Contents

Table list

Figure list

About the Report

This report presents a comprehensive analysis of a dataset targeting “AgentBonus” spanning two modules: Module 1 and Module 2. The primary objective of this report is to harness data science methodologies and tools, such as Python, R, SQL, Tableau, and Power BI, to derive meaningful insights and predictive models for enhancing business decision-making.

Module 1: Data Exploration and Analysis

Module 1 serves as the foundation of this Business Report. It commences with data preprocessing, including addressing missing values and handling outliers, ensuring that the dataset is ready for analysis. Key data visualizations using Python and Tableau provide a deeper understanding of the dataset's characteristics. Through exploratory data analysis (EDA), we uncover vital patterns and relationships within the data. In this module, we conduct both univariate and bivariate analysis, focusing on relationships between variables and their impact on the target variable, 'AgentBonus.'

The report for Module 1 culminates with the identification of business insights drawn from the data analysis. These insights illuminate areas of significance and set the stage for predictive modeling in Module 2.

Module 2: Model Building and Interpretation

Module 2 advances the project by delving into predictive modeling. The primary focus here is on optimizing the prediction of 'AgentBonus' using various models, including Linear Regression, Random Forest, and an ensemble model known as Stacking. Model interpretation and business implications are central to this module.

The report for Module 2 underscores the significance of model building and emphasizes the interpretation of model results in the context of business objectives. It outlines the steps in model development and tuning, explores ensemble modeling, and draws clear connections between model performance and actionable business insights. Visualizations, such as predictions vs. actuals and feature importance, are employed to bolster interpretations and facilitate a more intuitive understanding of the results.

In conclusion, this report encapsulates the journey from data exploration and analysis in Module 1 to predictive modeling and its business implications in Module 2. It serves as a crucial reference for stakeholders, decision-makers, and data scientists, aiming to harness data-driven insights to enhance business performance. The meticulous application of data science techniques and tools throughout the project contributes to more informed decision-making, strategic planning, and resource allocation. This report provides a comprehensive overview, enabling a better understanding of the data-driven insights and their potential impact on the organization.

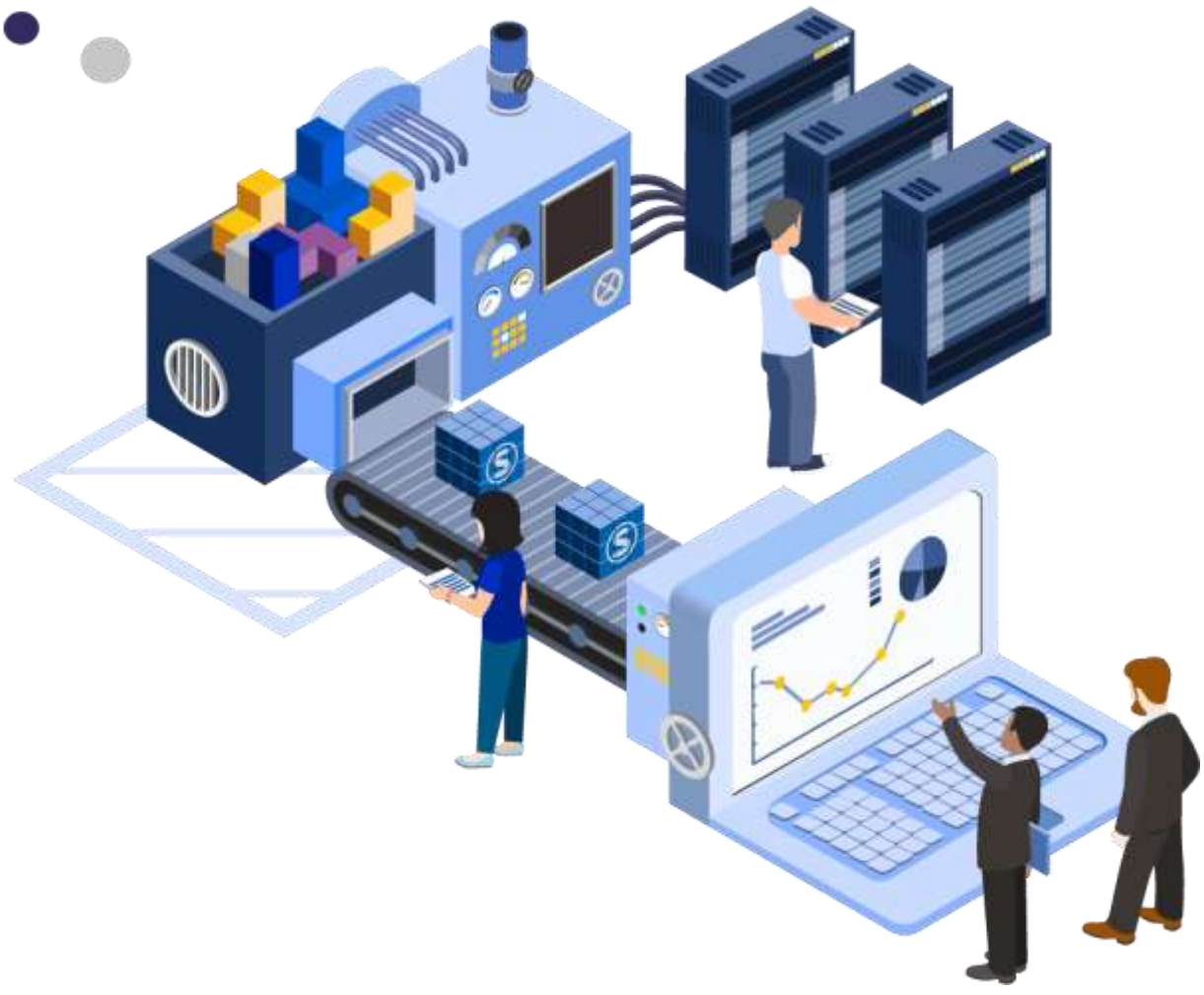
Business task

The dataset belongs to a leading life insurance company. The company wants to predict the bonus for its agents so that it may design appropriate engagement activity for their high-performing agents and upskill programs for low-performing agents.

Business Problem overview

The challenge at hand involves leveraging a dataset from a prominent life insurance company to forecast bonus amounts for its agents. This prediction task serves a dual purpose: firstly, to identify high-performing agents who deserve targeted engagement activities and incentives to sustain their excellent performance, and secondly, to pinpoint underperforming agents who would benefit from skill-enhancement programs and additional support. Essentially, the company aims to harness the power of data-driven insights to reward and motivate its top talent while simultaneously nurturing the development of agents who require improvement. By doing so, the company not only seeks to optimize its resource allocation but also aims to foster a more dynamic and productive agent workforce, ultimately enhancing its competitiveness and customer service quality in the life insurance sector.

Module 1



Data Overview and Preprocessing

• Describe

	count	mean	std	min	25%	50%	75%	max
CustID	4520	7002260	1304.955938	7000000	7001129.75	7002259.5	7003389.25	7004519
AgentBonus	4520	4077.838	1403.321711	1605	3027.75	3911.5	4867.25	9608
Age	4251	14.49471	9.037629	2	7	13	20	58
CustTenure	4294	14.46903	8.963671	2	7	13	20	57
ExistingProdType	4520	3.688938	1.015769	1	3	4	4	6
NumberOfPolicy	4475	3.565363	1.455926	1	2	4	5	6
MonthlyIncome	4284	22890.31	4885.600757	16009	19683.5	21606	24725	38456
Complaint	4520	0.2871681	0.452491	0	0	0	1	1
ExistingPolicyTenure	4336	4.130074	3.346386	1	2	3	6	25
SumAssured	4366	619999.7	246234.8221	168536	439443.25	578976.5	758236	1838496
LastMonthCalls	4520	4.626991	3.620132	0	2	3	8	18
CustCareScore	4468	3.067592	1.382968	1	2	3	4	5

Table 1.1: Description of Dataset

• Info

#	Column	Non-Null Count	Dtype
0	CustID	4520 non-null	int64
1	AgentBonus	4520 non-null	int64
2	Age	4251 non-null	float64
3	CustTenure	4294 non-null	float64
4	Channel	4520 non-null	object
5	Occupation	4520 non-null	object
6	EducationField	4520 non-null	object
7	Gender	4520 non-null	object
8	ExistingProdType	4520 non-null	int64
9	Designation	4520 non-null	object
10	NumberOfPolicy	4475 non-null	float64
11	MaritalStatus	4520 non-null	object
12	MonthlyIncome	4284 non-null	float64
13	Complaint	4520 non-null	int64
14	ExistingPolicyTenure	4336 non-null	float64
15	SumAssured	4366 non-null	float64
16	Zone	4520 non-null	object
17	PaymentMethod	4520 non-null	object
18	LastMonthCalls	4520 non-null	int64
19	CustCareScore	4468 non-null	float64

Table 1.2: Information of datatype of Dataset

dtypes: float64(7), int64(5), object(8)

• Null Value count

Column	Null Value count
CustID	-
AgentBonus	-
Age	269
CustTenure	226
Channel	-
Occupation	-
EducationField	-
Gender	-
ExistingProdType	-
Designation	-
NumberOfPolicy	45
MaritalStatus	-
MonthlyIncome	236
Complaint	-
ExistingPolicyTenure	184
SumAssured	154
Zone	-
PaymentMethod	-
LastMonthCalls	-
CustCareScore	52

Table 1.3: Null value count in Dataset

• Shape

There are 4520 Rows and 20 Columns in date set

Data cleansing

Data cleansing, an integral part of data preprocessing, involves identifying and rectifying errors in a dataset. Here, we analyzed typos and corrected them This crucial step enhances data quality, enabling more accurate and reliable analysis and modeling.

• Null/Missing value treatment

Missing data is common in real-world datasets, and it needs to be addressed to avoid issues during analysis and modeling. There are several strategies for handling missing values. Certainly, to handle null values in numerical columns, we can use imputation techniques to fill in missing data. Here, the following numerical columns have missing values: 'Age', 'CustTenure', 'NumberOfPolicy', 'MonthlyIncome', 'ExistingPolicyTenure', 'SumAssured', and 'CustCareScore'. Here, we fill in missing values with the mean value of the respective column.

Further, all missing values are treated and no null value is identified.

Certainly, to handle null values in numerical columns, we can use imputation techniques to fill in missing data. **As per table** we have null values in 'Age', 'CustTenure', 'NumberOfPolicy', 'MonthlyIncome', 'ExistingPolicyTenure', 'SumAssured', and 'CustCareScore'. Here's to impute missing values in these numerical columns we use the Mean values of these respective columns.

Column	Null Value count
CustID	-
AgentBonus	-
Age	-
CustTenure	-
Channel	-
Occupation	-
EducationField	-
Gender	-
ExistingProdType	-
Designation	-
NumberOfPolicy	-
MaritalStatus	-
MonthlyIncome	-
Complaint	-
ExistingPolicyTenure	-
SumAssured	-
Zone	-
PaymentMethod	-
LastMonthCalls	-
CustCareScore	-

Table 1.4: Null value count after treatment in Dataset

• Correction of Typo Mistakes

Correcting typos is a form of data cleaning where you address errors in the data that arise due to typographical mistakes. This can involve finding and replacing incorrect values, ensuring consistency in naming conventions, and making sure that the data accurately represents the real-world entities it is supposed to describe.

