Data Storm v6.0

— Preliminary Round —

Organized By

Rotaract Club of University of Moratuwa

May 7, 2025

Powerd By

OCTAVE

Team Name: TeamTitans

Team members:

1. Subhash Bandara Ekanayake
2. Gihan Lakmal
3. Tharindu Damruwan

EDA part – Exploratory Data Analysis

1. What are the key metrics and distributions in the dataset? (Summary statistics)

- **Overview of the dataset.**

```
train_df.shape #number of rows and columns
```

```
train_df.info() #check for base infos within each column datatypes
```

```
train_df.describe() #summary statistics
```

⇒ The dataset contains **15,308 records** with 22 columns, of which 18 are numerical and 4 are categorical (agent_code, agent_join_month, first_policy_sold_month, year_month).

⇒ The **non-null count** for all columns is 15,308, indicating no missing values in the dataset.

⇒ The majority of agents are relatively young (between 20–60 years old), potentially indicating a mix of experienced and new agents in the industry.

⇒ Features like net_income and ANBP_value are right-skewed, meaning most agents perform below average, with a few exceptional agents driving higher sales or profits.

- **Performing Data preprocessing**

1. Datetime Conversion:

- We **converted** the columns year_month, agent_join_month, and first_policy_sold_month to the **datetime**format using pd.to_datetime(). This ensures that the date columns are in a proper datetime format for time-based analysis.

2. Checking for Missing Data:

- We checked for **null values** using the isnull().sum() method. It was confirmed that **there are no missing values**in the dataset, meaning no further imputation or handling of missing data is needed.

3. Checking for Duplicates:

- We checked for **duplicate rows** using the duplicated().sum() method. It was confirmed that **no duplicate rows**are present, so there’s no need for further cleaning in terms of duplicates.

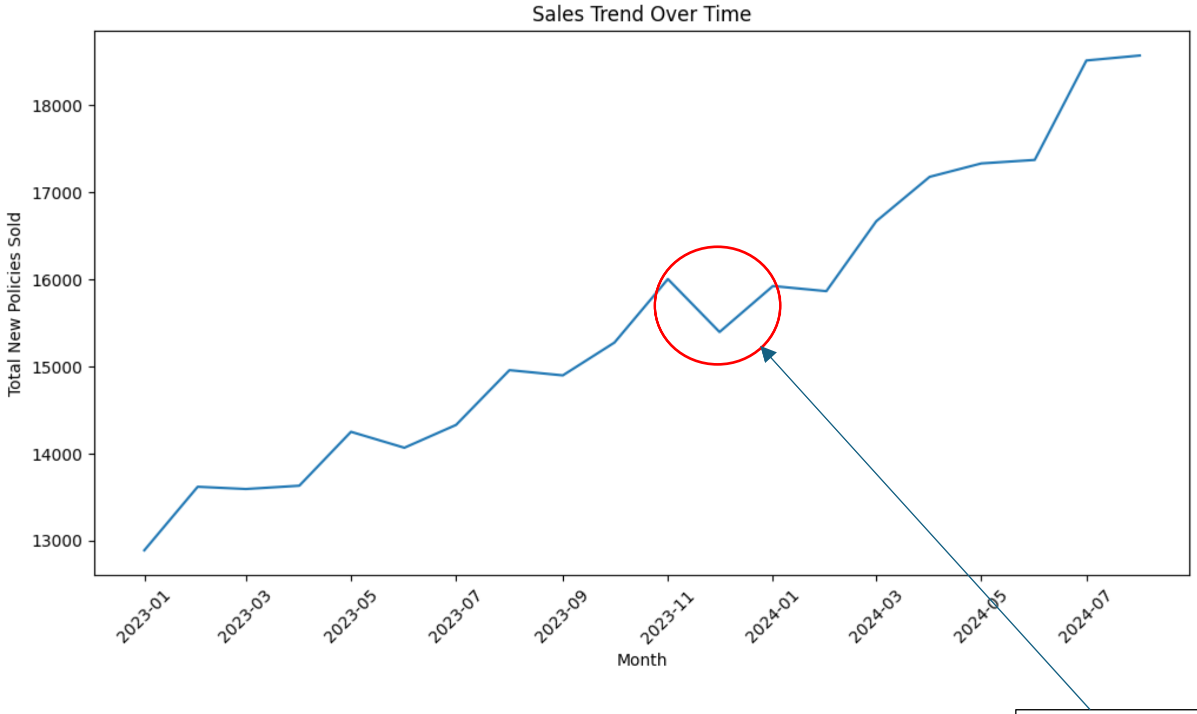
4. Counting Unique Values:

