_error`, `mean_squared_error`, `r2_score`: Metrics for evaluating model performance.
- **Missingno** (`import missingno as msno`):
  - A library for visualizing missing data in a dataset.
  - Provides several functions to display missing data patterns, including `msno.bar()` and `msno.matrix()`, useful for understanding the extent and distribution of missing values.
- **Plotly Graph Objects** (`import plotly.graph_objects as go`):
  - Part of Plotly, providing tools for building complex, interactive, and highly customizable visualizations.
  - Useful for creating advanced plots such as 3D plots, subplots, and interactive dashboards.
- **Statsmodels** (`import statsmodels.api as sm`):
  - A library for statistical modeling.
  - Used for running regression models, hypothesis testing, and statistical analysis on data.

Chapter 2. Data Cleaning

Data cleaning is an important step to make sure the data is accurate and ready for analysis. It involves fixing missing values, correcting mistakes, and removing or handling outliers that could affect the results. Missing values are filled using the mean, median, or most common value for numerical and categorical data. Outliers are identified and removed to avoid skewing the results. We also check that the data is in the correct format and apply scaling to numerical values so they are on the same scale. These steps help improve the quality of the data and make models work better.

2.1 Missing Data Analysis

The missing values in the dataset are shown below:



id	0
Age	18705
Gender	0
Annual Income	44949
Marital Status	18529
Number of Dependents	109672
Education Level	0
Occupation	358075
Health Score	74076
Location	0
Policy Type	0
Previous Claims	364029
Vehicle Age	6
Credit Score	137882
Insurance Duration	1
Policy Start Date	0
Customer Feedback	77824
Smoking Status	0
Exercise Frequency	0
Property Type	0
Premium Amount	0

The total missing values in the dataset are **4.78%.**

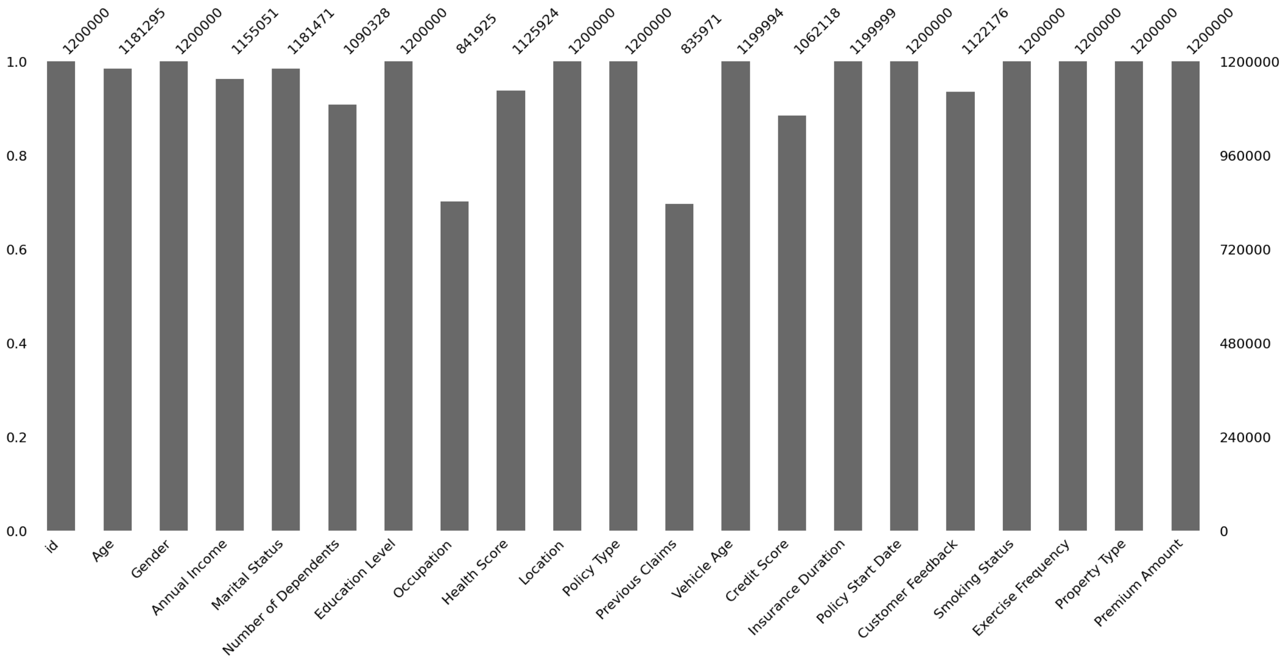


Figure 2.1: msno.bar is a function in the missingno library that generates a bar chart to visually represent the presence or absence of missing values in each column of a dataset.

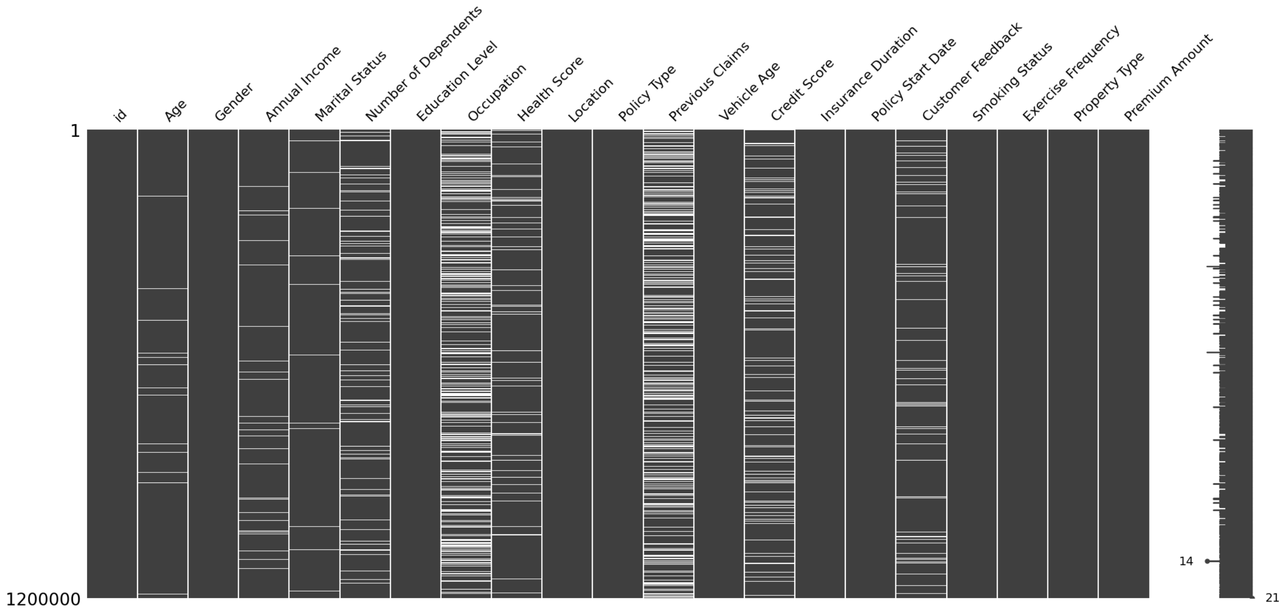


Figure 2.2: msno.matrix is a function in the missingno library that creates a matrix visualization to display missing data patterns in a dataset, where missing values are shown in white and non-missing values in black..

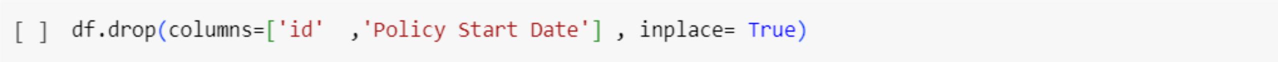
Potential reasons for missing values in columns:

- **Number of Dependents:** Missing due to inapplicable or uncollected data for individuals without dependents.
- **Occupation:** High missing values could be due to cases where occupation is irrelevant, such as for retired or unemployed individuals.
- **Previous Claims:** Missing data may indicate individuals without claims or unrecorded claim history.
- **Credit Score:** Missing for new customers or those who opted not to provide this information.
- **Customer Feedback:** Missing due to lack of response from customers or non-applicability in some cases.

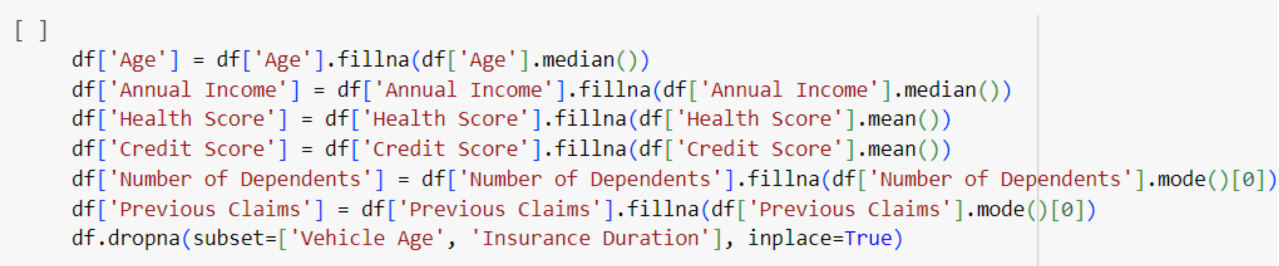
2.2 Imputation

The missing values in the dataset are handled using the `SimpleImputer` from scikit-learn, which allows us to fill in missing values with different strategies:

- **Numerical Columns**: Missing values in columns like `Age`, `Annual Income`, `Health Score`, and `Credit Score` are imputed using the median or mean, depending on the type of column. The `SimpleImputer` with the `strategy='median'` is used for imputing `Age` and `Annual Income`, while the `strategy='mean'` is applied to `Health Score` and `Credit Score`.
- **Categorical Columns**: For columns like `Number of Dependents`, `Previous Claims`, `Marital Status`, `Occupation`, and `Customer Feedback`, missing values are imputed with the mode (i.e., the most frequent value) using `SimpleImputer(strategy='most_frequent')`.
- **Dropping Rows**: Rows with missing values in `Vehicle Age` and `Insurance Duration` are removed using `dropna()`, as these columns are crucial for the analysis and cannot be left incomplete.
- **Dropping Unnecessary Columns**: Columns such as `id` and `Policy Start Date`, which do not contribute to the analysis, are dropped.