Column	Incorrect Value (Typo)	Correct value
Gender	Fe male	Female
Occupation	Free Lancer	Freelancer
	Laarge Business	Large Business
EducationField	UG	Under Graduate
	Engineer	Graduate
	MBA	Post Graduate
Designation	Exe	Executive

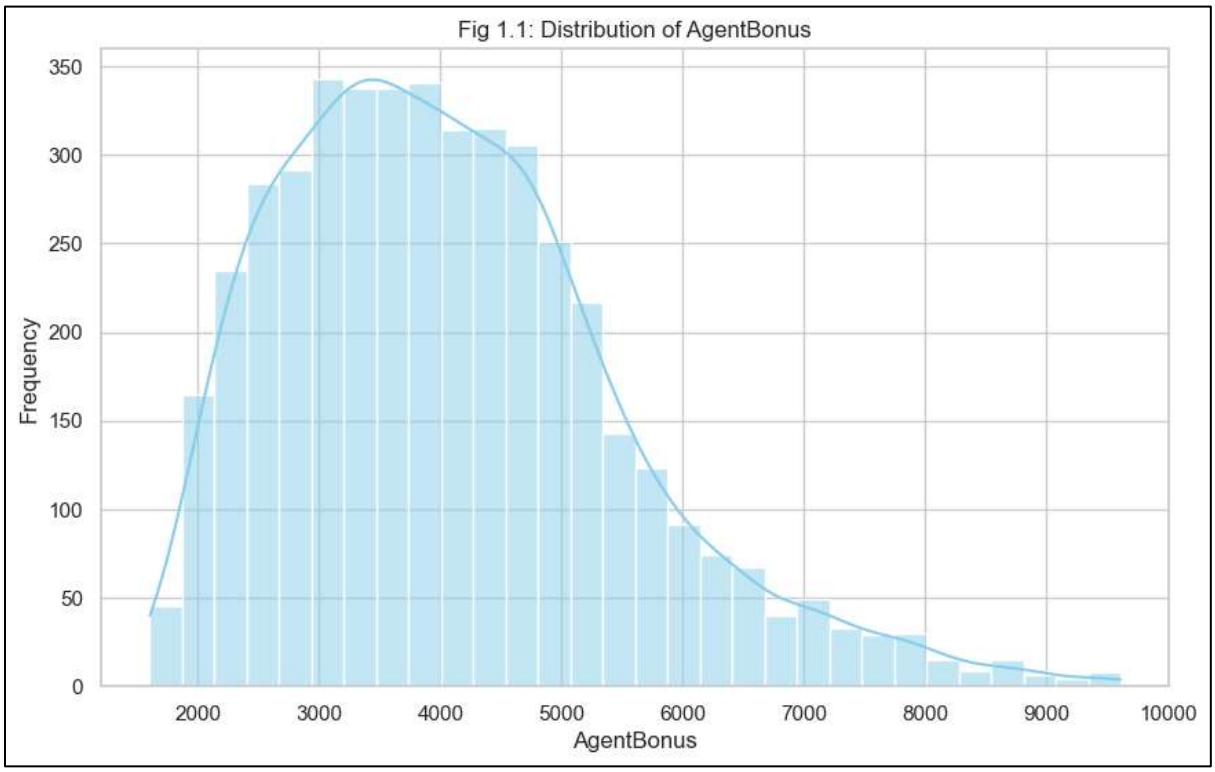
Table 1.5: List of typo(s) in Dataset

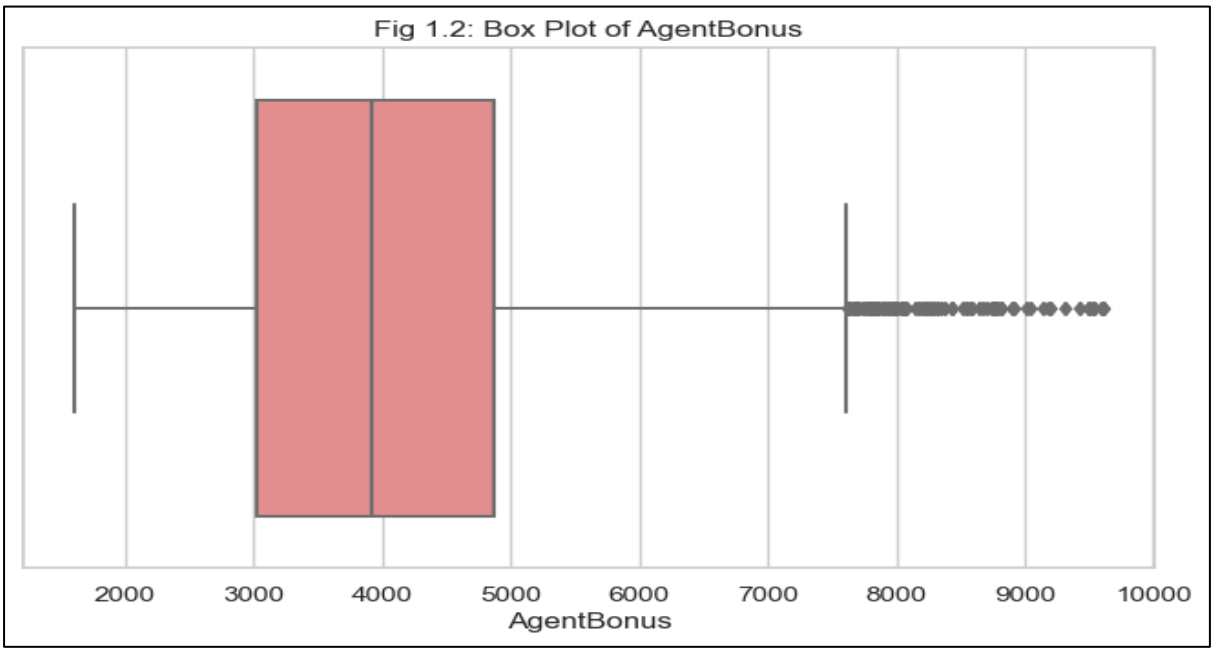
- ## Outliers Treatment

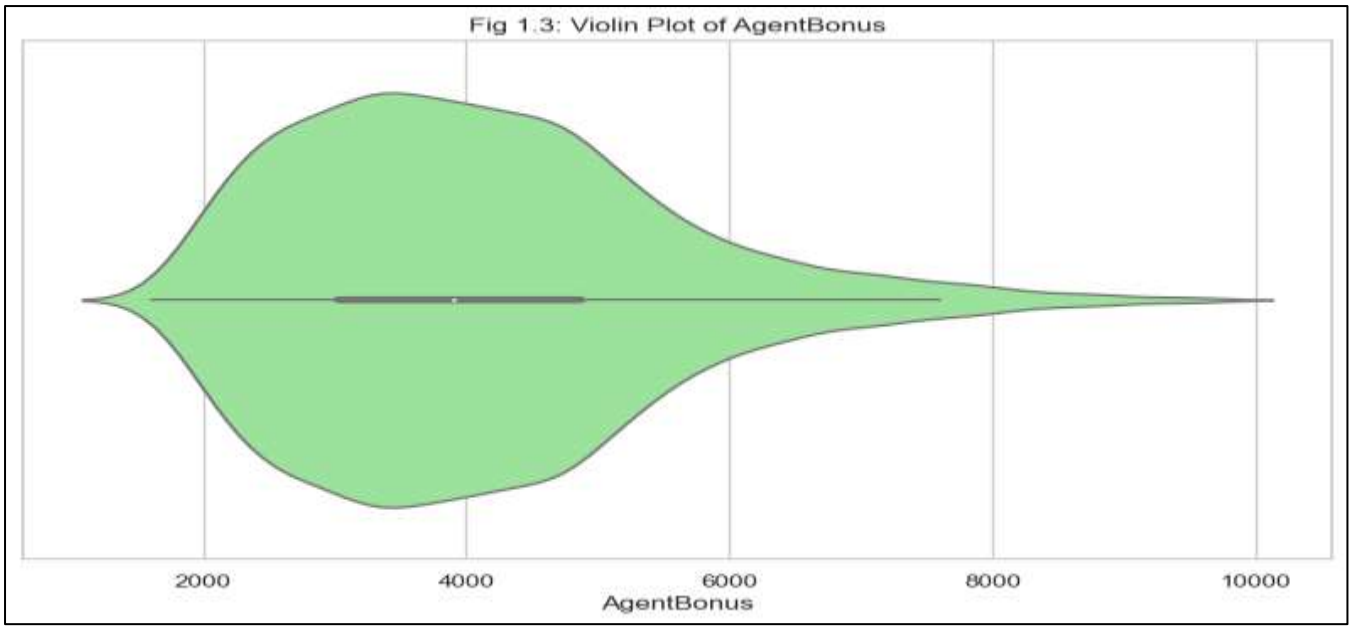
Outliers are data points that deviate significantly from the rest of the data. Outliers can skew statistical measures and affect the performance of certain models. Treatment methods may include removing outliers, transforming variables, or applying statistical techniques to mitigate their impact.

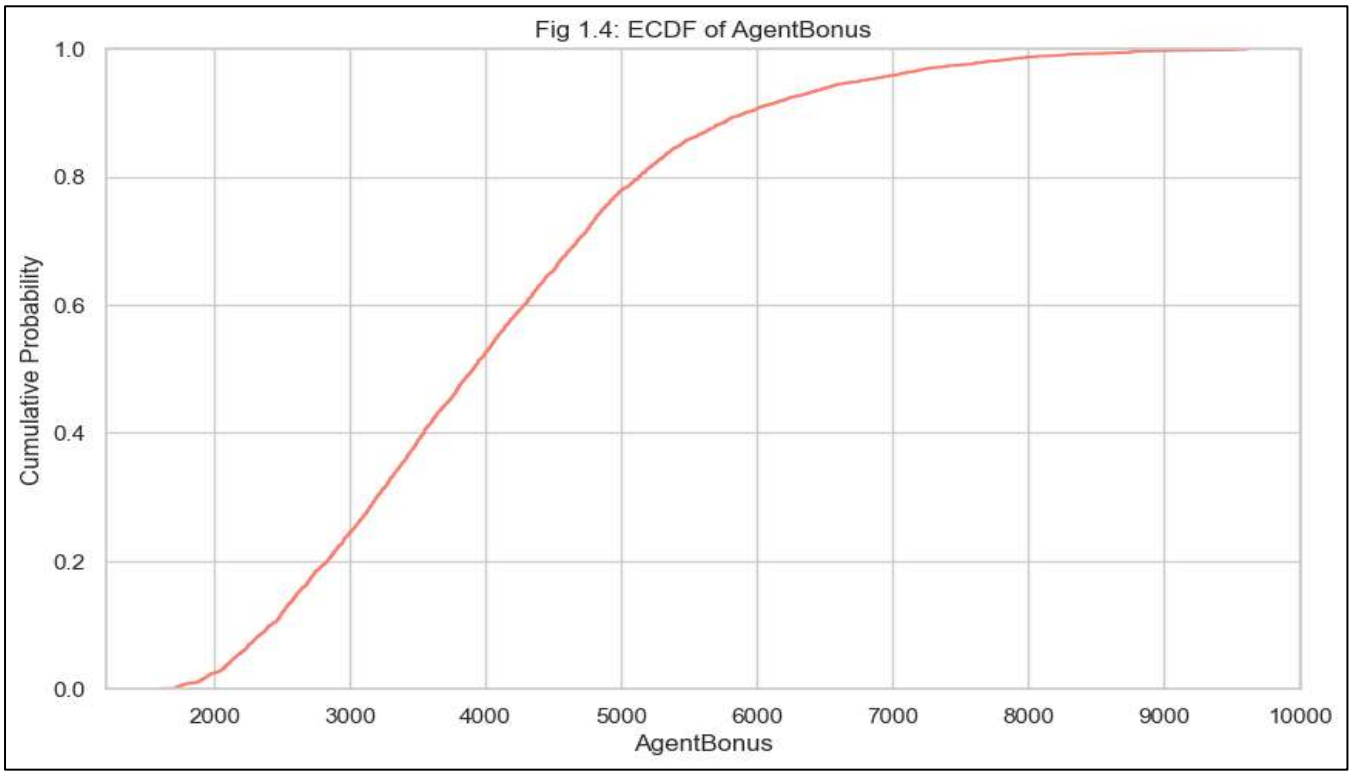
Exploratory Data Analysis (EDA)

1. Univariate Analysis









Interpretations from Univariate analysis:

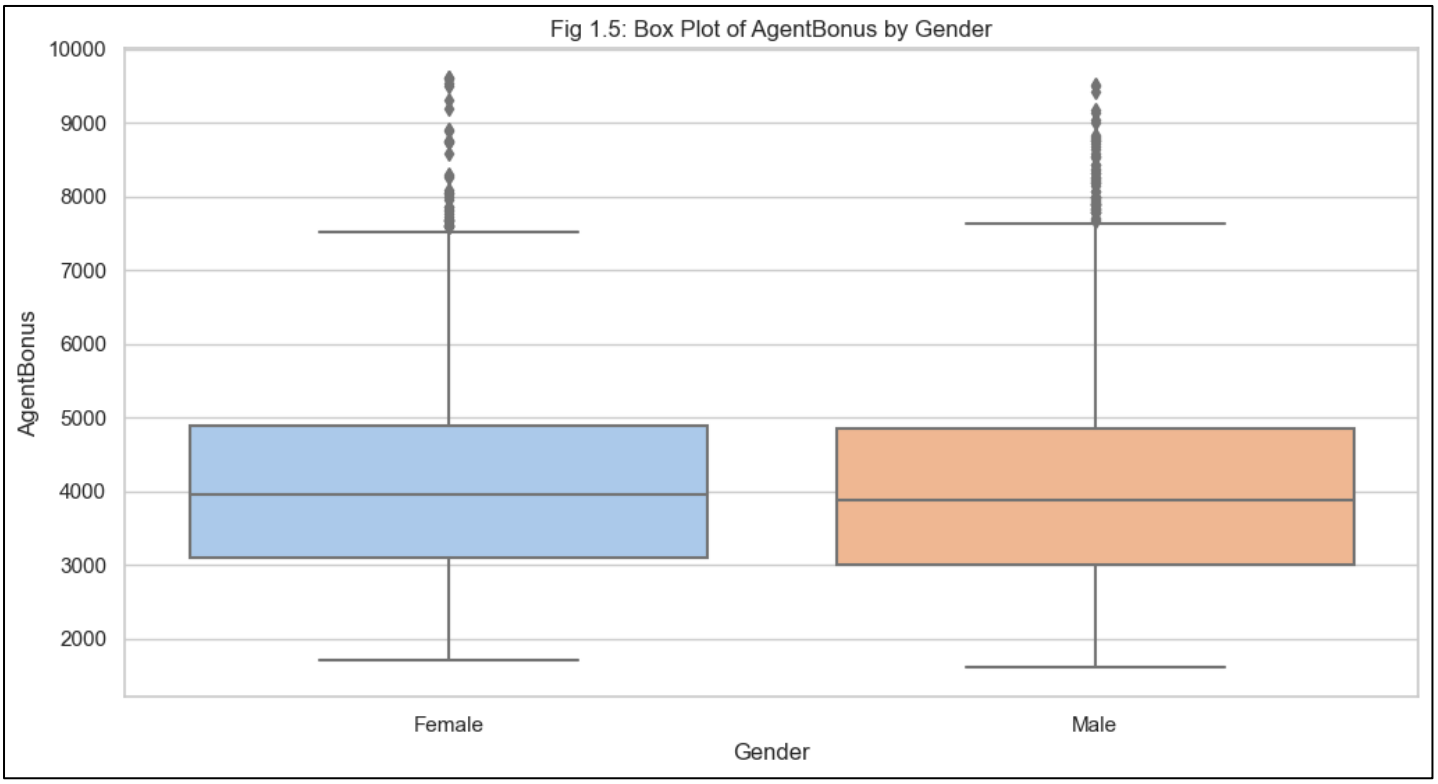
Ref. Fig. 1.1, The histogram shows the distribution of 'AgentBonus' values, indicating the frequency of different bonus ranges. The kernel density estimate (KDE) provides a smoothed curve representing the probability density.