- We counted the number of **unique values** for each column using nunique(). This gives insight into the cardinality of each feature:
  - For instance, agent_code has **905 unique values**, while year_month has **20 unique values**, indicating the monthly data spans 20 distinct months.
  - For categorical features like unique_proposals_last_7_days, unique_quotations_last_7_days, and unique_customers_last_7_days, they have a small number of unique values (4–7), likely indicating specific counts or categories.

2. How do sales patterns vary by month, and are there unexpected drops or spikes performance?

2.1. Sales Trend Visualization:

- We **aggregated sales data** by month using groupby('year_month') and plotted the total number of **new policies sold** over time.
- The plot shows an **increasing trend** in the sales of new policies, indicating a general growth in performance over time.



Unexpected drop in 2023-12

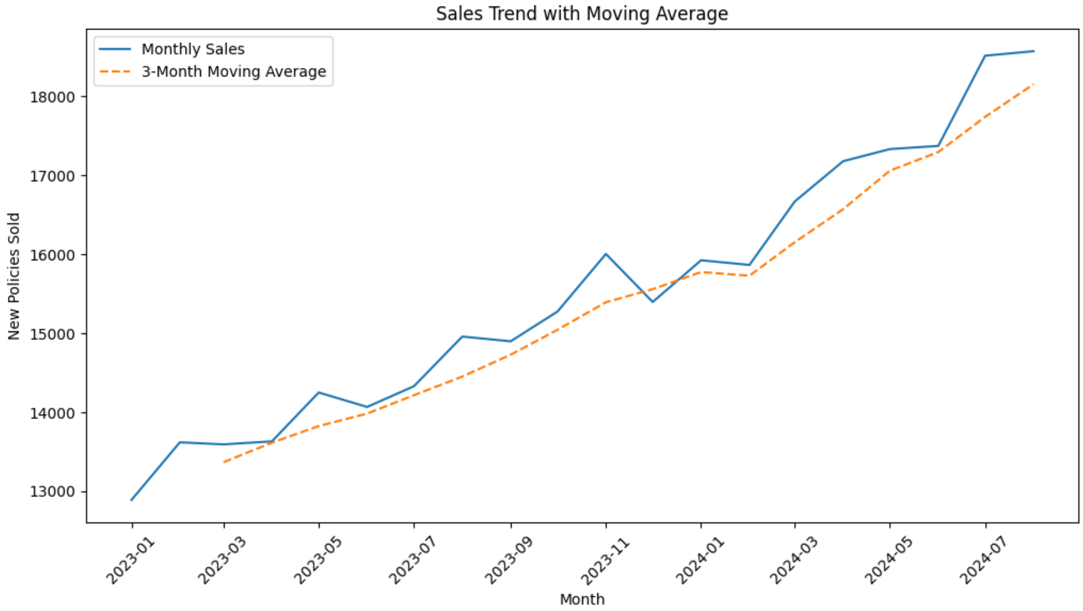
2.2. Identifying Unexpected Drops/Spikes:

- We calculated the **percentage change** in sales between months.
- The **mean and standard deviation** of the percentage change were calculated to define a threshold for significant drops or spikes in sales. This was done by setting an upper and lower threshold as **mean ± 2 standard deviations**.

- We identified months with unexpected **drops or spikes** based on the percentage change and flagged December 2023 as having an unexpected **drop** in sales.

2.3. Adding Moving Average:

- We added a **3-month moving average** to smooth out the sales trend, which helps identify long-term trends while reducing noise from month-to-month fluctuations.
- The moving average line confirmed the **unexpected drop** in December 2023, highlighting it as a significant deviation.