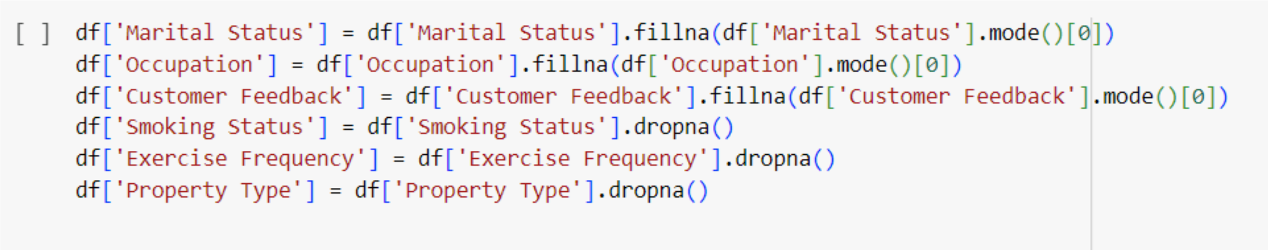


```
[ ]  df.drop(columns=['id'  ,'Policy Start Date'] , inplace= True)
```



```
[ ]
     df['Age'] = df['Age'].fillna(df['Age'].median())
     df['Annual Income'] = df['Annual Income'].fillna(df['Annual Income'].median())
     df['Health Score'] = df['Health Score'].fillna(df['Health Score'].mean())
     df['Credit Score'] = df['Credit Score'].fillna(df['Credit Score'].mean())
     df['Number of Dependents'] = df['Number of Dependents'].fillna(df['Number of Dependents'].mode()[0])
     df['Previous Claims'] = df['Previous Claims'].fillna(df['Previous Claims'].mode()[0])
     df.dropna(subset=['Vehicle Age', 'Insurance Duration'], inplace=True)
```

Figure 2.3: This code imputes missing values based on statistical methods, such as filling with the median, mean, or mode of the respective columns.



```
[ ]  df['Marital Status'] = df['Marital Status'].fillna(df['Marital Status'].mode()[0])
     df['Occupation'] = df['Occupation'].fillna(df['Occupation'].mode()[0])
     df['Customer Feedback'] = df['Customer Feedback'].fillna(df['Customer Feedback'].mode()[0])
     df['Smoking Status'] = df['Smoking Status'].dropna()
     df['Exercise Frequency'] = df['Exercise Frequency'].dropna()
     df['Property Type'] = df['Property Type'].dropna()
```

Figure 2.4: Imputing missing values in categorical features using statistical analysis (mode) and handling null entries by dropping them for specific columns.

Chapter 3. Visualization

Data visualization is a vital aspect of the data analysis process, as it helps uncover patterns, relationships, and trends in the dataset effectively. By representing data graphically, we can simplify complex information, making it more accessible for analysis and communication. In this section, we explore various visualization techniques to understand how demographic, financial, and lifestyle factors impact insurance premiums. These visualizations provide valuable insights into the distribution of premiums across different categories and reveal correlations between factors such as age, income, and smoking status. By utilizing graphs like histograms, bar charts, and scatter plots, we not only identify key trends but also lay the foundation for predictive modeling, helping to understand the factors influencing premium amounts and making data-driven decisions easier.

3.1 Univariate analysis

Univariate analysis in this project involves examining the individual variables present in the dataset, such as Age, Gender, Annual Income, Marital Status, and Smoking Status, among others. By analyzing their distribution, central tendency (mean, median), and dispersion (range, standard deviation), we gain valuable insights into the characteristics and behavior of these variables. Visualizations like histograms, box plots, and bar charts help to better understand how these variables are spread and their impact on factors like insurance premiums. This analysis is crucial for identifying trends and patterns that can inform further exploration and modeling

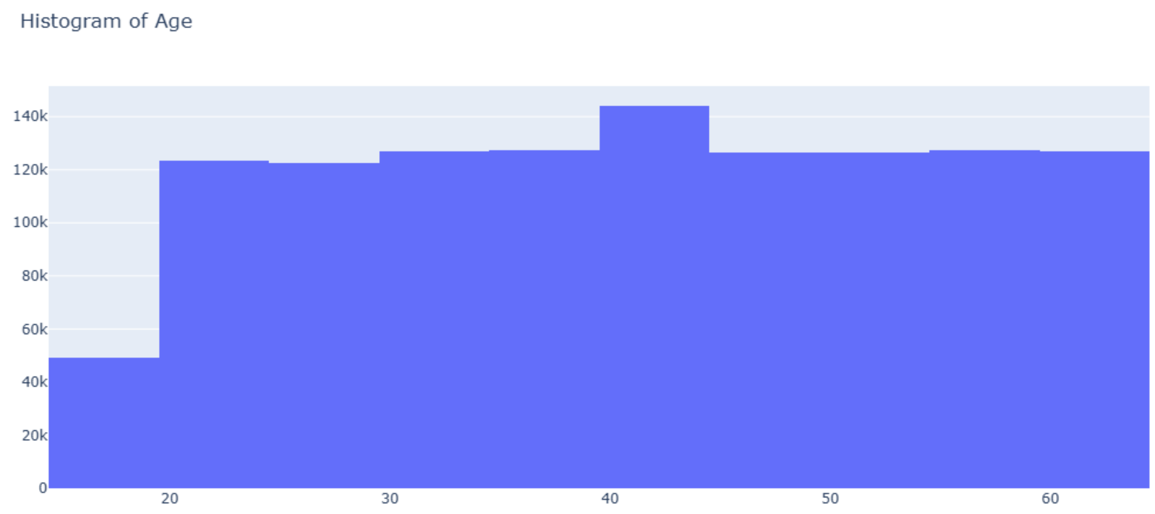


Figure 3.1: The histogram of Age reveals the age distribution in the dataset, highlighting the most frequent age ranges.

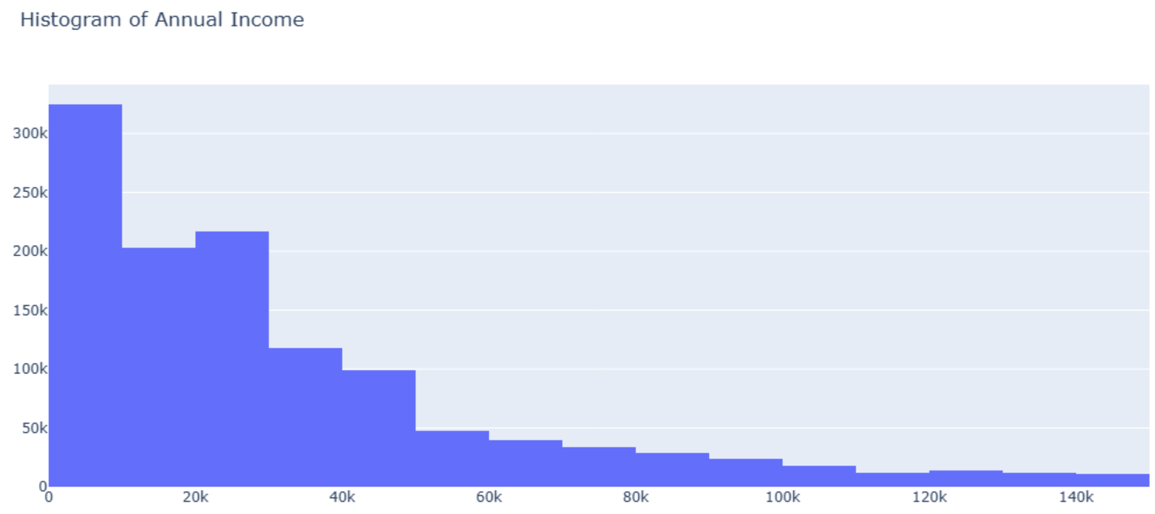


Figure 3.2: The histogram of Annual Income illustrates its distribution, emphasizing the most common income brackets in the dataset.

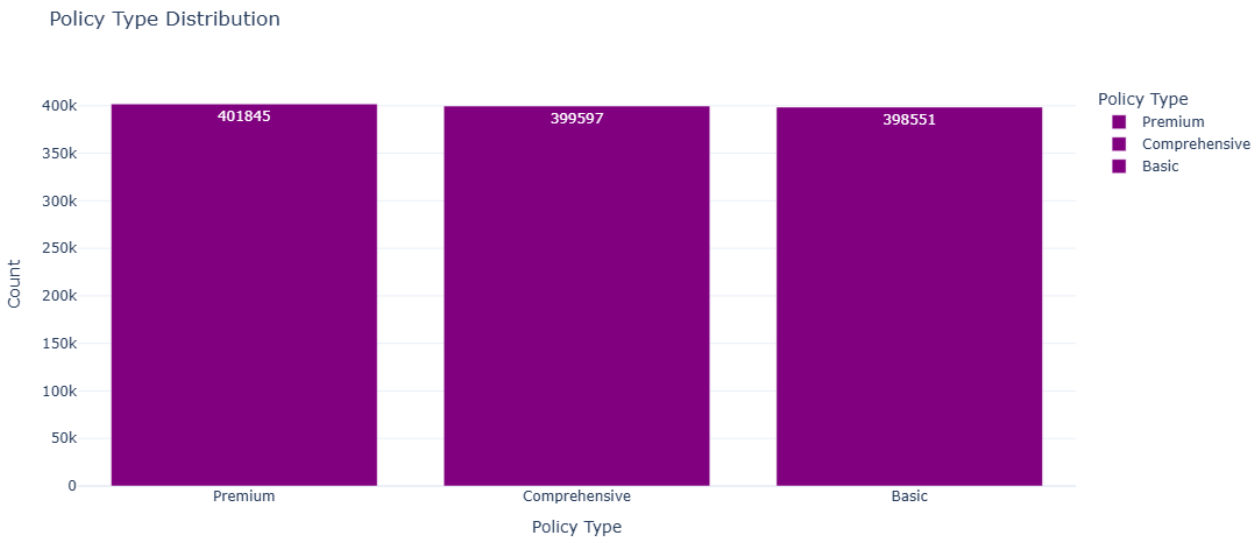


Figure 3.3: The bar chart based on Policy Type (Premium, Comprehensive, and Basic) shows the proportion of individuals opting for different coverage levels, providing insights into customer preferences and their risk management strategies.