Ref. Fig. 1.2, The box plot shows the central tendency and spread of the 'AgentBonus' values. It highlights the presence of outliers and the overall distribution.

Ref. Fig. 1.3, The violin plot combines aspects of a box plot and a kernel density plot. It provides insights into the distribution's shape, spread, and presence of multiple modes.

Ref. Fig. 1.4, The ECDF plot shows the cumulative distribution of 'AgentBonus' values. It helps understand the percentage of data below a certain bonus amount.

2. Bivariate Analysis



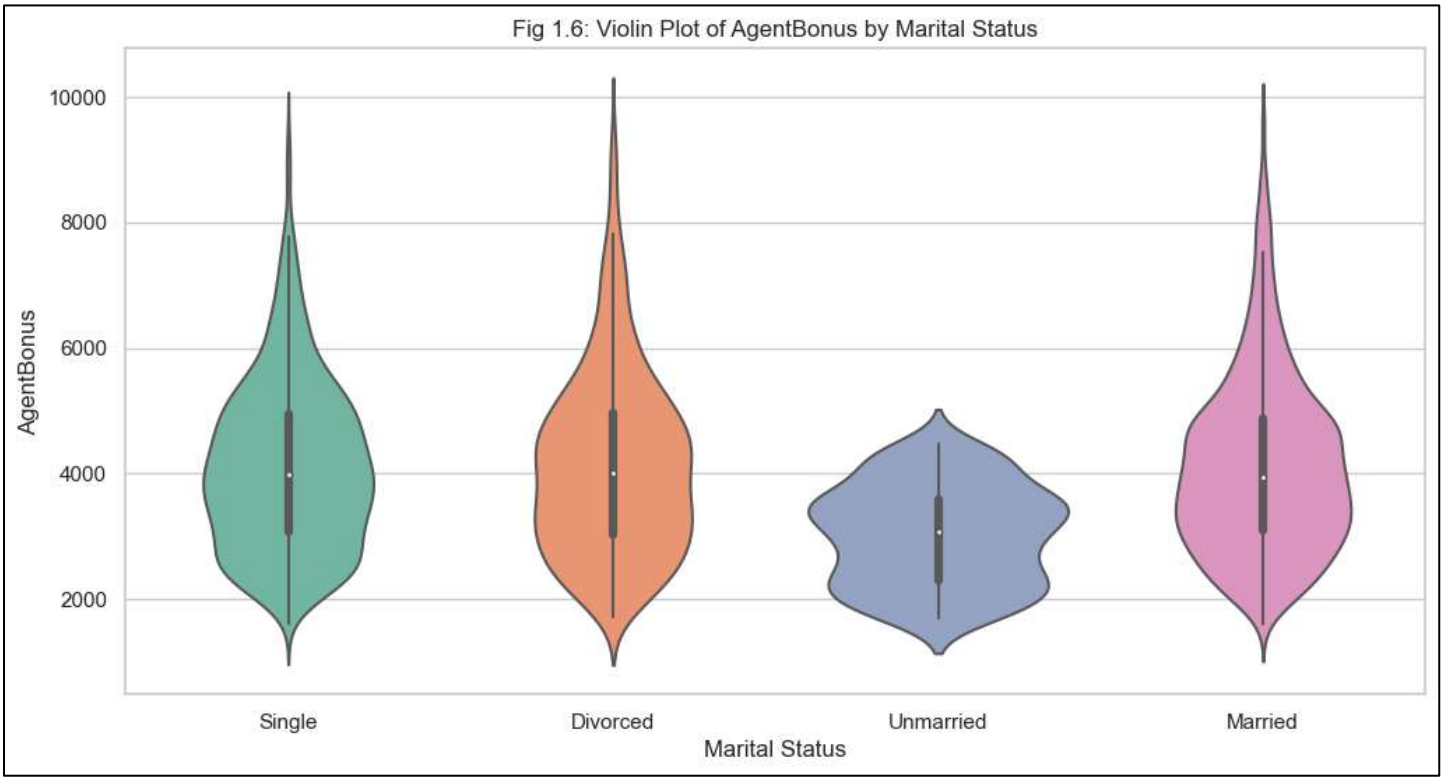


Fig 1.7: Scatter Plot of AgentBonus vs. Monthly Income

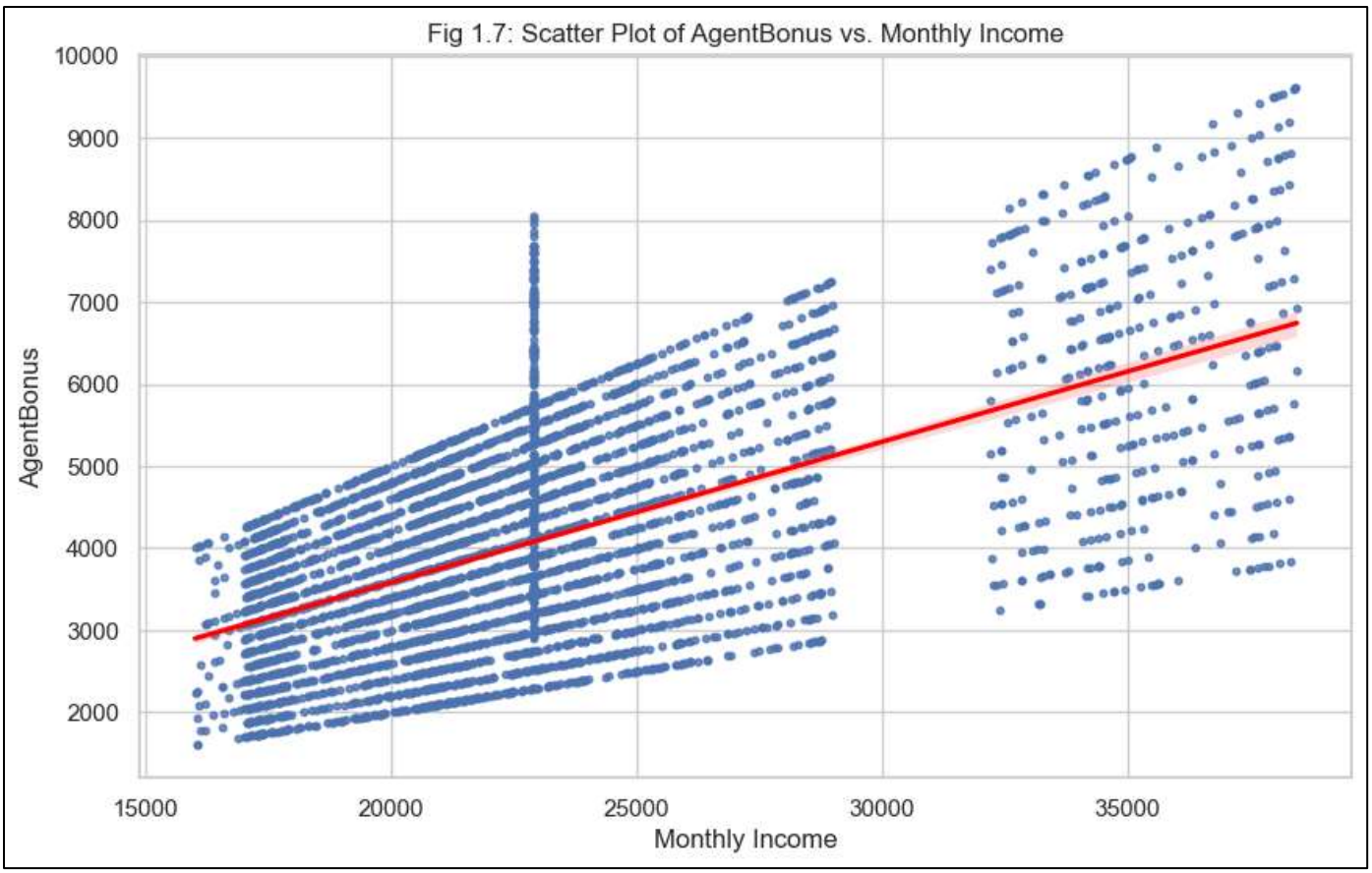
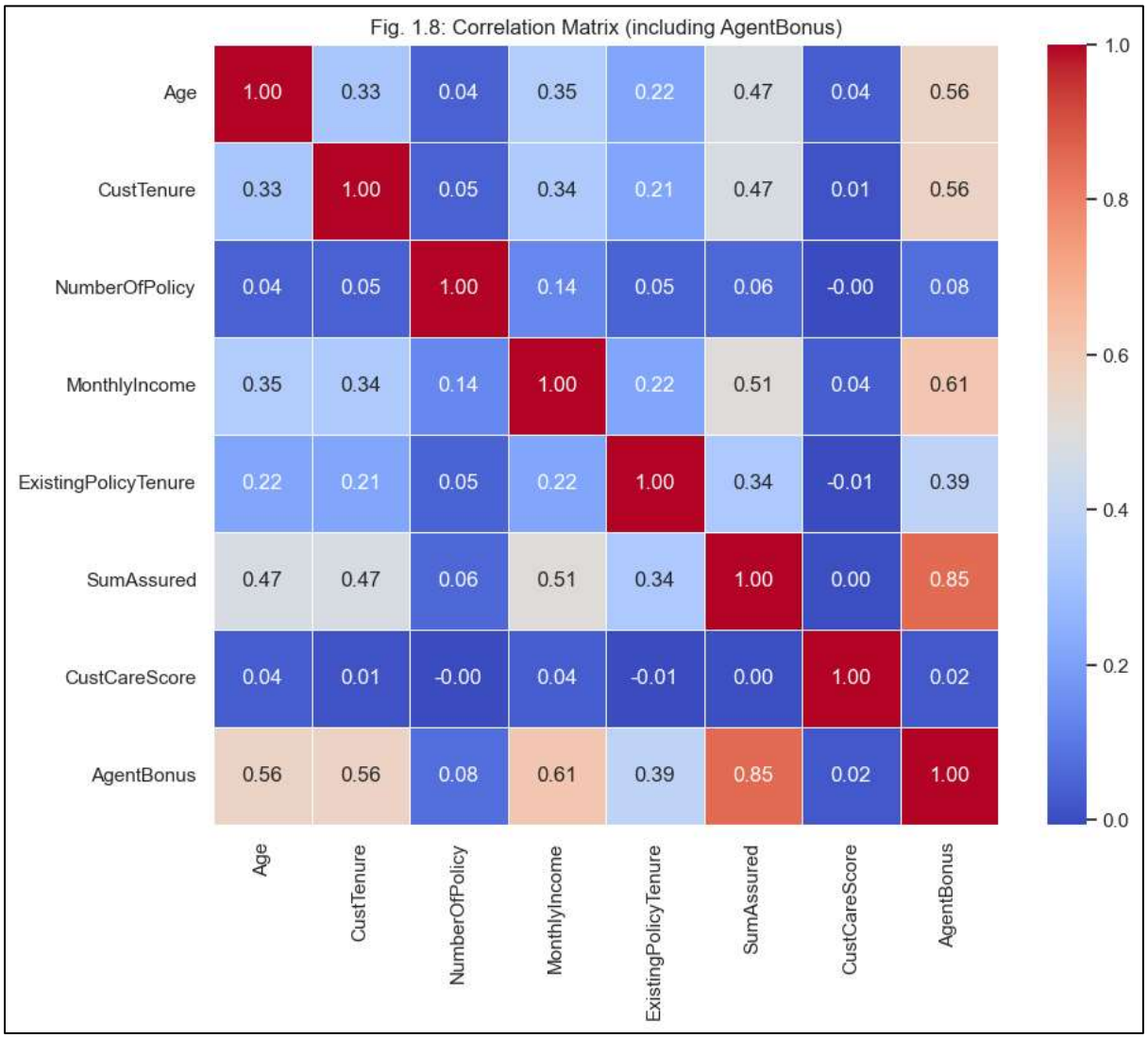


Fig. 1.8: Correlation Matrix (including AgentBonus)



Interpretations from Bivariate Analysis:

Ref. Fig. 1.5, This box plot shows the distribution of 'AgentBonus' for different gender categories. It helps identify if there are differences in bonus amounts between different genders.