2.4. Insights:

- **Growth in Sales**: There is an overall upward trend in sales, but the **December 2023 drop** may require further investigation. It could be attributed to **external factors** like holidays, market events, or seasonal changes.
- **Unexpected Drop**: Using percentage change and the moving average, we confirmed an unexpected drop in December 2023. This could be due to **seasonality, market conditions**, or **internal changes** (e.g., policy adjustments, agent performance issues).

3. How do all numerical features interact simultaneously? (Multivariate Analysis)

3.1 Multivariate Analysis

We analyzed how different numerical features interact with each other, such as agent_age, unique_proposals_last_7_days, unique_quotations_last_7_days, and others. By examining these features together, we can identify which factors are linked to agent performance. For example, agents making more proposals might also generate more quotations and engage more customers.

3.2 Univariate Analysis

We looked at the distribution of individual features using histograms:

- **Agent Age**: Most agents are between 20-40 years old.
- **Proposals and Quotations**: Most agents make fewer proposals and quotations, but a few agents make many more.
- These features are skewed(**Net Income , ANBP Value, number_of_policy_holders,number_of_cash_payment_policies**), with most agents having lower incomes but some earning significantly more.

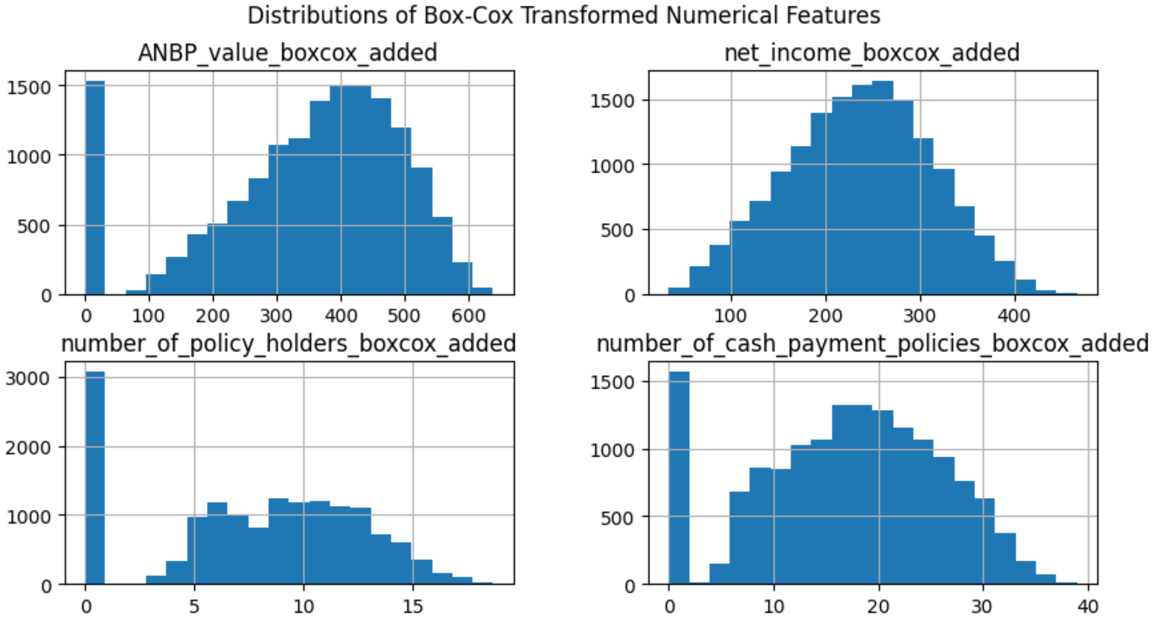
3.3 Insights

- **Skewed Data**: above skewed features, meaning the data might need to be adjusted before building models.
- **Performance Gaps**: There are clear differences between top and low-performing agents. Understanding these differences will help in improving performance.

3.4 Box-Cox Transformation for Numerical Columns

The **Box-Cox transformation** is applied to stabilize variance and make data more normally distributed. This is important because many machine learning models assume normal distribution of the input data. By applying Box-Cox to certain numerical features like ANBP_value, net_income, number_of_policy_holders, and number_of_cash_payment_policies, you've enhanced the performance of these features for predictive modeling.

In the business scenario, this transformation will help improve the accuracy of the model predicting "One NILL" agents (agents who will fail to make any sales in the next month) by addressing skewness in the data.