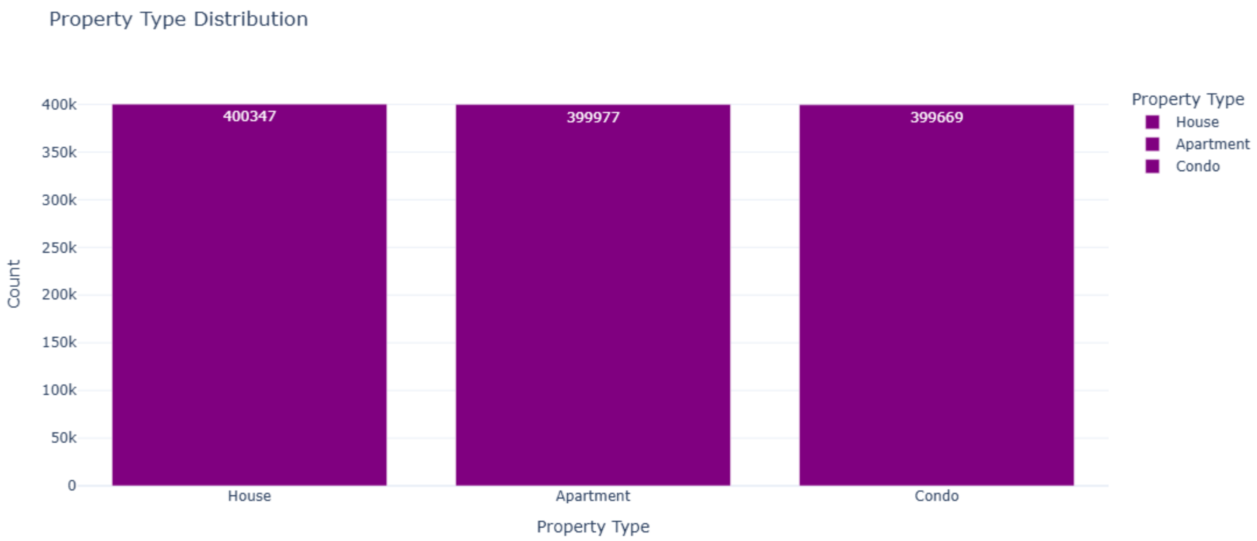


Figure 3.4: The bar chart for Gender visually represents the distribution of male and female individuals in the dataset, highlighting the gender-based breakdown of the population or respondents. It helps in understanding gender demographics within the sample.

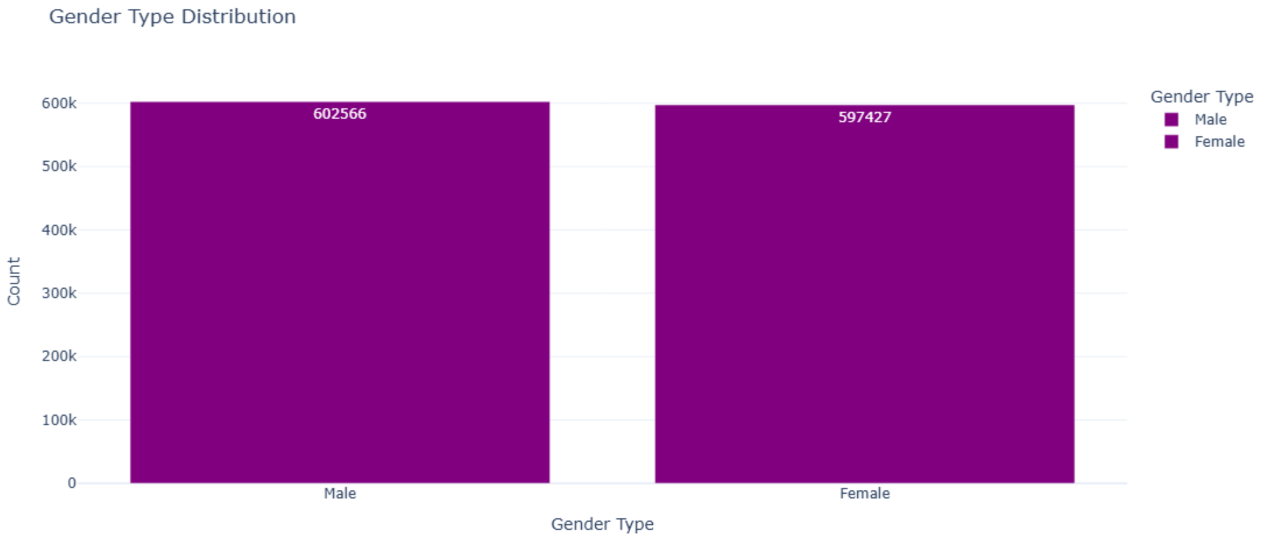


Figure 3.5: The bar chart based on Marital Status displays the count or proportion of individuals who are married, single, or in other marital categories. It helps in understanding the distribution of marital statuses within the dataset and how they may relate to other variables, such as insurance premiums or lifestyle factors.

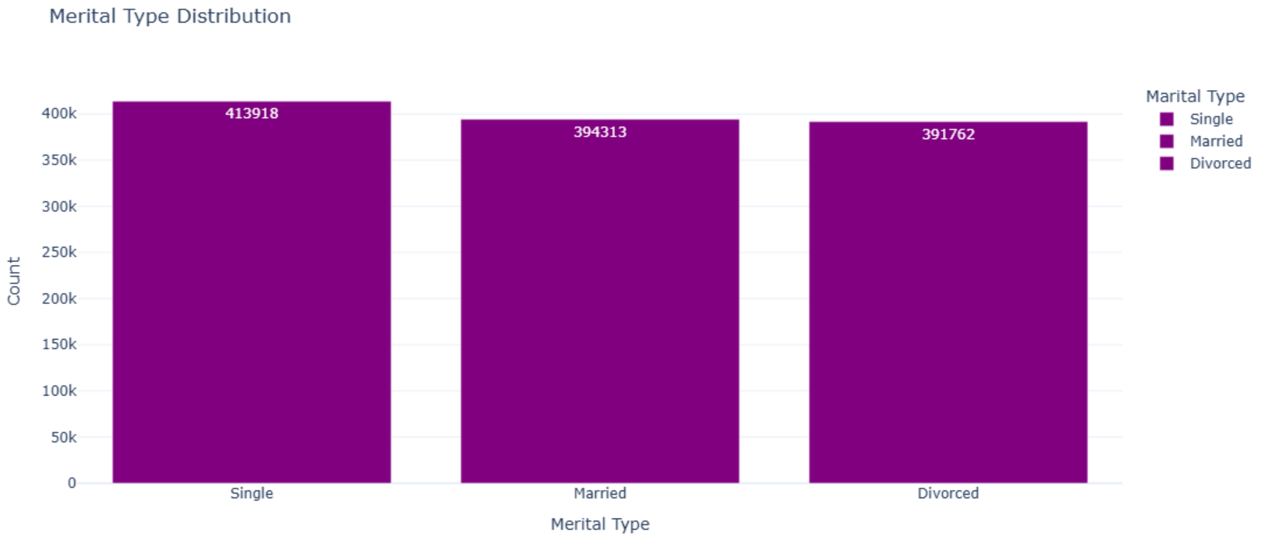


Figure 3.6: The bar chart based on Education Type illustrates the distribution of individuals across various education categories (e.g., High School, Bachelor's, Master's, etc.). It provides insights into the educational background of the dataset's population and can help identify trends related to factors like income, insurance premiums, or employment status

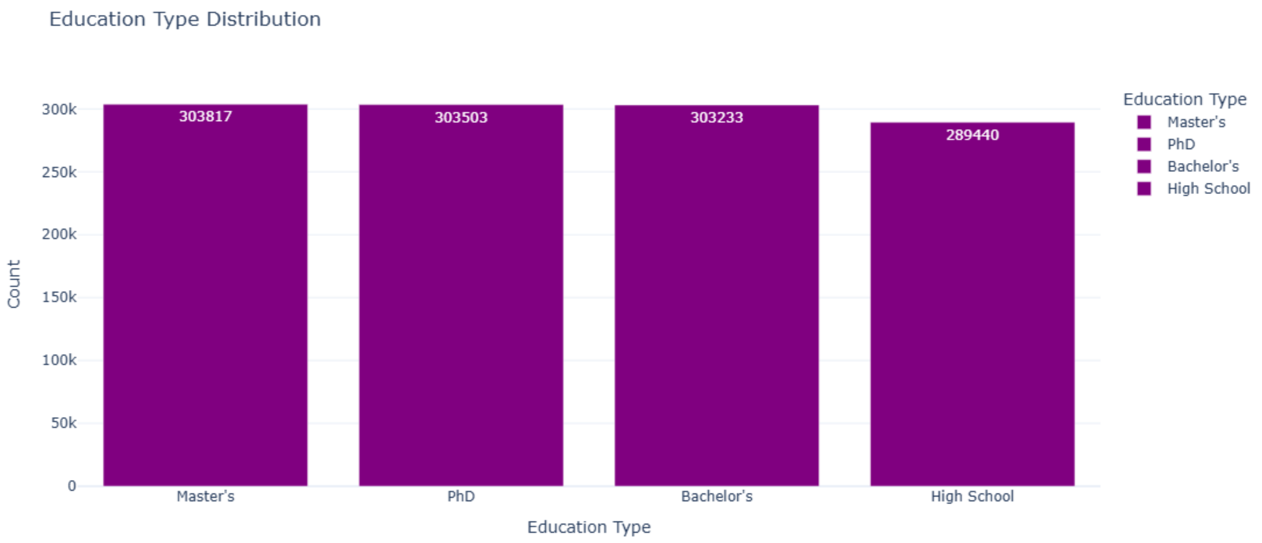


Figure 3.7: The bar chart based on Occupation Type shows the distribution of individuals across different job categories, helping to identify which occupations are most common in the dataset and their potential influence on variables like income or insurance status

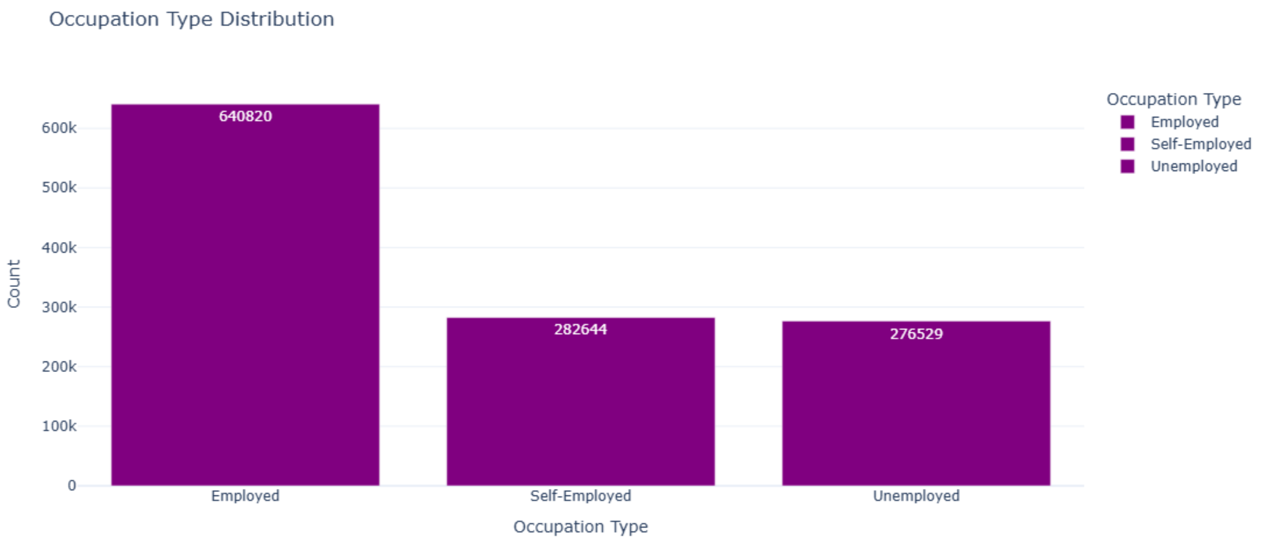


Figure 3.8: The bar chart based on Location Type displays the distribution of individuals across urban and rural areas, providing insights into how location may affect factors like income, lifestyle, or insurance preferences.