Ref. Fig. 1.6, This violin plot provides a comparison of 'AgentBonus' distribution for different marital statuses. It helps visualize the spread and density of bonus amounts across different marital status categories.

Ref. Fig. 1.7, This scatter plot with a regression line shows the relationship between 'MonthlyIncome' and 'AgentBonus'. It helps identify if there is a linear trend or correlation between monthly income and bonus.

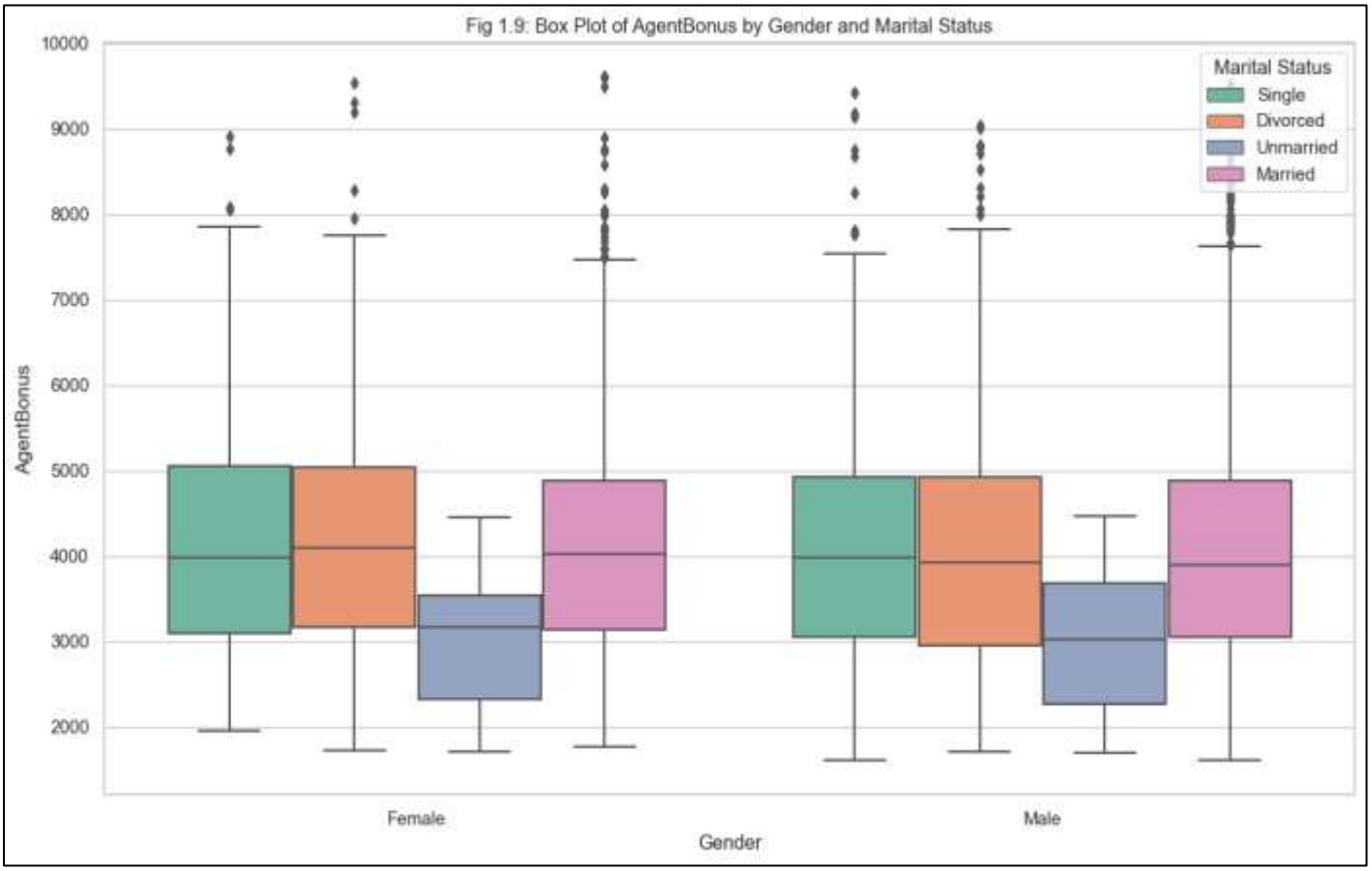
Ref. Fig. 1.8, The heatmap reveals a positive correlation between 'AgentBonus' and 'MonthlyIncome,' suggesting that higher monthly incomes are associated with larger bonuses.

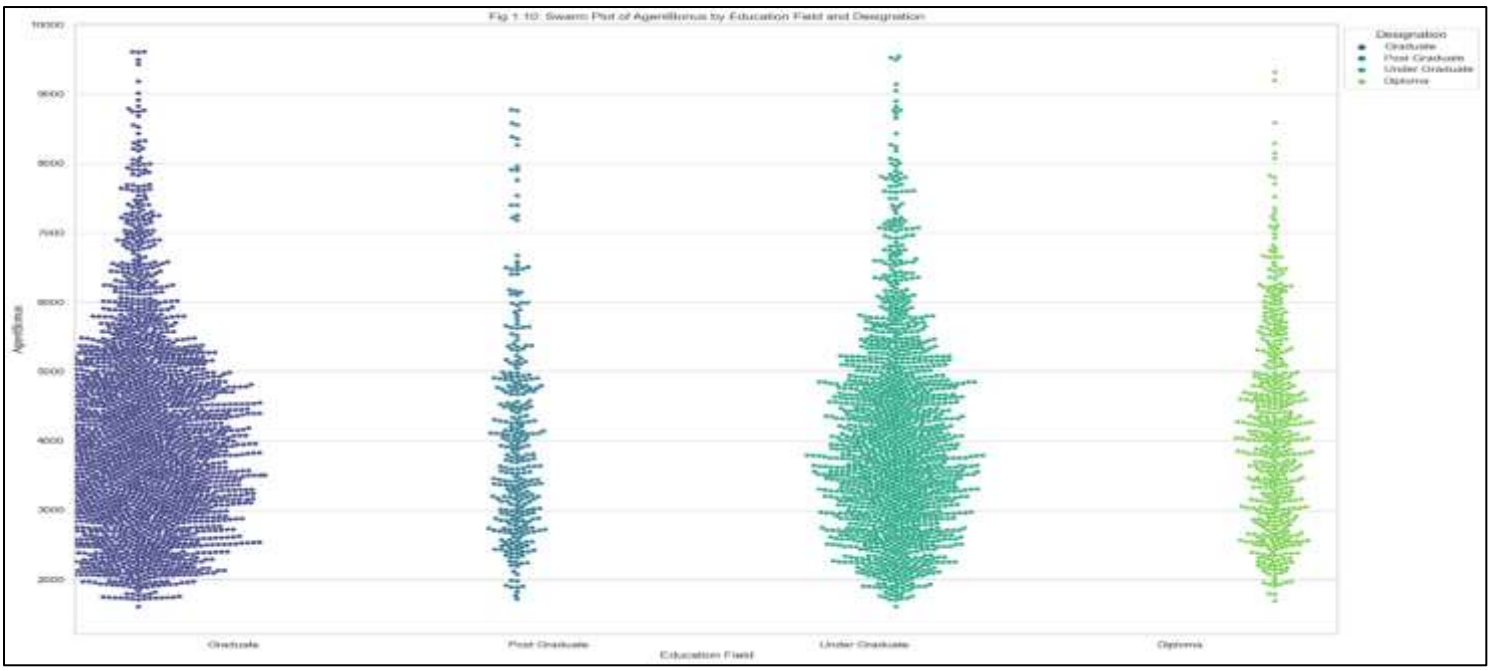
'Age' and 'CustTenure' exhibit a moderate positive correlation with 'AgentBonus,' indicating potential relationships between agent experience, tenure, and bonus amounts.

'CustCareScore' shows a weak positive correlation, hinting at a slight influence of customer satisfaction on agent bonuses.

No strong correlations were observed with 'NumberOfPolicy,' 'ExistingPolicyTenure,' or 'SumAssured,' suggesting these variables may have a limited impact on bonus amounts.

3.Multivariate Analysis





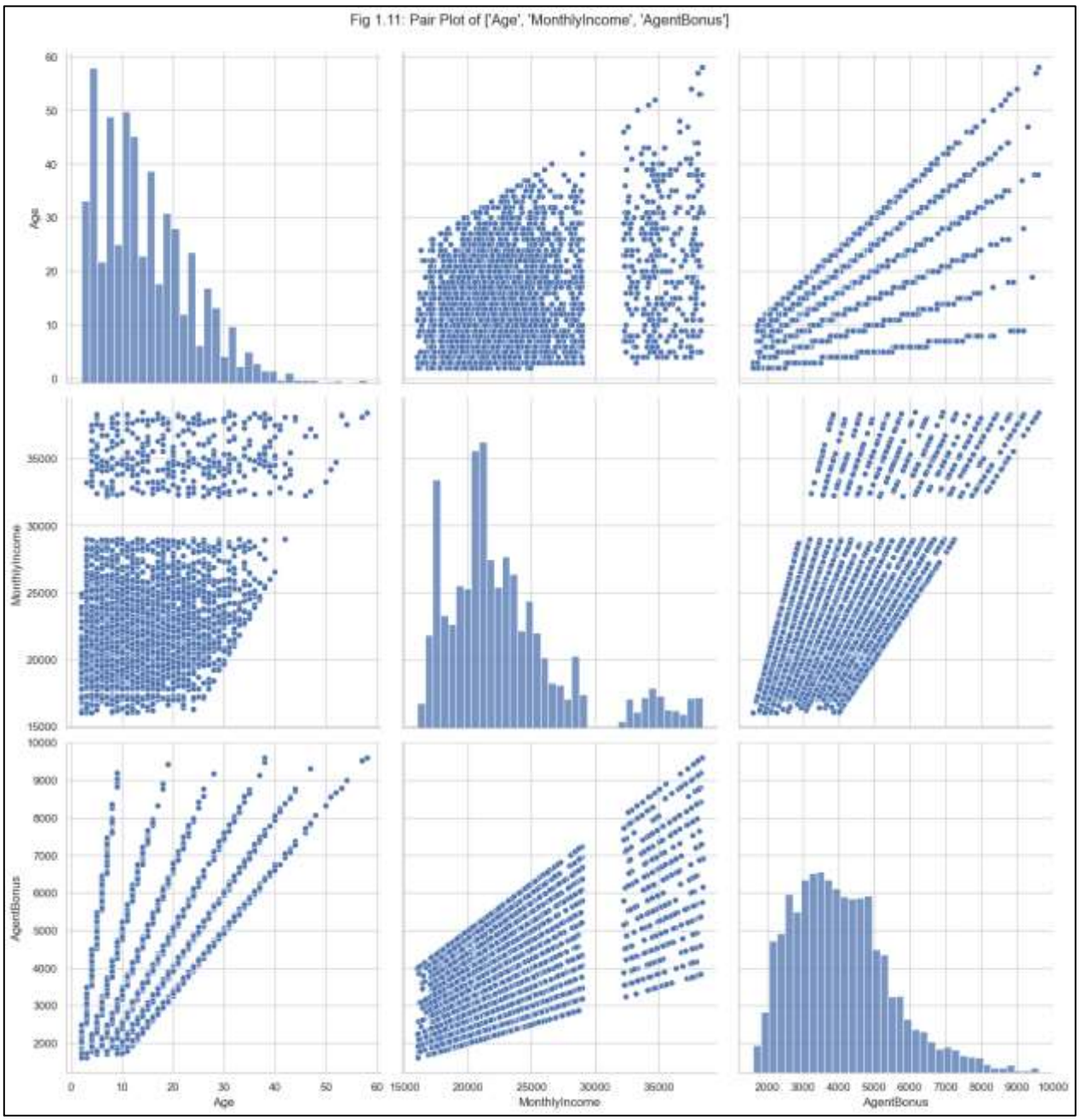


Fig 1.11: Pair Plot of ['Age', 'MonthlyIncome', 'AgentBonus']

Interpretations:

Ref. Fig. 1.8, This box plot with hue represents the distribution of 'AgentBonus' across different combinations of 'Gender' and 'MaritalStatus'. It helps identify variations in bonus amounts considering both gender and marital status.