3.5. Boxplots for Outlier Detection

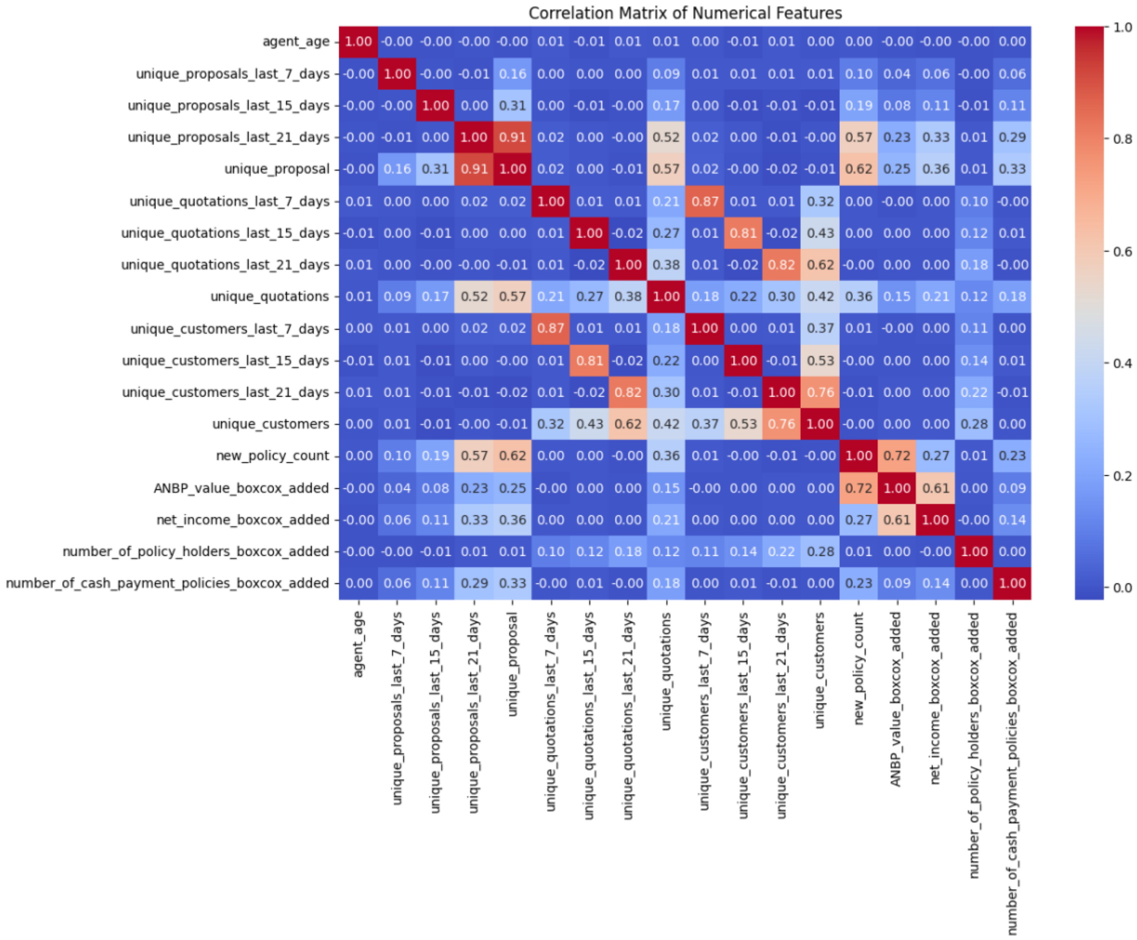
You've used **boxplots** to visualize potential outliers in the numerical data. The boxplots show distributions for each feature like agent_age, unique_proposals_last_7_days, unique_customers_last_7_days, and others. In business terms, identifying and handling outliers is critical because extreme values may mislead predictions and distort the analysis of agent performance.

By capping the outliers instead of removing them, you ensure that the data remains robust for predictive modeling while reducing the influence of extreme values.