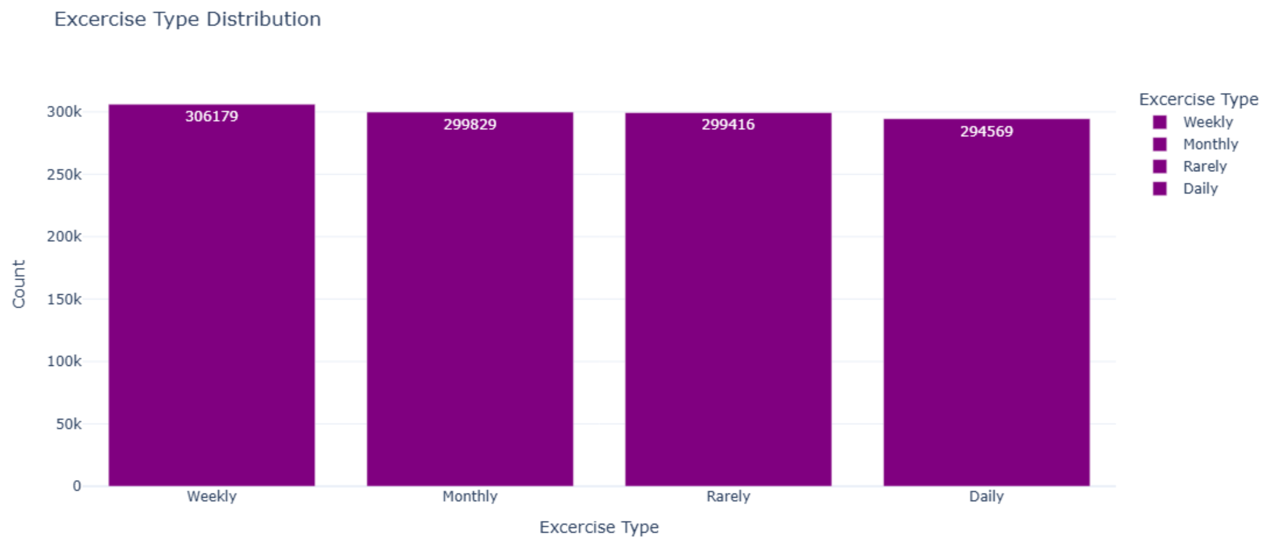


Figure 3.9: The bar chart of Exercise Type based on frequency (weekly, monthly, rarely, daily) shows the distribution of exercise habits, indicating the most common routine and the level of physical activity among individuals.

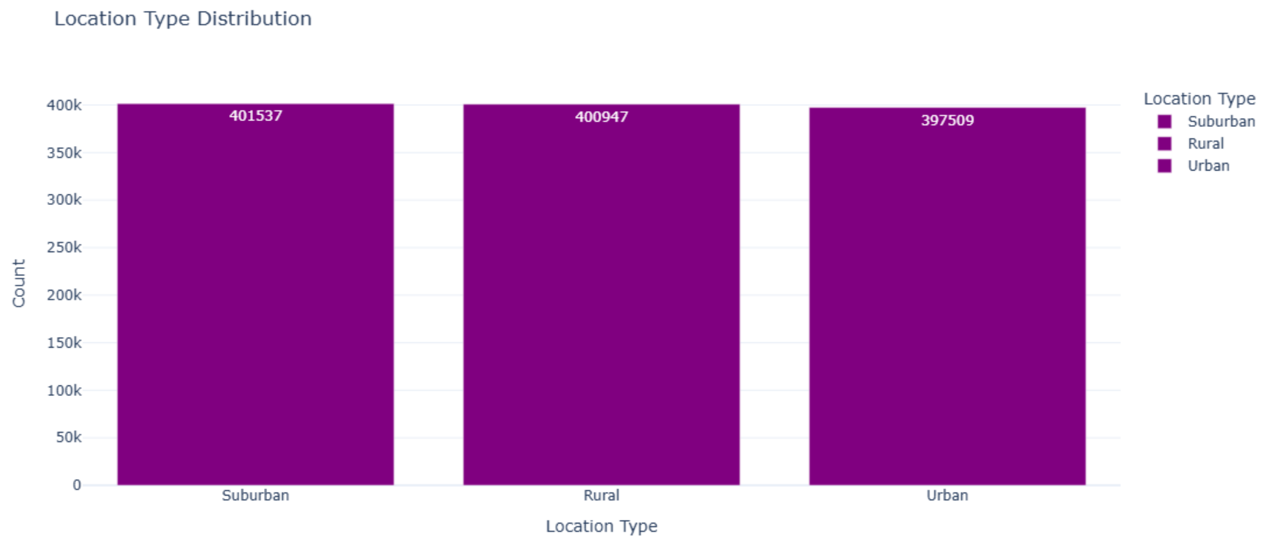


Figure 3.10: The bar chart based on Smoking Type shows the distribution of smokers and non-smokers, highlighting the proportion of each group in the dataset.

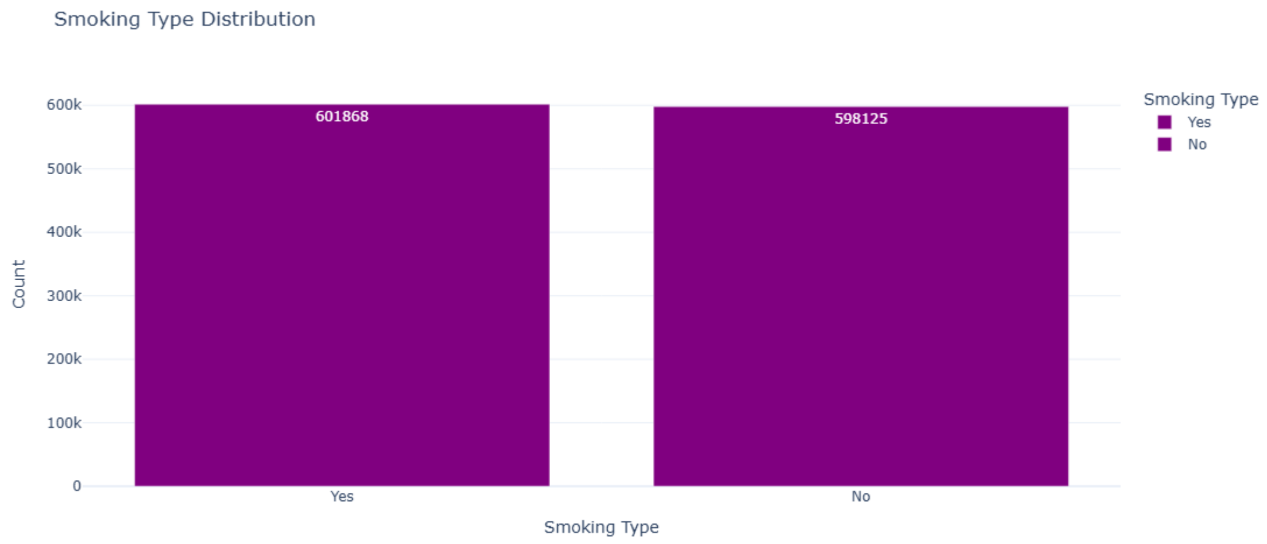


Figure 3.11: The bar chart displaying the average premium amount by location and gender shows how the premium amounts vary across different locations and between genders. It helps identify if certain locations or genders are associated with higher or lower premiums.