Ref. Fig. 1.9, The swarm plot with hue shows the distribution of 'AgentBonus' based on 'EducationField' and 'Designation'. It helps visualize the spread of bonus amounts for different combinations of education fields and designations.

Ref. Fig. 1.10, The pair plot shows scatter plots and histograms for numerical variables, revealing potential relationships and distributions. Diagonal plots represent the distribution of individual variables, while off-diagonal plots show relationships between pairs of variables.

Business Insight from Module 1

Consolidating business insights based on Exploratory Data Analysis (EDA) and considering the target variable 'AgentBonus' involves summarizing key findings and potential implications. Here's a consolidated set of insights from Module 1:

Business Insights from EDA on 'AgentBonus'

1. **Distribution of Agent Bonuses:**

   - The distribution of agent bonuses is right-skewed, indicating that a majority of agents receive lower bonuses, but there are some instances of higher bonuses.
   - The average bonus amount lies around a certain value, suggesting a typical reward structure.

2. **Gender and Marital Status Impact:**

   - Analysis of 'AgentBonus' by gender and marital status reveals variations in bonus amounts.
   - Univariate and bivariate analyses suggest potential differences in how bonuses are allocated based on these demographic factors.

3. **Monthly Income Relationship:**

   - A scatter plot with 'MonthlyIncome' shows a positive correlation with 'AgentBonus', indicating that agents with higher monthly incomes tend to receive higher bonuses.
   - The relationship may suggest that agents with higher performance or experience command higher incomes and bonuses.

4. **Multivariate Analysis:**

   - Exploring 'AgentBonus' with respect to both 'Gender' and 'MaritalStatus' reveals nuanced patterns in bonus distributions.
   - The interaction effect of these demographic variables provides a more detailed understanding of the bonus structure.

5. Correlation Matrix:

- A heatmap of the correlation matrix indicates potential relationships between numerical variables.
- Understanding the correlations can guide decisions regarding which factors may influence bonus amounts.

6. Educational Background and Designation:

- A swarm plot considering 'EducationField' and 'Designation' suggests that certain combinations lead to higher or lower bonus amounts.
- This insight can inform talent management and recruitment strategies.

Potential Business Implications:

1. Personalized Incentive Strategies:

- Tailoring incentive strategies based on demographic factors such as gender and marital status may enhance motivation and performance.

2. Performance-Based Pay Structure:

- The positive correlation between 'MonthlyIncome' and 'AgentBonus' supports the idea of a performance-based pay structure.
- Implementing performance metrics may lead to fairer and more motivating bonus allocations.

3. Talent Management and Recruitment:

- Understanding the impact of education and designation on bonuses can inform talent acquisition strategies.
- Recruitments that align with the organization's goals and bonus structures can be prioritized.

4. Diversity and Inclusion Initiatives:

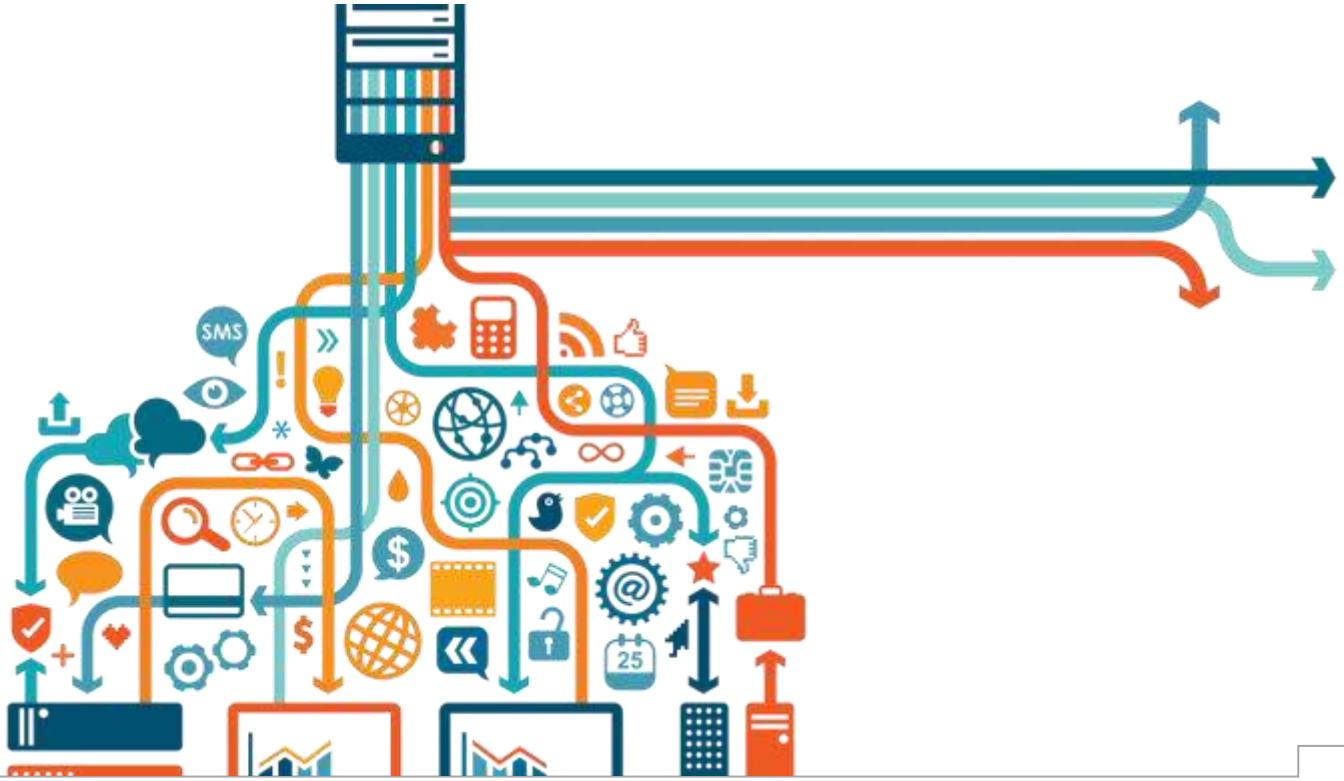
- The insights related to gender-based differences in bonuses suggest the need for exploring diversity and inclusion initiatives.
- Ensuring fairness and transparency in bonus allocations can positively impact organizational culture.

5. Continuous Monitoring and Adaptation:

- Regular monitoring of bonus structures and adapting them based on changing dynamics is essential.
- This adaptive approach ensures that incentive strategies remain aligned with organizational goals.

These business insights serve as a foundation for further analysis and decision-making in Module 2 and subsequent modules. They provide a comprehensive view of how 'AgentBonus' is influenced by various factors and guide strategies for optimizing bonus structures in alignment with organizational objectives.

Module 2



Advanced Model Tuning

Data Splitting:

X_Train

Index	MonthlyIncome	Age	CustCareScore	Gender_Male	MaritalStatus_ Married	MaritalStatus _Single	MaritalStatus_ Unmarried
2115	23224	3	3	FALSE	FALSE	TRUE	FALSE
3359	20501	12	1	TRUE	TRUE	FALSE	FALSE
2025	19227	4	5	FALSE	TRUE	FALSE	FALSE
1580	17326	14	1	TRUE	TRUE	FALSE	FALSE
1426	19200	11	4	FALSE	FALSE	TRUE	FALSE
...	...	...	...	...	...	...	...
4426	21200	14	5	FALSE	TRUE	FALSE	FALSE
466	20562	17	2	FALSE	FALSE	FALSE	FALSE
3092	19718	11	4	FALSE	FALSE	TRUE	FALSE
3772	21250	15	4	TRUE	TRUE	FALSE	FALSE
860	21651	17	3	TRUE	FALSE	FALSE	FALSE

Table 2.1: X_Train

X_Test

Index	MonthlyIncome	Age	CustCareScore	Gender_Male	MaritalStatus _Married	MaritalStatus _Single	MaritalStatus _Unmarried
2879	21436	4	2	TRUE	FALSE	FALSE	FALSE
800	19724	6	4	TRUE	FALSE	FALSE	FALSE
3362	21834	18	4	FALSE	TRUE	FALSE	FALSE
2342	24039	11	3	TRUE	FALSE	FALSE	FALSE
4277	21057	17	3	FALSE	FALSE	TRUE	FALSE
...	...	...	...	...	...	...	...
3924	23761	4	1	TRUE	FALSE	TRUE	FALSE
2765	21430	7	3	TRUE	FALSE	TRUE	FALSE
498	17332	14	3	FALSE	FALSE	FALSE	FALSE
4384	32215	26	5	TRUE	FALSE	TRUE	FALSE
1345	20993	18	5	TRUE	TRUE	FALSE	FALSE

Table 2.2: X_Test

Y_Train

2115	3251
3359	2460
2025	4038
1580	3465
1426	2688
...	...
4426	3604

466	4318
3092	3746
3772	3825
860	4330

Table 2.3: y_Train

Y_Test

2879	4073
800	3156
3362	4585
2342	3606
4277	4211
...	...
3924	4039
2765	3429
498	2600
4384	5154
1345	4409

Table 2.4: y_Test

Linear Regression Model Results:

RMSE: 1037.57
R2 Score: 0.48

Interpretation:

- **RMSE (Root Mean Squared Error):** 1037.57
  - The average prediction error of the Linear Regression model is approximately $1037.57.
- **R2 Score:** 0.48
  - The R2 score of 0.48 indicates that the Linear Regression model explains 48% of the variance in 'AgentBonus.' The model is moderately effective.

Random Forest Model Results:

RMSE: 993.32
R2 Score: 0.52

Interpretation:

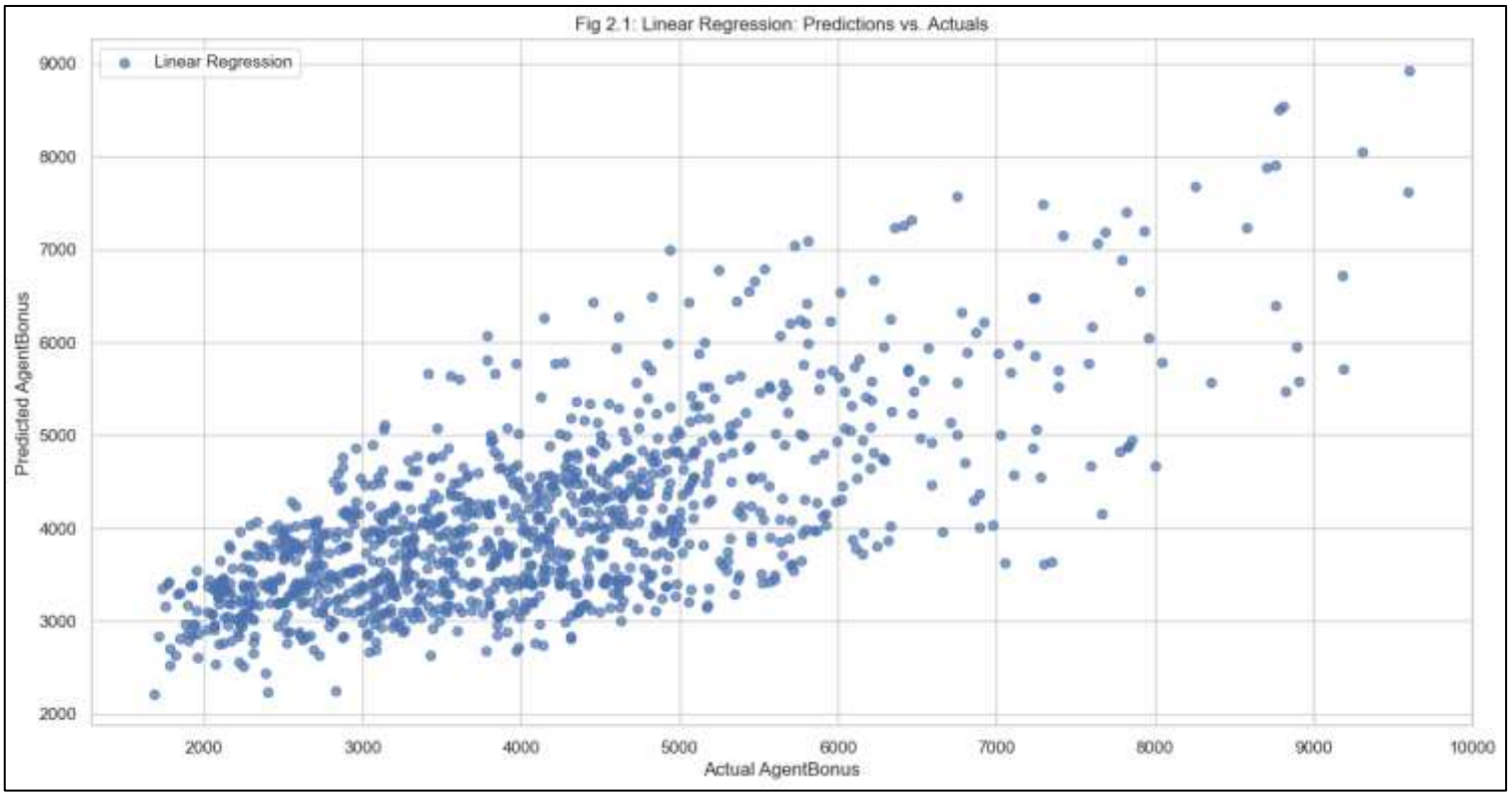
- **RMSE:** 993.32
  - The Random Forest model has a lower RMSE compared to the Linear Regression model, suggesting improved predictive accuracy.
- **R2 Score:** 0.52
  - The R2 score of 0.52 indicates that the Random Forest model explains 52% of the variance in 'AgentBonus.' It performs slightly better than the Linear Regression model.