3.6. Correlation Matrix

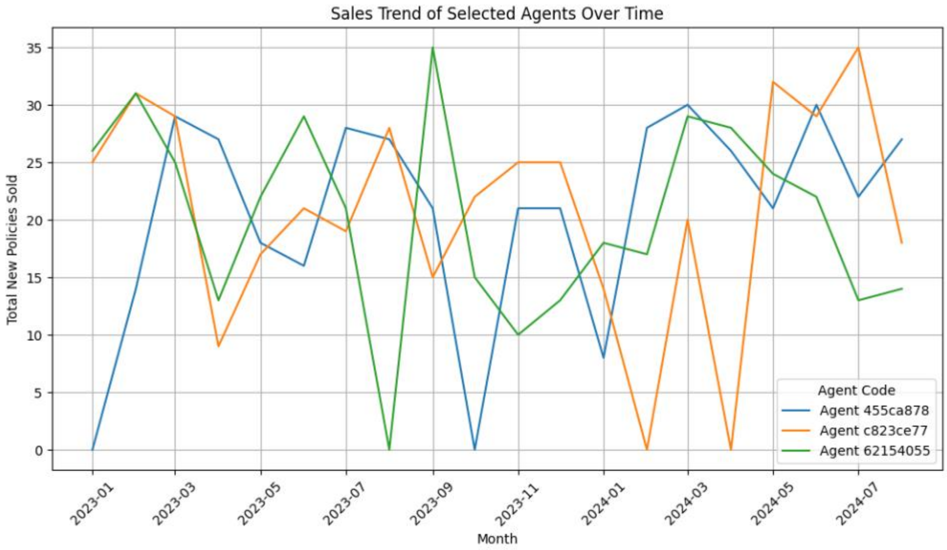
The **correlation matrix** reveals the relationships between different numerical features. In this case, the correlation between features like unique_proposals_last_7_days, unique_quotations_last_7_days, and new_policy_countcan indicate that agents who generate more proposals and quotations tend to sign more policies.

In the business scenario, this insight helps in identifying key predictors of agent success. For example, if a strong correlation exists between unique_proposals_last_7_days and new_policy_count, the business can focus on training agents to improve proposals and quotations to boost sales.



4. How do individual agent trajectories evolve over time?

1. **Extracted Agent IDs:**
   - We retrieved a list of unique agent IDs from the dataset to analyze specific agent performances. This allows you to focus on individual agents or a subset of agents for more detailed analysis.
2. **Selected Agents for Plotting:**
   - We selected specific agents (e.g., the first 3 or 10 agents) to track their sales performance over time, which helps in comparing their progress and identifying patterns or trends.
3. **Aggregated Sales Data by Month:**
   - For each selected agent, you aggregated their new_policy_count (number of new policies sold) by month. This aggregation provides a clearer view of each agent's monthly performance, smoothing out any fluctuations in daily or weekly data.
4. **Plotted Sales Trends:**
   - We plotted the total new policies sold over time (monthly) for each agent. This line graph shows how their sales have evolved across different months, providing a visual representation of each agent's performance.



5. **Customized Plots:**
   - We customized the plot to include titles, labels, and legends to make the graph easy to interpret. This is helpful for clear communication of results when sharing with others, especially stakeholders or decision-makers.

Insights:

1. **Identifying Sales Trends:**
   - The line graphs reveal the **sales trajectory** for each agent. For example, if an agent's sales consistently increase over time, it indicates strong performance, while a decline may signal potential issues or lack of motivation.
2. **Comparing Agents:**
   - By comparing multiple agents, we can spot high performers versus low performers. For instance, agents whose sales show upward trends may need fewer interventions, while those with erratic or declining sales could benefit from additional training or support.
3. **Detecting Seasonality or Cycles:**
   - If certain months show higher sales for all or most agents, it could point to **seasonality**, where insurance sales peak at certain times of the year. Identifying such patterns allows the company to strategize on resource allocation and agent support during peak periods.
4. **Actionable Insights for Business:**
   - The business can use this data to:
     - **Identify struggling agents**: Agents with poor performance can be flagged early for personalized coaching or mentoring.
     - **Develop targeted training**: Tailor training programs for agents who show inconsistent or declining performance.
     - **Optimize incentives**: Reward top performers to maintain motivation or provide additional incentives for underperforming agents to boost their results.