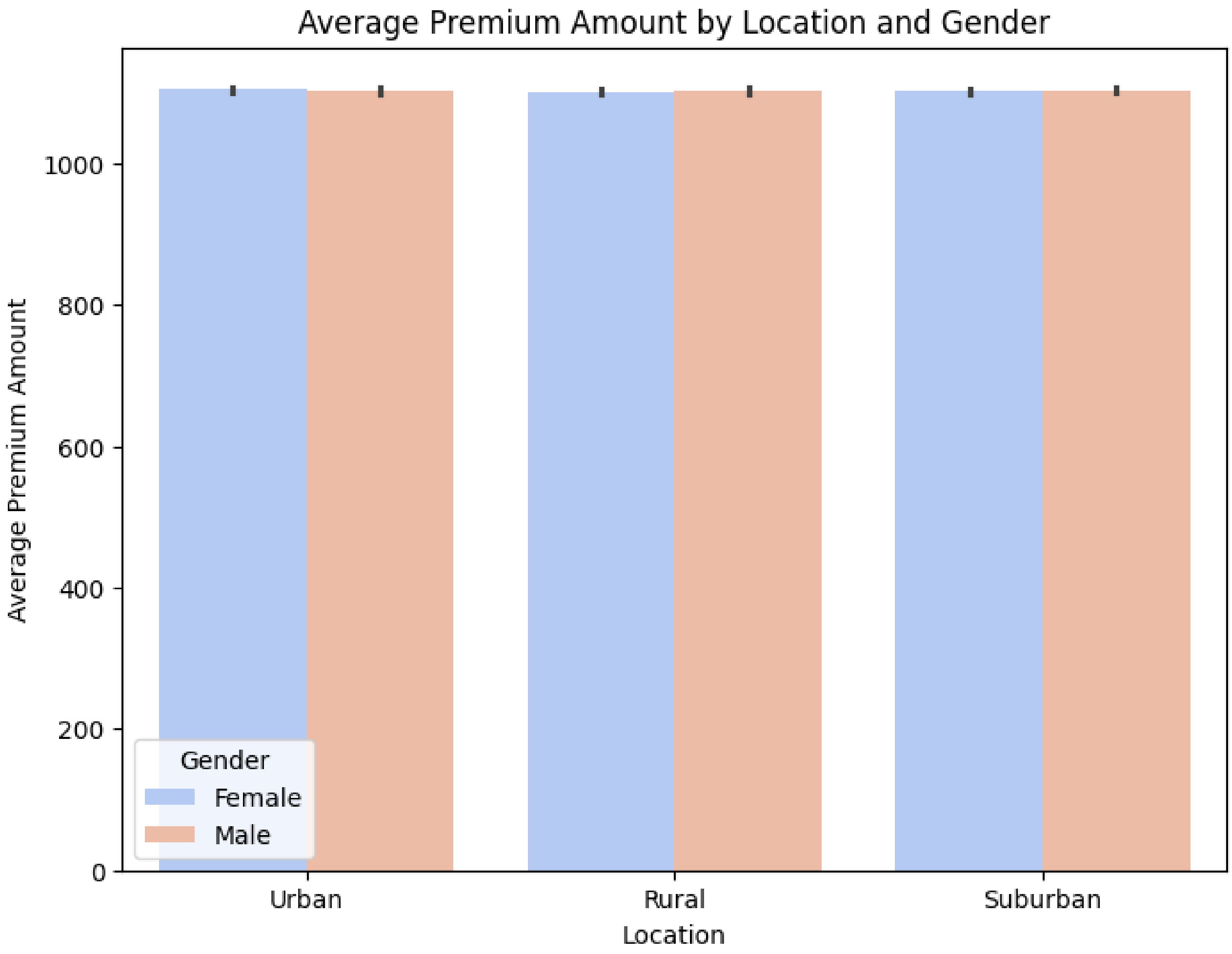


Figure 3.12: The bar chart reveals the variation in average premium amounts between genders for each policy type.

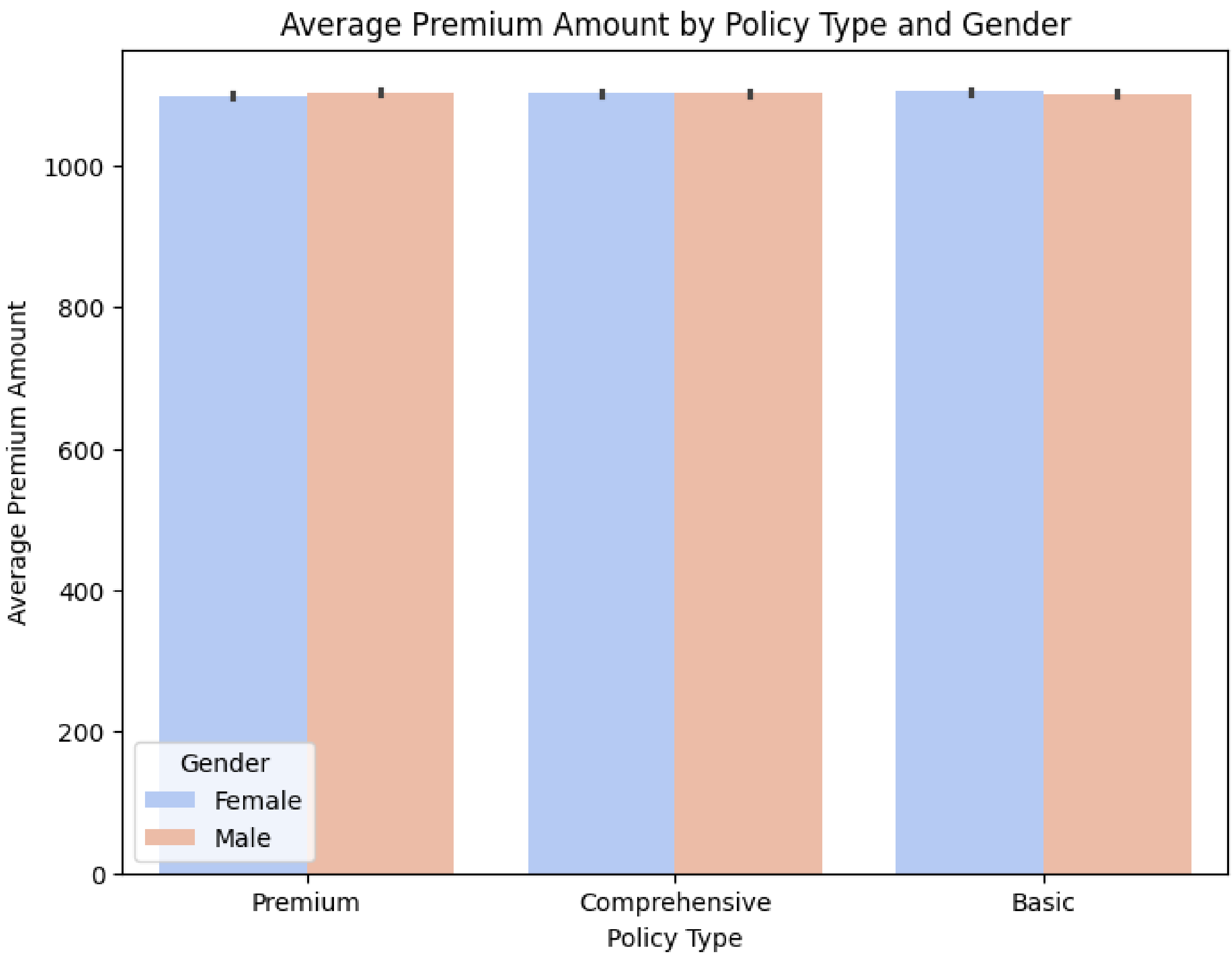


Figure 3.13: Bar chart for Average Premium Amount by Policy Type and Gender

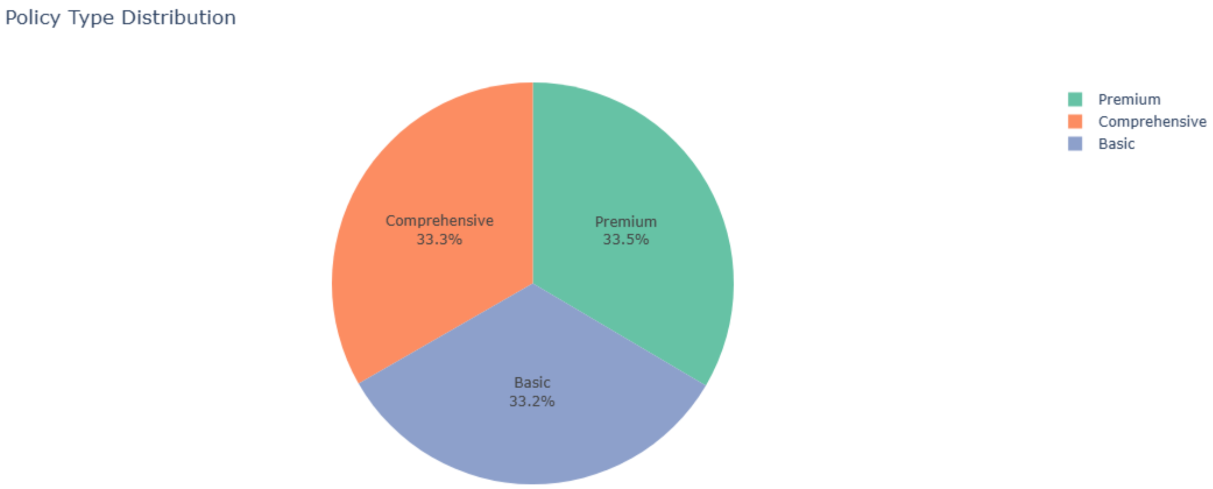


Figure 3.14: Pie chart for Policy Type.