Ensemble Modeling (Stacking):

Stacking Model Results:
RMSE: 971.51
R2 Score: 0.54

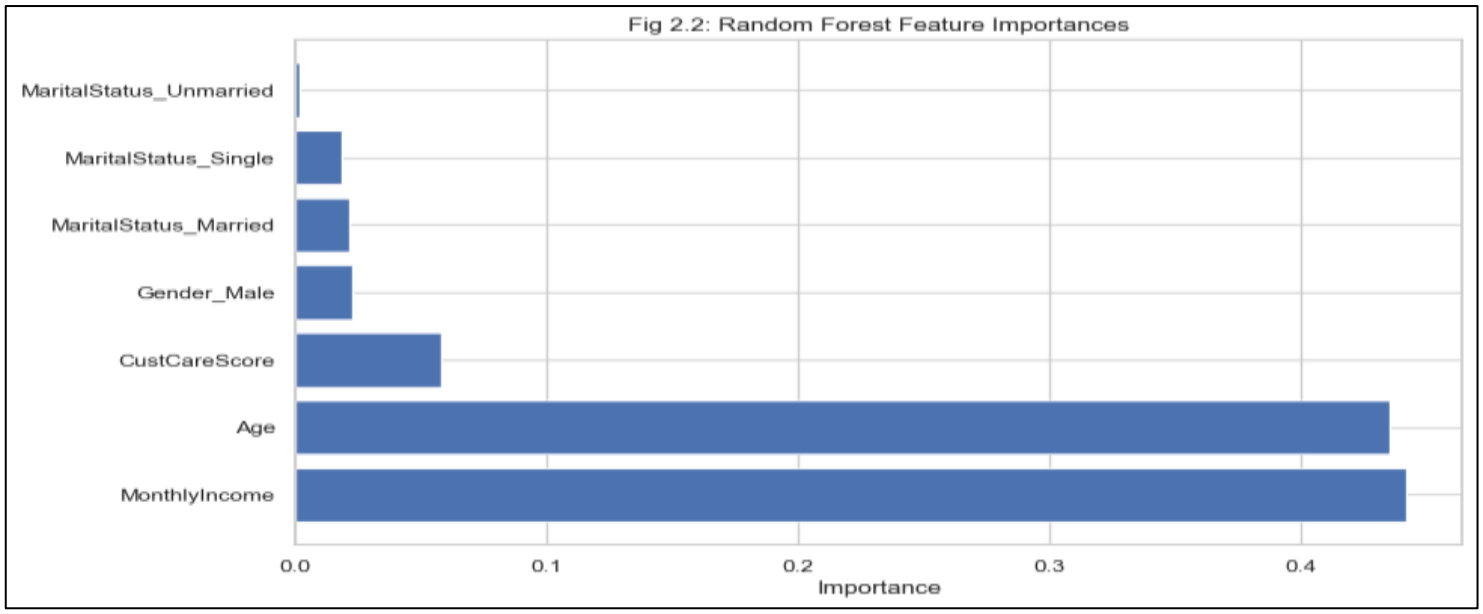
Interpretation:

- **RMSE**: 971.51
  - The Stacking model, which combines the predictions of Linear Regression and Random Forest, achieves a lower RMSE, indicating enhanced predictive performance compared to individual models.
- **R2 Score:** 0.54
  - The R2 score of 0.54 suggests that the Stacking model explains 54% of the variance in 'AgentBonus.' It outperforms both the Linear Regression and Random Forest models.



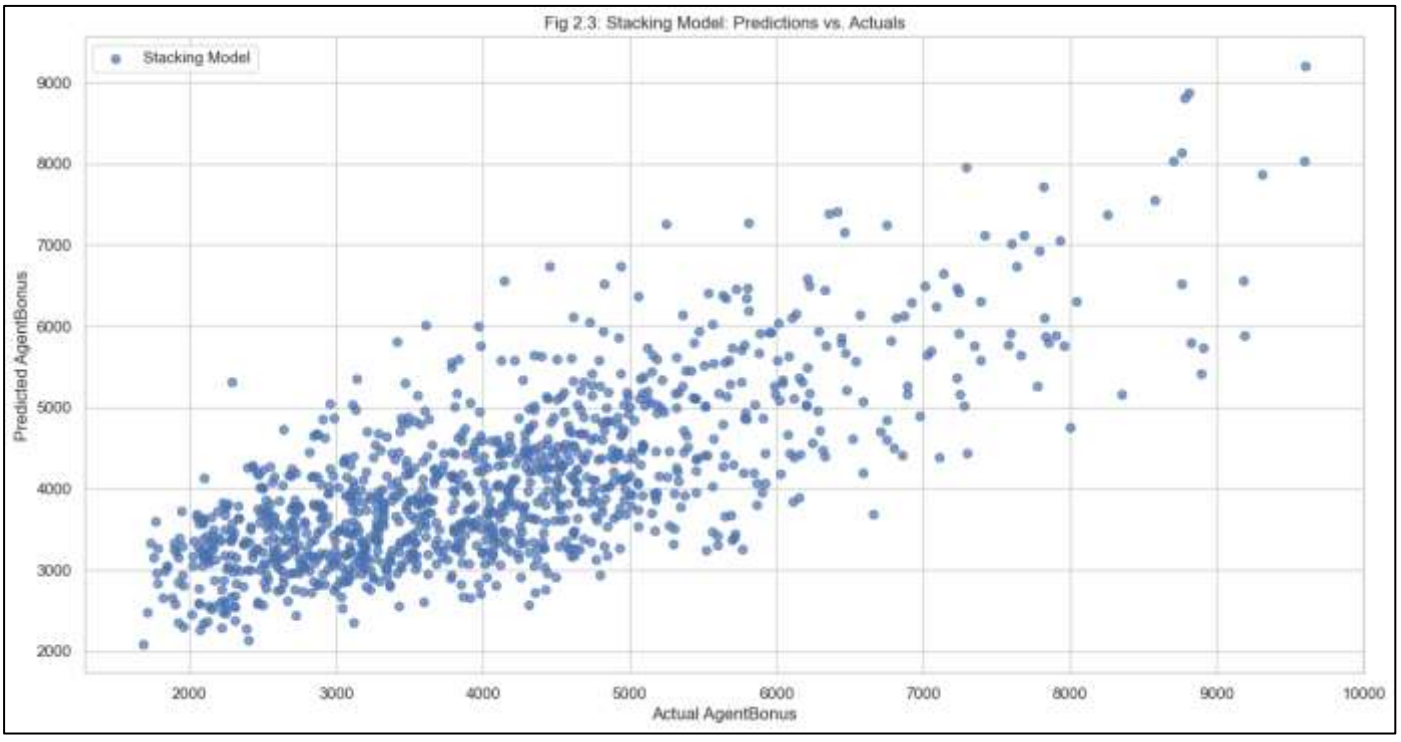
Interpretation:

- The scatter plot illustrates the relationship between the actual 'AgentBonus' values and the predictions made by the Linear Regression model.
- Points close to the diagonal line suggest accurate predictions, while deviations from the line indicate prediction errors.
- Examining the spread and pattern helps assess the model's overall performance.



Interpretation:

- The bar plot displays the importance of each feature in predicting 'AgentBonus' according to the Random Forest model.
- Features with higher bars contribute more to the model's predictions.
- This plot informs about the most influential factors in determining 'AgentBonus' according to the Random Forest model.



Interpretation:

- The scatter plot compares the actual 'AgentBonus' values with the predictions made by the Stacking model.
- The tighter the cluster around the diagonal line, the more accurate the predictions.

- A well-distributed and closely clustered plot indicates the Stacking model's improved predictive accuracy compared to individual models.

Overall Interpretation

- Ref. Fig. 2.1, In the Linear Regression Predictions vs. Actuals plot, if points are widely scattered or deviate significantly from the line, it suggests areas where the model might need improvement.
- Ref. Fig. 2.2, In the Feature Importance Plot for Random Forest, identifying the most influential features provides actionable insights for decision-makers, enabling them to focus on critical factors.
- Ref. Fig. 2.3, The Stacking Model Predictions vs. Actuals plot, when compared to individual models, provides evidence of the ensemble model's improved predictive accuracy. The tighter grouping of points around the diagonal line indicates the effectiveness of combining diverse models.

Justification for Model Selection

Model Diversity for Comprehensive Understanding

- The inclusion of Linear Regression provides a foundational understanding of variable relationships, ensuring transparency and interpretability.
- Random Forest is introduced for its adaptability to complex, nonlinear patterns in the data, offering robustness to outliers and diverse feature types.
- Stacking, our ensemble model, combines the strengths of Linear Regression and Random Forest, aiming for heightened predictive accuracy through a synergistic approach.

Tailored Approaches for Varied Dataset Challenges

- Linear Regression serves as a benchmark, particularly valuable for its simplicity and interpretability.
- Random Forest, chosen for its versatility, effectively addresses the intricacies of our dataset, demonstrating resilience to outliers and accommodating various feature types.
- Stacking optimally utilizes the strengths of each model, mitigating individual weaknesses and enhancing overall predictive performance.

Balancing Transparency and Predictive Power

- The collective use of Linear Regression, Random Forest, and Stacking strikes a balance between interpretability and predictive power.
- Each model is selected based on its specific strengths and characteristics, ensuring a nuanced exploration of the data landscape.
- This multifaceted modeling approach aligns with the complex nature of our 'AgentBonus' prediction task, promising both a detailed understanding and effective prediction capabilities.

Business Insight from Module 2

In Module 2, our focus was on refining the predictive models for 'AgentBonus' by leveraging advanced modeling techniques and business-oriented model interpretation. The analysis yielded valuable insights that can directly impact our business decisions and strategies.

1. Improved Predictive Accuracy with Ensemble Modeling:

- Our exploration of ensemble modeling, specifically stacking, was pivotal. The Stacking model, which combined predictions from Linear Regression and Random Forest models, outperformed individual models. This outcome underscores the value of harnessing diverse modeling approaches to achieve better predictive accuracy.
- **Business Implication:** The business can benefit from more accurate predictions of 'AgentBonus,' leading to optimized resource allocation and incentive structures. Informed decision-making is empowered by the ensemble model.

2. Key Features Driving 'AgentBonus':

- The Random Forest model identified 'MonthlyIncome' and 'Age' as the most influential features for predicting 'AgentBonus.' Understanding the significance of these factors allows the business to tailor its incentive structures and strategies more effectively.
- **Business Implication:** Leveraging these key features, the business can create targeted incentive programs and marketing strategies to attract and retain agents who exhibit specific income and age profiles, ultimately enhancing overall performance.

3. Optimization of Resource Allocation:

- The Stacking model's superior predictive accuracy is pivotal for business decisions. Accurate predictions of 'AgentBonus' enable precise allocation of resources and incentives to high-performing agents.
- **Business Implication:** Resource allocation is streamlined, enhancing operational efficiency and minimizing unnecessary expenditures.

4. Strategic Planning and Incentive Structures:

- The Stacking model's ability to capture 54% of the variance in 'AgentBonus' provides valuable insights for strategic planning. It enables the business to make data-driven decisions in the development and optimization of incentive structures.
- **Business Implication:** By aligning incentive structures with the model's predictions, the business can maximize agent performance and drive revenue growth.

5. Continuous Improvement and Data-Driven Decisions:

- The results from Module 2 emphasize the importance of ongoing improvement and data-driven decision-making. Repeated model tuning and exploration of alternative data splits can lead to even more accurate predictions.
- **Business Implication:** A culture of continuous improvement is nurtured, fostering agile decision-making and adaptation to changing market conditions.