5. **Performance Evaluation:**
    - By tracking the sales trend of agents over time, you can assess the **effectiveness of interventions** or changes in strategy (e.g., training, new policies, etc.). If interventions lead to better performance, it will show as an upward trend in the sales graph.

5. Innovative EDA (Explore more to uncover hidden insights).

5.1 Principal Component Analysis (PCA) for Feature Reduction

- **PCA** was applied to the quotations and customers features to reduce the dimensionality of the data. This technique helps in identifying the most significant features while removing noise.
- The **PCA Transformation** captures variance from the data and provides new components (pca_quotations_1, pca_quotations_2, pca_customers_1, pca_customers_2) that summarize the most important patterns.
- **Variance Explained**: For both the quotations and customers components, a large portion of the variance (84.88% for quotations and 92% for customers) is captured by two components, which means PCA has effectively reduced the data while preserving critical information.

5.2 Scaling and Normalization of PCA Components

- After applying PCA, we scaled the new components using the **Min-Max Scaler** to ensure that values are positive and within the same range. This scaling is important, especially when using techniques like K-Nearest Neighbors (KNN), which rely on distance metrics and are sensitive to the scale of the features.

5.3 Agent Seniority Analysis:

- We calculated the **agent seniority** in months by subtracting the agent's joining month from the current month. This information helps understand the experience level of agents.
- We plotted a **scatter plot** to visualize the relationship between agent seniority and new policy sales. However, the plot indicates that there isn't a clear dependency between the two.
- A **boxplot** was used to show the distribution of new policies based on agent seniority, which also indicates that performance does not significantly depend on seniority.

5.4 Statistical Analysis (ANOVA):

- We performed an **ANOVA (Analysis of Variance)** to test if there’s a significant difference in new policy sales across different agent seniority groups. The **p-value (0.758)** suggests that seniority does not significantly affect policy sales, which aligns with the scatter and boxplot results.

Insights

- **PCA** has effectively reduced the data dimensions without losing critical variance, allowing the model to focus on the most important components.
- **Scaling** the PCA components ensures consistency, particularly for algorithms like KNN.
- **Agent Seniority**: The lack of a clear relationship between seniority and performance suggests that other factors may influence agent success more significantly.
- The **ANOVA results** confirm that **seniority** does not impact sales, which means that interventions should focus on factors other than experience, such as training or motivation.

5.5 Time to First Sale Calculation

- We calculated the **time to first sale** for agents, which measures the number of months it takes for an agent to make their first sale after joining. This gives us insight into how quickly agents are able to start making sales after they begin their role.

5.6 Visualizing Time to First Sale

- We visualized the **distribution** of the time it takes agents to make their first sale using a **histogram** and a **boxplot**.
  - The **histogram** shows that most agents make their first sale within 20 months, with some taking much longer.
  - The **boxplot** helps identify any **outliers**, showing that while most agents take a moderate amount of time, some take significantly longer.

5.7 Agent Performance Analysis

- We analyzed agent performance by grouping data by **agent code** and **year-month** to calculate the sum of **new policies sold** each month.
- We visualized the monthly performance of a specific agent using a **line plot** to track how their policy sales change over time. This helps in identifying trends and periods of high or low performance.

5.8 Seniority Analysis

- We explored the relationship between **agent seniority** (the time agents have been with the company) and **policy sales**. The **scatter plot** and **boxplot** showed that there isn't a clear correlation between seniority and new policy sales.
  - This suggests that factors other than experience might be driving agent performance.

5.9 Statistical Testing (ANOVA)

- We performed **ANOVA** to statistically test whether there's a significant difference in **new policy sales** across different levels of **agent seniority**. The **p-value**

(0.758) indicated that seniority does not significantly impact sales performance, supporting the idea that factors other than experience are more influential.