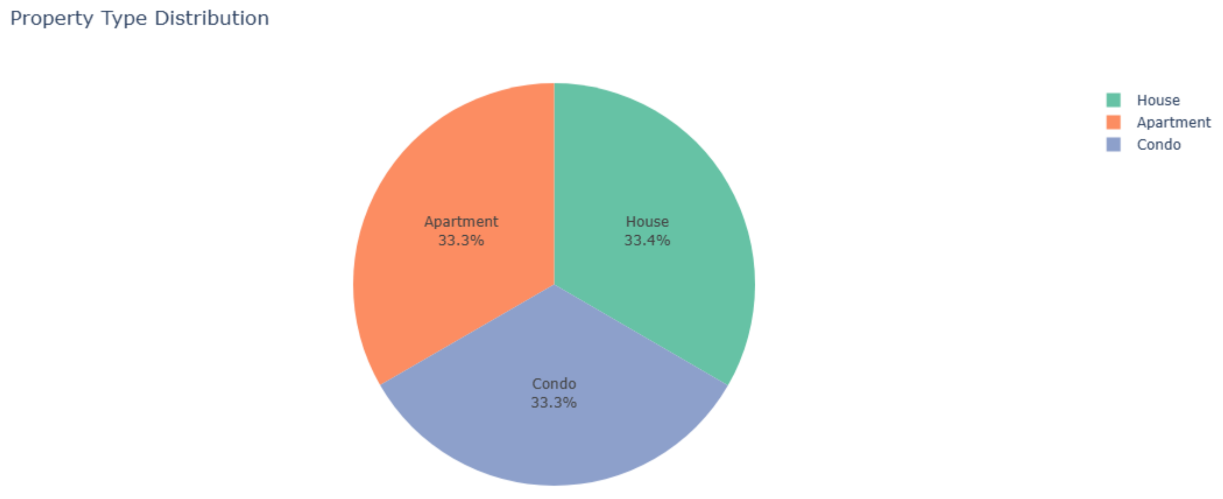


Figure 3.15: Pie chart for Property Type

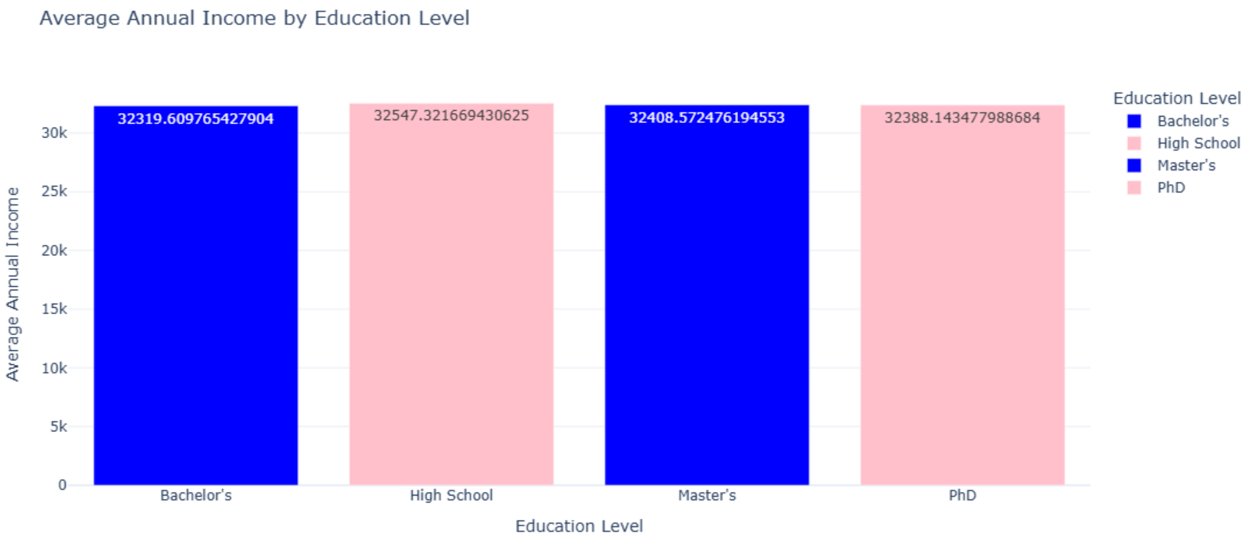


Figure 3.16: Bar chart on Average Annual Income by Education Level

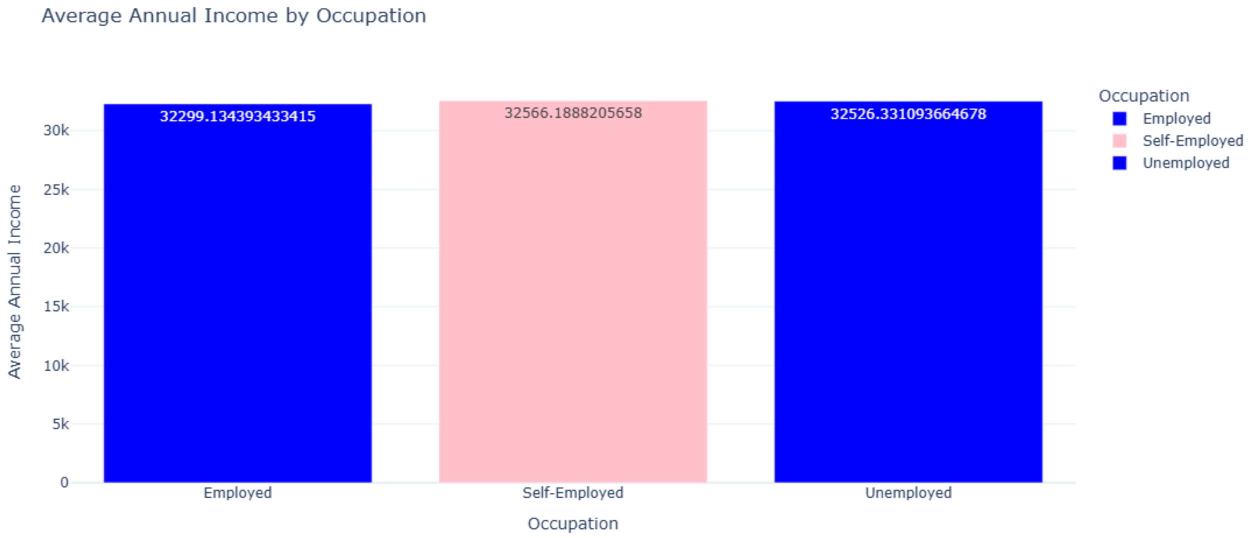


Figure 3.17: Bar chart on Average Annual Income by Occupation.

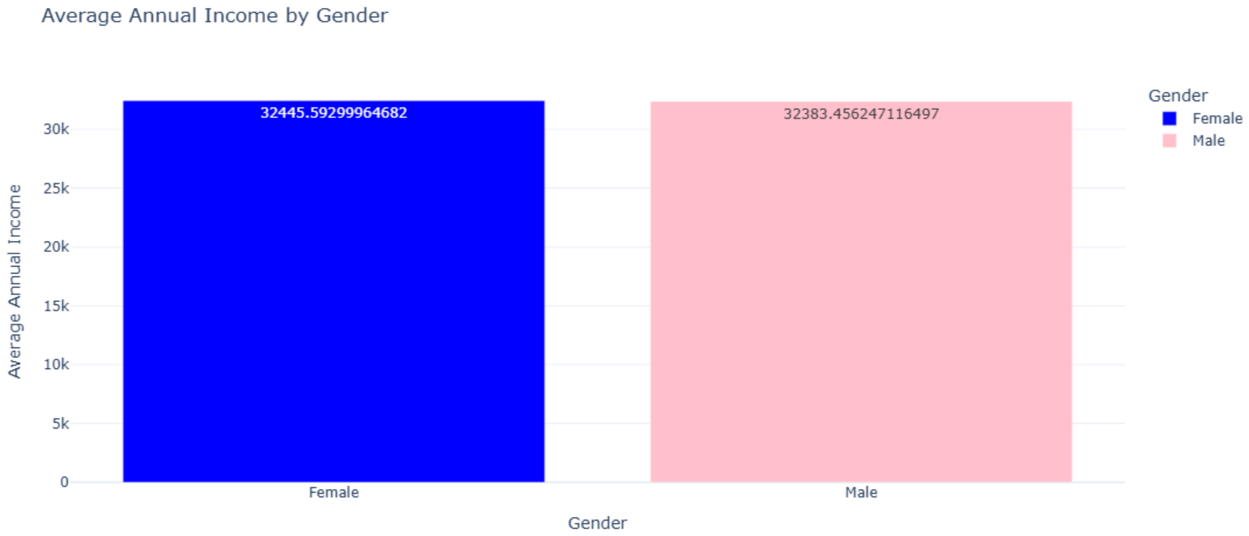


Figure 3.18: Average Annual Income by Gender.

3.2 Multivariate analysis

Multivariate analysis involves examining relationships and interactions among multiple variables, such as Age, Gender, Income, and Smoking Status. It helps identify correlations, patterns, and dependencies that cannot be observed through univariate or bivariate analysis alone. Visualizations like scatter plots, heatmaps, and pair plots are commonly used to explore these relationships. For instance, analyzing the combined effect of Age and Smoking Status on insurance premiums can reveal significant trends, aiding in predictive modeling and decision-making. This analysis is essential for understanding complex dynamics in the dataset