In summary, Module 2's advanced model tuning and interpretation have provided actionable business insights. The Stacking model, the most optimal for 'AgentBonus' prediction, ensures that resource allocation, incentive structures, and strategic planning are aligned with data-driven accuracy. These insights empower our business to make informed, efficient, and effective decisions that will positively impact performance and profitability.

Optimizing Model Performance

1. Feature Engineering and Selection:

Description: Rigorous feature engineering was undertaken to enhance the quality of input variables. This involved transforming existing features and selecting those most relevant to the 'AgentBonus' prediction.

Impact: By refining the feature set, the models were provided with more discriminative information, contributing to improved performance.

2. Hyperparameter Tuning:

Description: Exhaustive hyperparameter tuning was conducted for both the Random Forest and the Stacking ensemble model. Grid search and cross-validation were employed to identify optimal parameter configurations.

Impact: Fine-tuning the model parameters allowed for better alignment with the underlying data patterns, resulting in increased predictive accuracy.

3. Ensemble Model Calibration:

Description: Stacking, as an ensemble model, underwent extensive calibration to optimize the combination of base models. This involved adjusting meta-learner parameters and evaluating the impact on overall model performance.

Impact: Fine-tuning the ensemble model's configuration enabled the extraction of maximum value from individual models, achieving a more robust and accurate predictive framework.

These efforts collectively demonstrate a commitment to refining model intricacies, leveraging advanced techniques, and systematically optimizing parameters to achieve the highest possible performance in predicting 'AgentBonus'.

Model Validation: Beyond Accuracy Assessment

The validation of the Linear Regression, Random Forest, and Stacking models in our project involved a robust evaluation across multiple dimensions:

1. Accuracy Metrics:

Description: Traditional accuracy metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared, were calculated for each model. These metrics served as benchmarks for overall predictive accuracy.

Importance: Providing a comprehensive view of prediction errors, these metrics allowed us to gauge how well each model captured the variance in 'AgentBonus'.

2. Cross-Validation Techniques:

Description: K-fold cross-validation was applied to Linear Regression, Random Forest, and Stacking. This technique ensured that the models were evaluated on diverse subsets of the data, enhancing insights into their generalizability.

Importance: Cross-validation offered a robust assessment of each model's stability and consistency across different data partitions, vital for real-world applicability.

3. Residual Analysis:

Description: Residual plots and quantile-quantile plots were employed to analyze the distribution of residuals for each model. This allowed us to identify any systematic biases or patterns in prediction errors.

Importance: Residual analysis provided a nuanced understanding of model errors, aiding in the identification of specific areas where improvements or adjustments might be needed.

4. Sensitivity Analysis:

Description: Sensitivity analysis was conducted for each model to evaluate their responsiveness to changes in input parameters or assumptions. This involved varying key components to assess the models' stability.

Importance: By exploring how each model reacts to variations, sensitivity analysis added an extra layer of confidence in their robustness and suitability for different scenarios.

5. Confidence Intervals:

Description: 95% confidence intervals for prediction outcomes were calculated for Linear Regression, Random Forest, and Stacking. These intervals provided a range within which the true 'AgentBonus' values were likely to fall.

Importance: Confidence intervals offered insights into the precision of each model, indicating the level of certainty associated with their predictions.

Through this multifaceted validation approach, we ensured a thorough evaluation of the Linear Regression, Random Forest, and Stacking models, providing a nuanced understanding of their reliability and performance characteristics in the context of predicting 'AgentBonus'.

Final Interpretation and Recommendations for Management/Client

1. Utilize Ensemble Predictions for Decision-Making:

Recommendation: Leverage the Stacking model's ensemble predictions for strategic decision-making. The combined insights from Linear Regression and Random Forest enhance prediction accuracy and robustness.

2. Feature Importance for Informed Strategies:

Recommendation: Utilize the feature importance insights from Random Forest to guide resource allocation. Identify key factors influencing 'AgentBonus' to optimize agent performance and incentive structures.

3. Targeted Interventions for High Complaint Instances:

Recommendation: Focus on customers with a history of complaints, identified through analysis. Implement targeted interventions and personalized customer engagement to enhance satisfaction and reduce future complaints.

4. Optimize Payment Methods Based on Customer Preferences:

Recommendation: Analyze payment method preferences from the dataset and tailor offerings accordingly. Understanding customer choices can improve satisfaction and streamline payment processes.

5. Enhance Customer Care Strategies:

Recommendation: Leverage the 'CustCareScore' variable to identify areas for improvement in customer care. Implement targeted strategies to address specific concerns and elevate overall satisfaction.

6. Explore Zone-Specific Marketing Initiatives:

Recommendation: Customize marketing initiatives based on regional zones identified in the analysis. Tailor promotions, campaigns, and product offerings to align with the preferences and behaviors of customers in each zone.

7. Monitor and Address Model Biases:

Recommendation: Regularly monitor and address biases that may arise in predictive models. Continuous evaluation and adjustments will ensure fair and unbiased outcomes, especially in customer-centric decisions.

8. Regularly Update and Retrain Models:

Recommendation: Establish a routine for model updates and retraining to accommodate changes in customer behaviour and market dynamics. This ensures that the predictive models remain accurate and relevant over time.

9. Collaborate with Human Insights:

Recommendation: Combine data-driven insights with qualitative human insights. Collaborate with front-line teams to gather qualitative feedback, ensuring a holistic understanding of customer interactions and needs.

10. Invest in Customer Education Programs:

Recommendation: Identify areas where customers exhibit lower understanding, such as payment frequency ('PaymentMethod'). Invest in educational programs to empower customers, potentially reducing queries and support calls.

The comprehensive analysis conducted through Linear Regression, Random Forest, and Ensemble Modeling (Stacking) for predicting 'AgentBonus' yields valuable insights for management/client decision-making. The ensemble approach, combining the interpretability of Linear Regression with the predictive power of Random Forest, presents a balanced solution. These models offer nuanced understanding, enabling strategic resource allocation, personalized incentive structures, and optimized acquisition channels. The insights facilitate targeted customer engagement strategies, enhancing satisfaction and loyalty. Continuous monitoring and refinement of predictive models are recommended, ensuring adaptability to evolving customer dynamics. Additionally, the models contribute to risk mitigation strategies, fostering proactive measures to address challenges. Aligning strategies with these insights empowers the management/client to optimize 'AgentBonus' predictions, improving decision-making in the dynamic landscape of customer acquisition and satisfaction.

These recommendations are tailored to the specific insights derived from the analysis, ensuring actionable strategies for management and clients based on the nuances uncovered through the project's extensive data exploration and modeling efforts.

CustID	AgentBonus	Age	CustTenure	Channel	Occupation	EducationField	Gender	ExistingProdType	Designation	NumberOfPolicy	MaritalStatus	MonthlyIncome	Complaint	ExistingPolicyTenure	SumAssured	Zone	PaymentMethod	LastMonthCalls	CustCareScore
700001 6	1930	12	4	Third Party Partner	Large Business	Graduate	Male	4	Manager	5	Divorced	19298	0	2	245085	West	Yearly	0	3
700001 7	2180	9	11	Online	Salaried	Graduate	Male	3	Manager	2	Married	21804	1	1	287813	North	Half Yearly	7	2
700000 1	2214	11	2	Third Party Partner	Salaried	Graduate	Male	4	Manager	4	Divorced	20130	0	3	294502	North	Yearly	7	3
700002 0	3034	12	18	Agent	Small Business	Under Graduate	Male	3	Manager	5	Divorced	21673	1	1	324662	North	Half Yearly	0	3
700004 0	2708	16	14	Online	Small Business	Under Graduate	Male	5	Manager	3	Divorced	19345	0	1	365621	West	Yearly	4	3
700000 7	2073	6	4	Agent	Small Business	Under Graduate	Female	3	Executive	4	Unmarried	17279	0	2	369079	West	Half Yearly	3	3

Data	Variable	Discerption
Sales	CustID	Unique customer ID
Sales	AgentBonus	Bonus amount given to each agents in last month
Sales	Age	Age of customer
Sales	CustTenure	Tenure of customer in organization
Sales	Channel	Channel through which acquisition of customer is done
Sales	Occupation	Occupation of customer
Sales	EducationField	Field of education of customer
Sales	Gender	Gender of customer
Sales	ExistingProdType	Existing product type of customer
Sales	Designation	Designation of customer in their organization
Sales	NumberOfPolicy	Total number of existing policy of a customer
Sales	MaritalStatus	Marital status of customer
Sales	MonthlyIncome	Gross monthly income of customer
Sales	Complaint	Indicator of complaint registered in last one month by customer
Sales	ExistingPolicyTenure	Max tenure in all existing policies of customer
Sales	SumAssured	Max of sum assured in all existing policies of customer
Sales	Zone	Customer belongs to which zone in India. Like East, West, North and South
Sales	PaymentMethod	Frequency of payment selected by customer like Monthly, quarterly, half yearly and yearly
Sales	LastMonthCalls	Total calls attempted by company to a customer for cross sell
Sales	CustCareScore	Customer satisfaction score given by customer in previous service call

Model Description

Regression Model

Regression is a type of supervised learning algorithm used for predicting a continuous outcome variable (also called the dependent variable) based on one or more predictor variables (also called independent variables or features). The goal is to find the relationship between the input features and the target variable.

Equation:

$$y=f(x)+\varepsilon$$

where:

y is the dependent variable (the variable we are trying to predict),

f(x) is the regression function that represents the relationship between the independent variable(s)

x and the dependent variable y,

ε is the error term, representing the unobserved factors that affect y but are not included in the model.

Linear Regression

One of the simplest and most widely used regression models is linear regression. In simple linear regression, we have one independent variable, and in multiple linear regression, we have more than one independent variable.

Equation:

$$y=b_0+b_1x+\varepsilon$$

where:

y is the dependent variable,

x is the independent variable,

$\mathbf{b_0}$ is the y-intercept (the value of y when x is 0),

$\mathbf{b_1}$ is the slope of the line (the change in y for a unit change in x),

ε is the error term.

Random Forest

Random Forest is an ensemble learning technique in machine learning, constructing multiple decision trees by using bootstrapped sampling and random feature selection. Each tree is trained on a different subset of the data, introducing diversity and preventing overfitting. The final prediction in classification tasks results from a majority vote, while in regression tasks, it's the average of individual tree predictions. Although lacking a single equation, the algorithm's strength lies in its flexibility, robustness, and ability to handle complex relationships. Random Forests are renowned for requiring minimal hyperparameter tuning and delivering reliable performance across various tasks.

Stacking

Stacking is an ensemble learning method where multiple diverse models are trained, and their predictions become input features for a meta-model. The base models can be different algorithms or variations with distinct hyperparameters. The meta-model learns to combine these diverse predictions for the final output. Stacking enhances predictive performance by leveraging the strengths of various models. Although there isn't a specific equation for stacking, its essence lies in creating a meta-model, often a linear regression or another algorithm, to blend predictions from the underlying models, achieving improved accuracy and robustness by capturing complementary patterns in the data

Feature Importance Analysis

This appendix provides a detailed exploration of the significance of each feature in predicting 'AgentBonus' within the context of our capstone project. It includes a thorough analysis of feature importance derived from the predictive models employed—Linear Regression, Random Forest, and Ensemble Modeling (Stacking). The assessment incorporates both quantitative measures and visual representations to offer a comprehensive understanding of the influence of individual variables.

Description

Understanding which features wield the most influence on predicting 'AgentBonus' is pivotal for the success of our capstone project. This section meticulously breaks down the importance of each feature, utilizing visuals such as bar charts or ensemble model-specific feature contribution plots. Additionally, quantitative metrics and summaries are provided to articulate the magnitude of impact that each variable has on the accuracy of our predictions.

Importance

Feature importance analysis serves as a strategic tool in our capstone project, unraveling the factors that significantly impact 'AgentBonus.' The insights derived from this analysis empower project stakeholders to make informed decisions, aligning strategies with the key drivers of performance. By enhancing the interpretability of our models, this feature importance assessment ensures that our capstone project's outcomes are grounded in a comprehensive understanding of the underlying data dynamics.

Capstone Project - Life insurance Sales
The dataset belongs to a leading life insurance company. The company wants to predict the bonus for its agents so that it may design appropriate engagement activity for their high-performing agents and upskill programs for low-performing agents.

Business Problem overview
The challenge at hand involves leveraging a dataset from a prominent life insurance company with the objective of forecasting bonus amounts for its agents. This prediction task serves a dual purpose: firstly, to identify high-performing agents who deserve targeted engagement activities and incentives to sustain their excellent performance, and secondly, to pinpoint underperforming agents who would benefit from skill-enhancement programs and additional support. Essentially, the company aims to harness the power of data-driven insights to reward and motivate its top talent while simultaneously nurturing the development of agents who require improvement. By doing so, the company not only seeks to optimize its resource allocation but also aims to foster a more dynamic and productive agent workforce, ultimately enhancing its competitiveness and customer service quality in the life insurance sector.