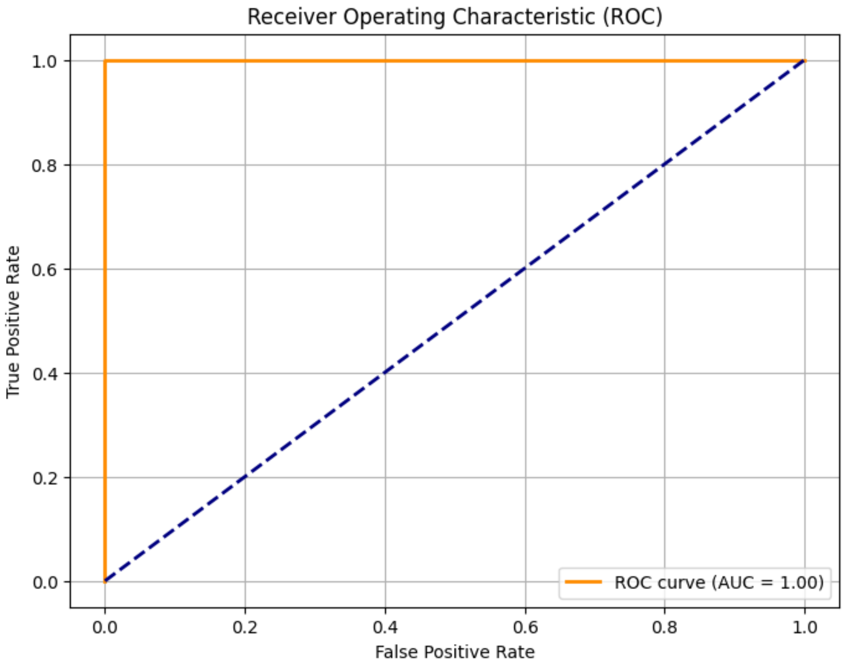
Insights

- **Time to First Sale**: Most agents take less than 20 months to make their first sale, but a few take much longer, which could indicate a need for intervention or more targeted training for new agents.
- **Agent Performance**: Tracking performance over time can help identify periods of high or low sales, and adjusting strategies accordingly can help improve outcomes.
- **Seniority**: Seniority alone doesn't appear to significantly impact agent performance, meaning that other factors like training, motivation, or resources might play a more important role.

Part 1 - Predict NILL Agents

1. **Data Inspection:**
    - We loaded two datasets: `train_df` and `test_df`, each containing numerical columns.
    - We checked for missing values, duplicates, and unique values in the columns to ensure the quality and integrity of the data.
    - The `new_policy_count` column was identified as irrelevant and dropped from both datasets.
2. **Data Preprocessing:**
    - We performed some preprocessing steps such as converting certain columns to integers and scaling the data.
    - We used `RandomUnderSampler` to balance the class distribution in the target variable.
    - We scaled the feature data using `MinMaxScaler` to normalize the values between 0 and 1.
3. **Model Training:**
    - We used the XGBoost classifier for the model and performed hyperparameter tuning using `GridSearchCV`to find the optimal model parameters, including learning rate, max depth, and number of estimators.
    - After tuning, the model was trained using the best parameters found from the grid search.
4. **Model Evaluation:**
    - We evaluated the model using various metrics, including accuracy, confusion matrix, classification report, and ROC curve.
    - The model performed exceptionally well, achieving a perfect classification result on both training and test data, with an AUC of 1.00, indicating the model's strong ability to differentiate between classes.
5. **Validation:**
    - We also predicted the values for the validation dataset (`Val`) and evaluated the performance using similar metrics as above, with equally strong results.

   - We plotted the ROC curve and achieved an AUC of 1.00, confirming the robustness of the model.



6. **Final Output:**
   - We saved the predictions for the test data into a CSV file named `submission.csv`.

Business Insights:

- **High Model Accuracy:** The model's high accuracy (1.00 AUC) suggests it can effectively predict whether customers will buy a new policy based on the provided features, making it highly reliable for decision-making processes in the business.
- **Customer Targeting:** By using this model, the business can focus marketing efforts on customers who are predicted to purchase a new policy, ensuring more efficient allocation of resources.
- **Data-driven Decisions:** The strong performance of the model highlights the potential for data-driven decision-making, where insights from customer behavior can improve customer acquisition strategies and retention programs.
- **Operational Efficiency:** Automating predictions with this model could streamline operations, reducing human intervention in decision-making and ensuring more consistent and accurate assessments of customer behavior.