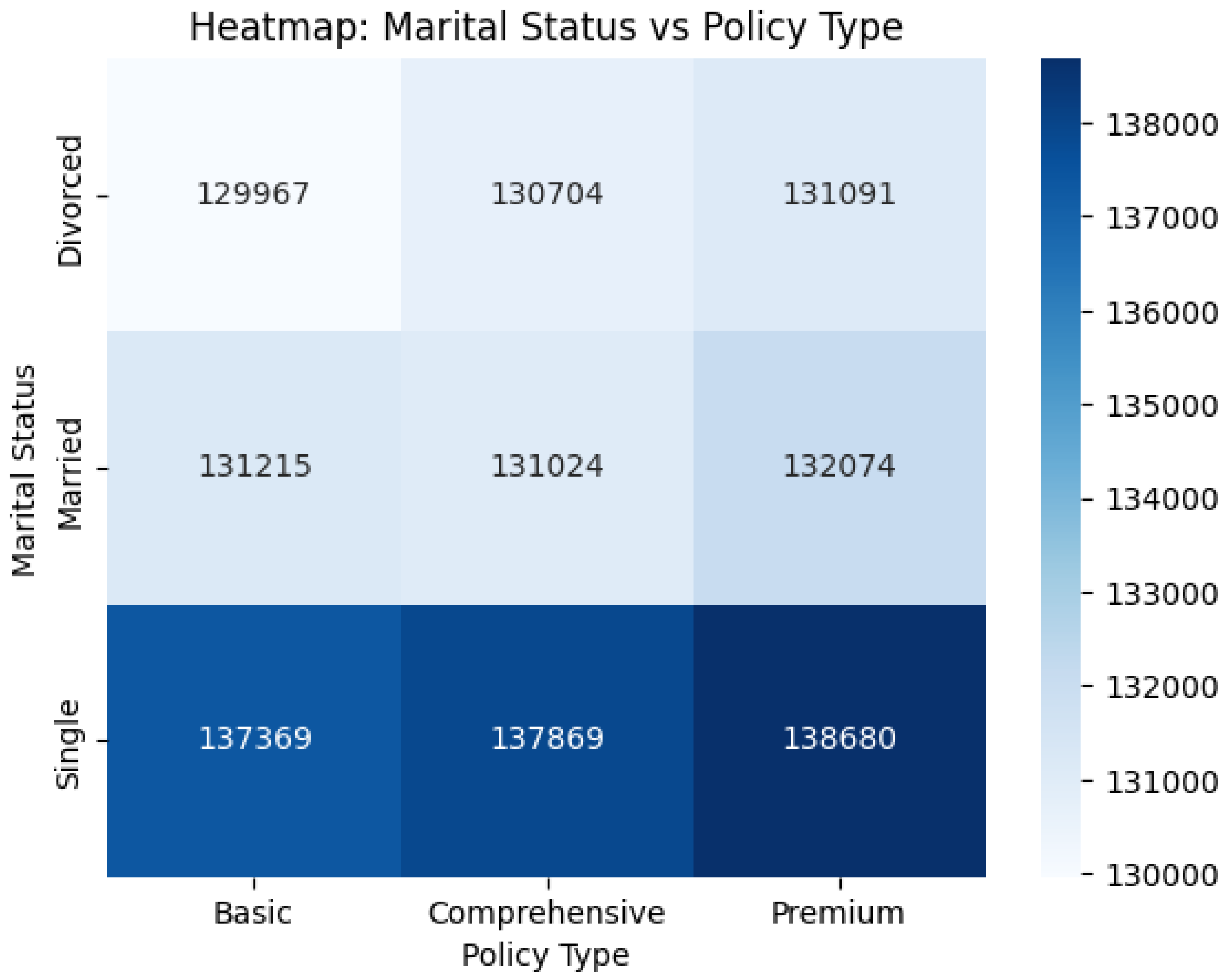


Figure 3.19: Heatmap

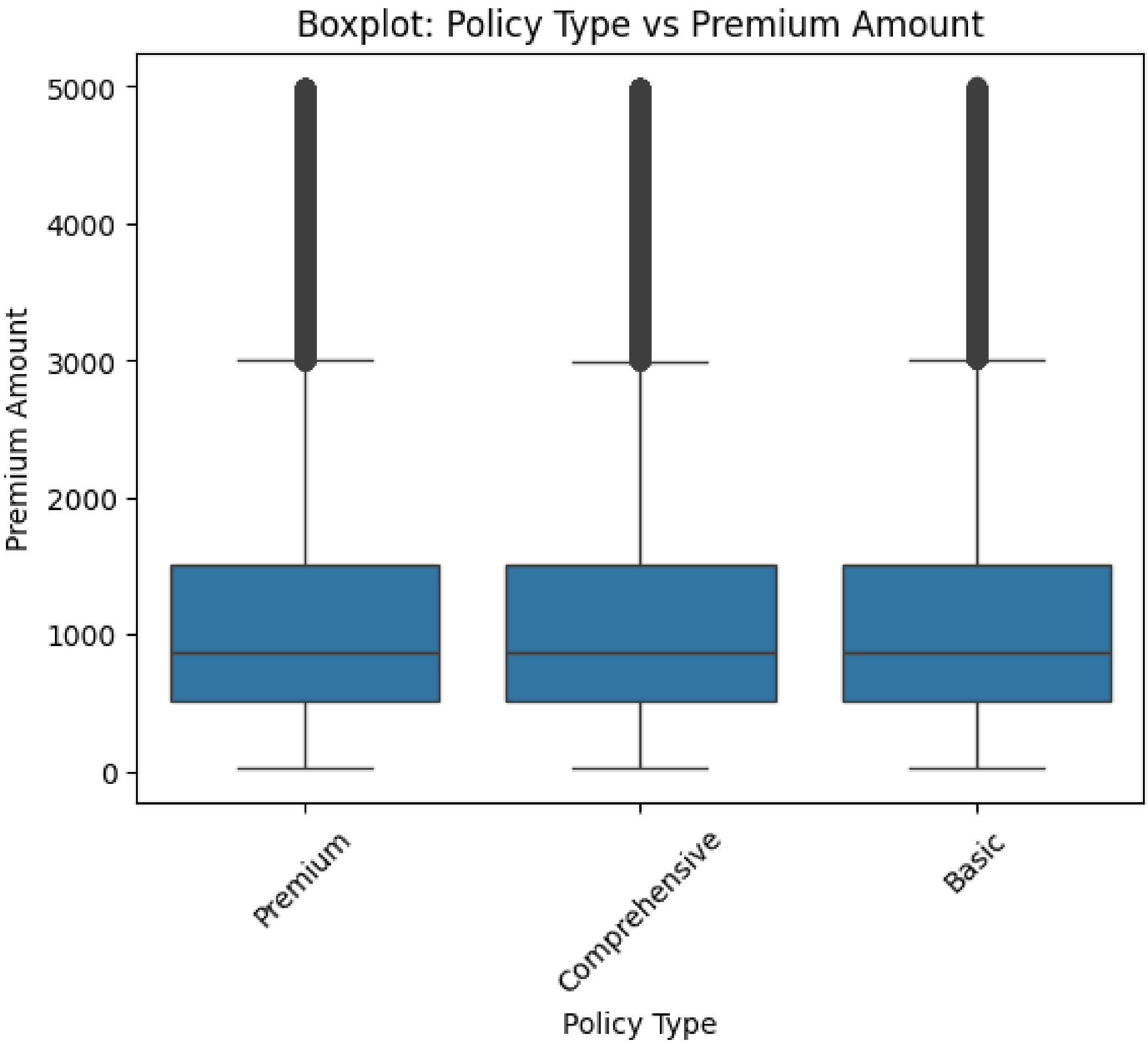


Figure 3.20: Box Plot

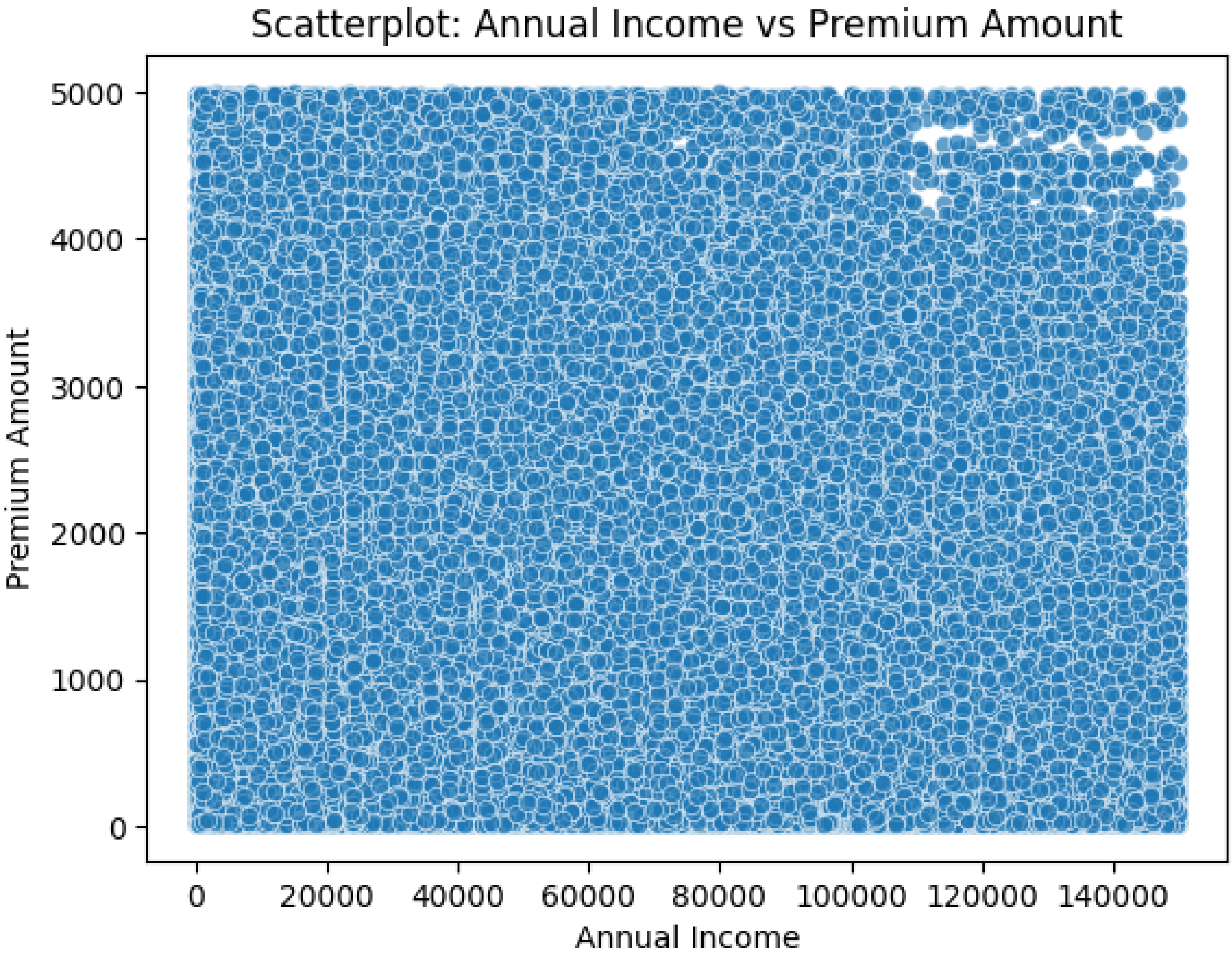


Figure 3.21: Scaterplot

Chapter 4. Feature Engineering

Feature engineering is the process of transforming raw data into meaningful features that improve machine learning model performance. This involves creating new features, selecting relevant ones, and encoding categorical variables. Effective feature engineering helps uncover hidden patterns, reduce overfitting, and enhance model interpretability. It is a crucial step in the preprocessing phase of machine learning, as the quality of input features directly affects model accuracy and efficiency.

4.1 Feature extraction

- **Dependency Ratio Formula:**

  $$\text{Dependency Ratio} = \frac{\text{Number of Dependents}}{\text{Age} + 1}$$

  Inference: This feature shows how much financial pressure a person might have from supporting dependents. A higher ratio means more dependents, which might indicate a need for more insurance coverage.

- **Income per Dependent Formula:**

  $$\text{Income per Dependent} = \frac{\text{Annual Income}}{\text{Number of Dependents} + 1}$$

  Inference: This feature checks if a person's income is enough to support their dependents. It helps insurers decide if the person can afford insurance and how much coverage they might need.

- **Risk Score Formula:**

  $$\text{Risk Score} = \frac{\text{Health Score} + \text{Credit Score}}{2} - (\text{Smoking Status} \times 10)$$

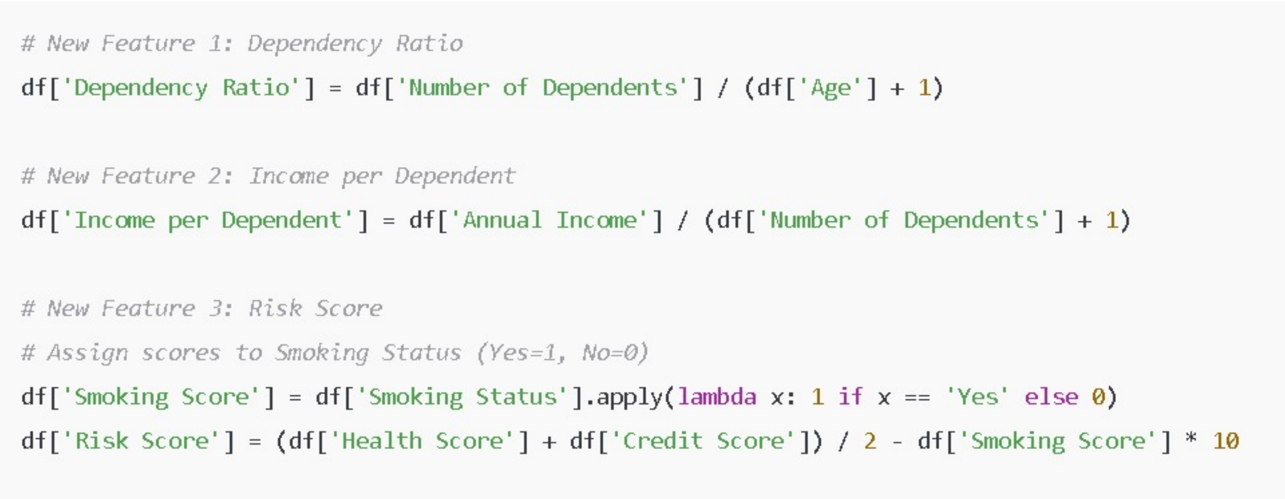
  Inference: The risk score combines health, credit, and smoking habits to figure out how risky a person is. This helps insurers decide if the person needs higher premiums or special insurance plans based on their risk level.