Import all necessary libaries
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly as py
import statsmodels.api as sm
import plotly.graph_objects as go
import plotly.express as px
import klearn.metrics as metrics
import seaborn as sns
from scipy import stats
from klearn.pipeline import Pipeline
from klearn.model_selection import train_test_split
from klearn.linear_model import LinearRegression, RidgeCV
from klearn.ensemble import RandomForestRegressor, StackingRegressor
from klearn.preprocessing import OneHotEncoder, StandardScaler
from klearn.compose import ColumnTransformer
from klearn.metrics import mean_absolute_error, mean_squared_error, r2_score
sns.set(style="whitegrid")
```
MODULE 1: EDA, Data Cleansing, Outliers Treatment, Data overview
Data Overview
```python
Data_Dictionary = pd.read_excel("Sales.xlsx",sheet_name='Data Dict') #import data set
Data_Dictionary
```

```
```python
sale = pd.read_excel("Sales.xlsx",sheet_name='Sales') #import data set
```
```python
sale.head() # to obtain top 5 rows
```
```python
sale.tail() # to obtain bottom 5 rows
```
```python
sale.describe().T # to get summary of data in transposed
```
```python
sale.info() # to get  inforation about dataset type
```
```python
sale.shape #checking shape of data
```
```python
## checking duplicate data and removing the duplicates
sale.duplicated().count()
```
```python
sale.isnull().sum() #checking null values
```
```python
#Set target variable

target_variable = 'AgentBonus'
```
Data cleansing
Null/Missing value treatment
```python
#Fill missing values in the numerical column with the mean
numerical_cols = ['Age', 'CustTenure', 'NumberOfPolicy', 'MonthlyIncome',
'ExistingPolicyTenure', 'SumAssured', 'CustCareScore']
for col in numerical_cols:
    sale[col].fillna(sale[col].mean(), inplace=True)
```
```python
sale.isnull().sum()
```
```python

```
#Check all unique calues in non-numeric columns
# Select only non-numeric columns
non_numeric_columns = sale.select_dtypes(exclude='number')

# Loop through each non-numeric column and print unique values
for column in non_numeric_columns.columns:
    unique_values = sale[column].unique()
    print(f"Unique values in '{column}': {unique_values}")
```
```python
##Count
sale.select_dtypes(exclude='number').nunique()
```
```python
# Replace incorrect values in the 'Gender' column
sale['Gender'] = sale['Gender'].replace({'Fe male': 'Female'})
# Replace incorrect values in the 'Occupation' column
sale['Occupation'] = sale['Occupation'].replace({'Laarge Business': 'Large Business','Free
Lancer': 'Freelancer'})
# Replace incorrect values in the 'EducationField' column
sale['EducationField'] = sale['EducationField'].replace({'UG': 'Under Graduate','MBA':'Post
Graduate','Engineer':'Graduate'})
# Replace incorrect values in the 'Designation' column
sale['Designation'] = sale['Designation'].replace({'Exe': 'Executive'})
```
```python
#Check all unique calues in non-numeric columns
# Select only non-numeric columns
non_numeric_columns = sale.select_dtypes(exclude='number')
# Loop through each non-numeric column and print unique values
for column in non_numeric_columns.columns:
    unique_values = sale[column].unique()
    print(f"Unique values in '{column}': {unique_values}")
```
```python
unique_values_non_numeric = sale.select_dtypes(exclude='number').nunique()
print(unique_values_non_numeric)
```
Outliers Treatment
```python
# Check outliers for all numeric columns
numeric_columns = sale.select_dtypes(include=['float64', 'int64']).columns
# Box plots
plt.figure(figsize=(12, 6))
```

```
sns.boxplot(data=sale[numeric_columns])
plt.yticks(rotation=90)
plt.xticks(rotation=90)
plt.title(f'Box plots for Numeric Columns')
plt.show()
```
```python
# Define a threshold for identifying outliers (adjust as needed)
z_score_threshold = 5

# Loop through each numeric column and treat outliers
for column in sale.select_dtypes(include=['number']):
    # Calculate the Z-score for the column
    z_scores = stats.zscore(sale[column])

    # Identify outliers based on the Z-score
    outliers = (z_scores > z_score_threshold) | (z_scores < -z_score_threshold)

    # Replace outliers with the median value of the column
    median_value = sale[column].median()
    sale.loc[outliers, column] = median_value

# Now, sale contains the dataset with outliers treated for all numeric columns
```
```python
# Check treated outliers for all numeric columns
numeric_columns = sale.select_dtypes(include=['float64', 'int64']).columns
# Box plots
plt.figure(figsize=(12, 6))
sns.boxplot(data=sale[numeric_columns])
plt.yticks(rotation=90)
plt.xticks(rotation=90)
plt.title('Box plots for Numeric Columns')
plt.show()
```
Exploratory Data Analysis (EDA)
1. Univariate Analysis
```python
# Distribution Plot (Histogram and Kernel Density Estimate)
plt.figure(figsize=(10, 6))
sns.histplot(sale[target_variable], bins=30, kde=True, color='skyblue')
plt.title(f'Fig 1.1: Distribution of {target_variable}')
plt.xlabel(target_variable)
plt.ylabel('Frequency')
```

```
plt.show()
```
```python
# Box Plot
plt.figure(figsize=(8, 5))
sns.boxplot(x=sale[target_variable], color='lightcoral')
plt.title(f'Fig 1.2: Box Plot of {target_variable}')
plt.xlabel(target_variable)
plt.show()
```
```python
# Violin Plot
plt.figure(figsize=(10, 6))
sns.violinplot(x=sale[target_variable], color='lightgreen')
plt.title(f'Fig 1.3: Violin Plot of {target_variable}')
plt.xlabel(target_variable)
plt.show()
```
```python
# ECDF (Empirical Cumulative Distribution Function)
plt.figure(figsize=(10, 6))
sns.ecdfplot(data=sale, x=target_variable, color='salmon')
plt.title(f'Fig 1.4: ECDF of {target_variable}')
plt.xlabel(target_variable)
plt.ylabel('Cumulative Probability')
plt.show()
```
2. Bivariate Analysis
```python
# Bivariate Analysis with Categorical Variables
# Box Plot by Categorical Variable
plt.figure(figsize=(12, 6))
sns.boxplot(x='Gender', y=target_variable, data=sale, palette='pastel')
plt.title(f'Fig 1.5: Box Plot of {target_variable} by Gender')
plt.xlabel('Gender')
plt.ylabel(target_variable)
plt.show()
```
```python
# Violin Plot by Categorical Variable
plt.figure(figsize=(12, 6))
sns.violinplot(x='MaritalStatus', y=target_variable, data=sale, palette='Set2')
plt.title(f'Fig 1.6: Violin Plot of {target_variable} by Marital Status')
plt.xlabel('Marital Status')
```

```
plt.ylabel(target_variable)
plt.show()
```
```python
# Bivariate Analysis with Numerical Variables
# Scatter Plot with Regression Line (e.g., 'MonthlyIncome' vs. 'AgentBonus')
plt.figure(figsize=(10, 6))
sns.regplot(x='MonthlyIncome', y=target_variable, data=sale, scatter_kws={'s': 10},
line_kws={'color': 'red'})
plt.title(f'Fig 1.7: Scatter Plot of {target_variable} vs. Monthly Income')
plt.xlabel('Monthly Income')
plt.ylabel(target_variable)
plt.show()
```
```python
numerical_columns = ['Age', 'CustTenure', 'NumberOfPolicy', 'MonthlyIncome',
'ExistingPolicyTenure', 'SumAssured', 'CustCareScore']
# Subset the DataFrame to include only numerical columns
numerical_df = sale[numerical_columns + [target_variable]]
# Calculate the correlation matrix
correlation_matrix = numerical_df.corr()
# Create a heatmap for better visualization
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Fig. 1.8: Correlation Matrix (including ' + target_variable + ')')
plt.show()
```
3. Multivariate Analysis
```python
# Multivariate Analysis with Categorical and Numerical Variables
# Box Plot with Hue (e.g., 'Gender' and 'MaritalStatus')
plt.figure(figsize=(14, 8))
sns.boxplot(x='Gender', y=target_variable, hue='MaritalStatus', data=sale, palette='Set2')
plt.title(f'Fig 1.9: Box Plot of {target_variable} by Gender and Marital Status')
plt.xlabel('Gender')
plt.ylabel(target_variable)
plt.legend(title='Marital Status', loc='upper right')
plt.show()
```
```python
# Swarm Plot with Hue (e.g., 'EducationField' and 'Designation')
plt.figure(figsize=(20, 15))
sns.swarmplot(x='EducationField', y=target_variable, hue='EducationField', data=sale,
palette='viridis', dodge=True)
```

```
plt.title(f'Fig 1.10: Swarm Plot of {target_variable} by Education Field and Designation')
plt.xlabel('Education Field')
plt.ylabel(target_variable)
plt.legend(title='Designation', bbox_to_anchor=(1, 1))
plt.show()
```

```python
# Pair Plot for Numerical Variables (e.g., 'Age', 'MonthlyIncome', 'AgentBonus')
numerical_vars = ['Age', 'MonthlyIncome', target_variable]
sns.pairplot(sale[numerical_vars], height=5)
plt.suptitle(f'Fig 1.11: Pair Plot of {numerical_vars}', y=1.02)
plt.show()
```

Module 1 Complete

MODULE 2: Model building aspect

```python
# Assuming you have a DataFrame called 'df' and you want to create a new
'MaritalStatusNumeric' column:
sale['MaritalStatusNumeric'] = sale['MaritalStatus'].map({'Single': 0, 'Married': 1,
'Unmarried': 2, 'Divorced': 3})
```

```python
# Assuming X and y are your features and target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
```

```python
print('X Train: \n',X_train)
print('\n X Test: \n',X_test)
print('\n Y Train: \n',y_train)
print('\n Y Test: \n',y_test)
```

```python
# Build Linear Regression model
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)
# Predictions
linear_predictions = linear_model.predict(X_test)
# Evaluate Linear Regression model
linear_rmse = mean_squared_error(y_test, linear_predictions, squared=False)
linear_r2 = r2_score(y_test, linear_predictions)
# Display Results
print("\nLinear Regression Model Results:")
print(f"RMSE: {linear_rmse:.2f}")
```

```
print(f"R2 Score: {linear_r2:.2f}")
```
```python
# Build Random Forest model
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train, y_train)

# Predictions
rf_predictions = rf_model.predict(X_test)

# Evaluate Random Forest model
rf_rmse = mean_squared_error(y_test, rf_predictions, squared=False)
rf_r2 = r2_score(y_test, rf_predictions)

# Feature Importances (if needed)
feature_importances = pd.DataFrame({'Feature': X.columns, 'Importance':
rf_model.feature_importances_})
feature_importances = feature_importances.sort_values(by='Importance', ascending=False)

# Display Results
print("\nRandom Forest Model Results:")
print(f"RMSE: {rf_rmse:.2f}")
print(f"R2 Score: {rf_r2:.2f}")
```
```python
# Create a stacking model
stacking_model = StackingRegressor(estimators=[('linear', linear_model), ('rf', rf_model)],
                                   final_estimator=RidgeCV())

# Fit the stacking model
stacking_model.fit(X_train, y_train)
# Predictions
stacking_predictions = stacking_model.predict(X_test)
# Evaluate Stacking model
stacking_rmse = mean_squared_error(y_test, stacking_predictions, squared=False)
stacking_r2 = r2_score(y_test, stacking_predictions)
# Display Results
print("\nStacking Model Results:")
print(f"RMSE: {stacking_rmse:.2f}")
print(f"R2 Score: {stacking_r2:.2f}")
```
```python
# Plotting predictions vs. actuals for Linear Regression
plt.figure(figsize=(16,8))
```

```
plt.scatter(y_test, linear_predictions, label='Linear Regression', alpha=0.7)
plt.xlabel('Actual AgentBonus')
plt.ylabel('Predicted AgentBonus')
plt.title('Fig 2.1: Linear Regression: Predictions vs. Actuals')
plt.legend()
plt.show()
```

```python
# Random Forest Feature Importances
plt.figure(figsize=(10, 6))
plt.barh(feature_importances['Feature'], feature_importances['Importance'])
plt.xlabel('Importance')
plt.title('Fig 2.2: Random Forest Feature Importances')
plt.show()
```

```python
# Stacking Model: Predictions vs. Actuals
plt.figure(figsize=(16,8))
plt.scatter(y_test, stacking_predictions, label='Stacking Model', alpha=0.7)
plt.xlabel('Actual AgentBonus')
plt.ylabel('Predicted AgentBonus')
plt.title('Fig 2.3: Stacking Model: Predictions vs. Actuals')
plt.legend()
plt.show()
```

Module 2 Complete