Part 2 - Monitor and Improve Existing Agent Performance

1. Data Preparation and Feature Selection

- We first selected relevant performance metrics for each agent such as `new_policy_count`, `ANBP_value`, `net_income`, `number_of_policy_holders`, and `number_of_cash_payment_policies`.
- The dataset was then **scaled** using the **StandardScaler** to standardize the features. This is necessary because K-means clustering relies on distances between data points, and scaling ensures all features contribute equally to the clustering process.

2. K-means Clustering

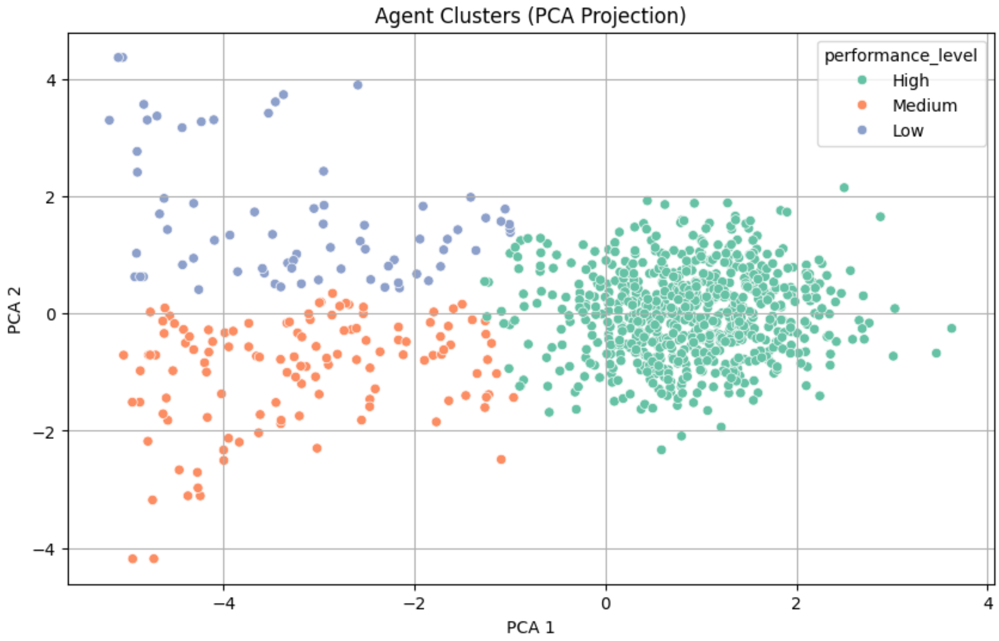
- We applied **K-means clustering** to categorize agents into **3 performance clusters** (high, medium, and low) based on their performance metrics. The **Silhouette score (0.54)**, which measures how well the clustering works, suggests that the model performs reasonably well.
- After running the clustering algorithm, we grouped the agents into clusters and examined the average values of the selected features in each cluster. The **cluster means** show that:
  - **High-performing agents** have a higher number of policies sold, net income, and policyholders.
  - **Medium-performing agents** have moderate values for these metrics.
  - **Low-performing agents** have the lowest values for policies sold and other performance metrics.

3. Categorizing Agents

- Based on the **cluster means**, we categorized agents into **High, Medium, and Low** performance groups.
  - **High**: These agents are performing well and likely require retention strategies or further development into leadership roles.
  - **Medium**: These agents need improvement and could benefit from training or new strategies to boost their performance.
  - **Low**: These agents need more targeted interventions, such as mentorship or more structured training.

4. PCA Visualization

- We applied **PCA** to reduce the data dimensions and plotted the agent clusters in a 2D space. The scatter plot shows that the **high** and **low** performing agents are well-separated, which suggests that the clustering was successful.



5. Recommendations

- Based on the performance groupings, we **assigned custom recommendations**:
    - **High performers**: Offer bonuses or leadership tracks to retain and motivate them.
    - **Medium performers**: Provide skill development programs to enhance their sales.
    - **Low performers**: Assign mentors and structured training to improve their performance.

Business Implications

- **K-means clustering** helped us identify groups of agents with different performance levels. By clustering agents, we can develop targeted interventions and allocate resources more efficiently.
- **PCA** provided a clear visualization of agent performance clusters, making it easier to interpret and communicate the results.