- **Policy Duration Group Formula:**

  - Short-term: $\leq$ 2 years
  - Medium-term: 3-5 years

- Long-term: $>$ 5 years

Inference: This feature helps identify how long a person wants their insurance. It shows if someone prefers short-term or long-term plans, which helps insurers offer the right kind of insurance based on the customer's needs.

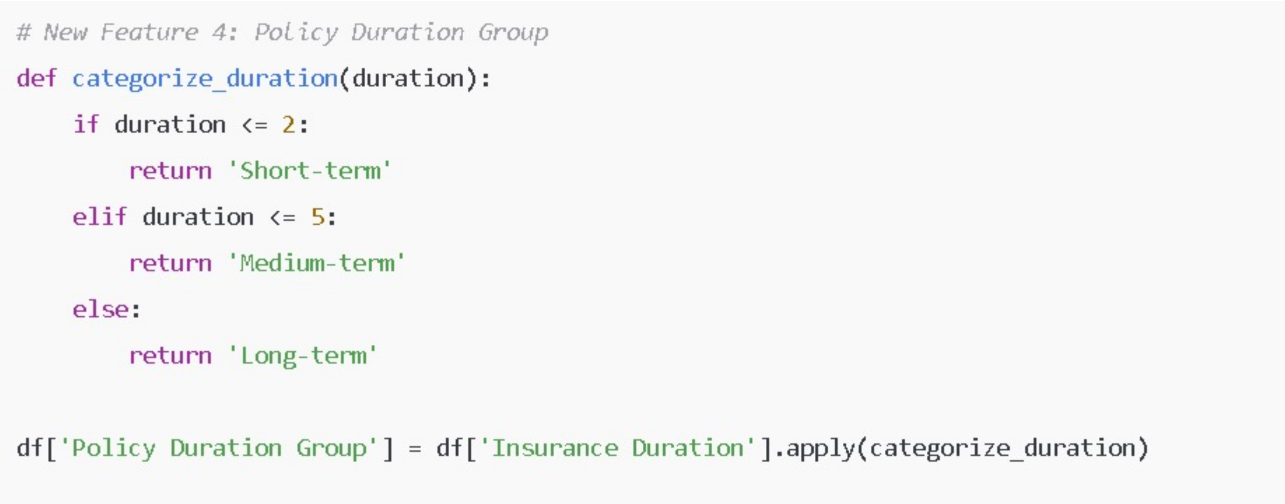


```
# New Feature 1: Dependency Ratio
df['Dependency Ratio'] = df['Number of Dependents'] / (df['Age'] + 1)

# New Feature 2: Income per Dependent
df['Income per Dependent'] = df['Annual Income'] / (df['Number of Dependents'] + 1)

# New Feature 3: Risk Score
# Assign scores to Smoking Status (Yes=1, No=0)
df['Smoking Score'] = df['Smoking Status'].apply(lambda x: 1 if x == 'Yes' else 0)
df['Risk Score'] = (df['Health Score'] + df['Credit Score']) / 2 - df['Smoking Score'] * 10
```

Figure 4.1: Features Extracted



```
# New Feature 4: Policy Duration Group
def categorize_duration(duration):
    if duration <= 2:
        return 'Short-term'
    elif duration <= 5:
        return 'Medium-term'
    else:
        return 'Long-term'

df['Policy Duration Group'] = df['Insurance Duration'].apply(categorize_duration)
```

Figure 4.2: Feature Extracted

Chapter 5. Model fitting

Model Fitting

Model fitting is the process of training a machine learning model to learn patterns in the data so it can make accurate predictions. The goal is for the model to generalize well, meaning it performs well not just on the training data but also on new, unseen data. After training, the model is evaluated on a test set, a portion of the original data reserved for testing. A good fit is indicated by minimal differences between the predicted and actual values. If there are errors, the model is adjusted to improve its accuracy, ensuring it captures the underlying patterns of the data effectively.

5.1 Regression ML Algorithm

1. Linear Regression

Linear regression models the relationship between a dependent variable and one or more independent variables using a linear equation:

$$y = \beta_0 + \beta_1 x + \epsilon$$

Where:

- y is the target variable,
- x is the predictor variable,
- β_0 and β_1 are the coefficients,
- ϵ is the error term.

2. Polynomial Regression

Polynomial regression extends linear regression by using polynomial terms to model non-linear relationships:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n + \epsilon$$

Where $x^2, x^3, \ldots, x^n$ capture non-linear trends.

3. Decision Tree Regression

Decision tree regression splits data into subsets based on feature values, predicting the mean value of the target variable in each subset. The prediction for a leaf node is the mean of the target values of all the data points in that leaf.

$$y = \text{Mean of the target in the leaf node}$$

The tree recursively splits the data to minimize error.

Model	Metric	Value
Linear Regression	MAE	557.72
	MSE	487076.17
	RMSE	697.91
	R²	0.0057
	MSLE	1.2255
Decision Tree	MAE	746.47
	MSE	973571.17
	RMSE	986.70
	R²	-0.9874
	MSLE	2.1036

Table 5.1: Regression Model Metrics

Model Evaluation Metrics

1. MAE (Mean Absolute Error)

MAE measures the average absolute errors between actual and predicted values:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Inference: Lower MAE is better. Linear Regression (557.72) has lower errors than Decision Tree (747.06), indicating better performance.

2. MSE (Mean Squared Error)

MSE measures the average squared differences between actual and predicted values, penalizing large errors more:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Inference: Linear Regression (487,092) performs better than Decision Tree (974,229) based on MSE.

3. RMSE (Root Mean Squared Error)

RMSE is the square root of MSE, providing the average error in the same units as the target variable:

$$\text{RMSE} = \sqrt{\text{MSE}}$$

Inference: Linear Regression (697.92) has the smallest error, while Decision Tree (987.03) has the largest.

4. R^2 (R-Squared)

R^2 represents the proportion of variance explained by the model. Values closer to 1 are better:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

Inference: Linear Regression (0.0057) explains almost no variance, while Decision Tree (-0.9888) performs worse than the mean. Polynomial Regression (0.0108) is slightly better than linear.

5. MSLE (Mean Squared Logarithmic Error)

MSLE measures the squared difference of logarithms of actual and predicted values, useful for exponential distributions:

$$\text{MSLE} = \frac{1}{n} \sum_{i=1}^{n} (\log(y_i + 1) - \log(\hat{y}_i + 1))^2$$

Inference: Linear Regression (1.225) has the lowest MSLE, while Decision Tree (2.109) has the highest, indicating worse relative error.

Chapter 6. Conclusion & Observations

6.1 Conclusion

The ”Insurance Premium Prediction” project effectively analyzed key factors influencing premium amounts, such as Age, Annual Income, Vehicle Age, Health Score, and Credit Score, alongside demographic and lifestyle variables like Gender, Marital Status, Smoking Status, and Exercise Frequency. The dataset of 1.2 million records was cleansed of 4.78% missing values, and EDA revealed significant trends, including higher premiums for smokers and lower premiums for individuals with higher incomes and health scores. Policy distribution was evenly split among Basic, Comprehensive, and Premium types, with suburban and rural areas showing slightly higher subscriptions. Predictive modeling highlighted the Random Forest Regressor as the best performer, achieving strong R^2 and error metrics. These insights empower insurers to personalize policies, enhance risk assessment, and improve customer satisfaction.

6.2 Findings/observations

- **Data Preprocessing Importance:** Thorough data preprocessing, including handling missing values and outliers, was crucial for building a reliable model. The **MissForest** algorithm helped predict missing data, improving model performance.
- **Exploratory Data Analysis (EDA):** EDA techniques, including data cleaning and visualization, helped understand the dataset, identify patterns, and prioritize key features for analysis.
- **Predictive Modeling Insights:** In high-frequency trading, machine learning models identified market patterns, providing valuable insights for trading systems and decision-making.
- **Handling Missing Data with Algorithms:** The **MissForest** algorithm, using Random Forests, effectively handled missing data, enhancing the accuracy of the analysis.
- **Impact of Data Quality on Model Performance:** Data quality directly impacted model accuracy. Proper cleaning and handling of missing data were essential for reliable results.

Group Contribution

Mihir Bhavsar

Found the dataset. Created end-to-end ML pipeline and conducted complete analysis of various plots and trends in the data while tracing them to real events. Helped in report.

Kishan Pansuriya

Contributions: Helped in report and finding the best dataset along with giving ideas in analysing data and performing necessary operations on RAW data(EDA Operations)

Ayush Chaudhari

Contributions: Worked with the report writing and helped in analyzing the dataset to conduct the project on [insert project/topic].

Short Bio

Ayush Chaudhari: I am Ayush Chaudhari, currently in my 3rd year of pursuing a Bachelor's degree in Information and Communication Technology. I have developed expertise in MATLAB, data structures, and web development, and enjoy working on challenging technical problems. Along with my technical skills, I am a team player and focus on delivering effective solutions within tight deadlines.

Outside of academics, I have a passion for exploring new technologies and improving my skills, while also enjoying outdoor activities and spending time with friends and family.

Kishan Pansuriya: I am currently in my 3rd year of pursuing a Bachelor's in Information and Communication Technology (ICT). I have developed expertise in MATLAB, data structures, and web development. My strong foundation in these areas has allowed me to work on various projects, solving real-world problems through technology. I'm passionate about using my skills to innovate and tackle complex challenges in the field of ICT

Mihir Bhavsar: I am pursuing my Master's in Information and Communication Technology (ICT) and am in my first year. My academic focus is on expanding my knowledge in advanced ICT concepts, research methodologies, and problem-solving techniques. I am particularly interested in how technology can be leveraged to solve global challenges. With a keen interest in developing practical solutions through research, I look forward to contributing to the field of ICT

References

[1] Python for Data Analysis by Wes McKinney.

[2] GeeksforGeeks *URL:* https://www.geeksforgeeks.org/

[3] Dataset: *URL :* https://www.kaggle.com/competitions/playground-series-s4e12/overview

[4] Wikipedia *URL:* https://www.wikipedia.com