PG Certificate Program by EICT IIT Roorkee

CAPSTONE PROJECT REPORT

LIFE INSURANCE SALE BONUS PREDICTION

By-

Pankaj Tambe

April 2024

Table of Contents

Objective

This project is designed to optimize the performance management system of a leading life insurance company by predicting bonuses for its agents. This prediction aims to enhance both the recognition of high-performing agents and the developmental strategies for those who are underperforming. By analysing extensive sales data, the project seeks to identify performance trends and patterns that are critical for strategic decision-making.

The primary goals of this study are:

Performance Insight: To develop a deeper understanding of agent performance across various demographics and operational metrics. This insight will help tailor specific programs that boost productivity and ensure agents are effectively motivated.

Agent Segmentation: To segment agents based on their sales performance and other relevant criteria, thereby enabling targeted developmental and reward programs.

Predictive Analysis: To employ predictive analytics for forecasting agent bonuses, which will help in budgeting and financial planning while also aligning agent incentives with company goals.

Strategic Training and Rewards: To design and implement targeted engagement activities for high-performing agents and create upskilling programs for those not meeting performance benchmarks.

The expected outcome of this project is a robust analytical model that not only forecasts agent bonuses accurately but also provides actionable insights into how performance metrics relate to sales outcomes. This model will serve as a critical tool for the HR and sales departments to drive better sales practices, enhance agent satisfaction, and ultimately, contribute positively to the company's profitability and market competitiveness.

INTRODUCTION:

- 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.

Need for this Study/Project

- With the help of this problem, we want to know that how the insurance company agents are performing.
- With the predictions it's better for the company to understand that where they really needs to focus on more as for the agents selling less policies the company needs some booster training performs. As the policies are as good as the agents portray it to be the potential customer.
- While the agents those are performing good i.e those who are selling more policies there should be a way to reward them, to make their contribution known, so that they perform the same or even better in the future.

Understanding business/social opportunity

- A company is as good as their employers.
- Speaking for a Life Insurance Company, their agents are the best way to make the companies policies, aims, and perks known to the customers. Once the customer gets interested in the policy , it is easier to convince the customer hence improving the sales and also motivating the agent at the same time.
- With this, the market share of the company will receive more ground dominating the potential opponents.
- Moreover, the agents can be classified into categories, which helps the company to get better insight that where they really need to put more effort.
- The customer feedback can be helpful for the company to develop, improved and updated policies/products, meeting customer needs.
- Hereby, the easiest way to retain their agents.

- Overall, multiplying and adding to company's profit.

Approach to Solve the Problem

To address the challenges outlined in the objective, we employed a systematic approach combining data exploration, preprocessing, and advanced predictive modeling. The following steps detail our methodology:

1. **Data Acquisition and Understanding:**

- **Data Collection**: The dataset, consisting of sales records and agent performance metrics, was sourced from the company's internal database.
- **Preliminary Analysis**: Initial data exploration was conducted to understand the variables, identify data types, and detect any obvious data quality issues.

2. **Data Preprocessing:**

- **Cleaning**: Irrelevant and redundant data, such as the 'CustID' column, was removed to streamline the analysis. Missing values were imputed using appropriate statistical methods, ensuring the integrity of the dataset.
- **Transformation**: Continuous variables were normalized, and categorical variables were encoded using one-hot encoding to prepare them for modeling. This step was crucial to address non-numerical data and scale differences among features.

3. **Exploratory Data Analysis (EDA):**

- **Univariate Analysis**: Each variable was analyzed individually to summarize its main characteristics and distribution.
- **Bivariate and Multivariate Analysis**: Relationships between variables were explored using statistical correlations and visualizations to understand the interactions between features and their impact on agent bonuses.

4. **Feature Engineering:**

- **Selection and Construction**: New features were crafted based on domain knowledge to enhance the model's predictive capability. For instance, a 'Premium' feature might be calculated from existing data points to reflect potential revenue from an agent's sales.
- **Reduction**: Techniques such as Principal Component Analysis (PCA) were considered to reduce dimensionality, focusing on the most informative features.

5. **Predictive Modeling:**

- **Model Selection**: Regression models were evaluated to predict agent bonuses based on performance indicators. We started with simple linear regression to establish a baseline and progressively tested more complex models like Random Forest and Neural Networks to compare performance.
- **Training and Validation**: The dataset was split into training and testing sets to ensure the model was trained and validated on different data samples. Cross-validation techniques were employed to generalize the findings and avoid overfitting.

6. **Model Optimization:**

- **Parameter Tuning**: Hyperparameters were optimized using techniques like Grid Search to find the best model settings.
- **Performance Evaluation**: Models were assessed based on metrics such as RMSE (Root Mean Squared Error) and R-squared to ensure accuracy and explainability.

7. **Deployment and Monitoring:**

- **Implementation**: The final model was deployed within the company's operational framework to start predicting bonuses and influencing agent management strategies.
- **Feedback Loop**: Continuous monitoring was established to track the model's performance over time, adjusting strategies as needed based on feedback and evolving business conditions.

By adhering to this structured approach, we aim to deliver a solution that not only meets the initial objectives but also adapts to future changes and improvements in data quality and modeling techniques.

Code Implementation and Results

EDA - Uni-variate / Bi-variate / Multi-variate analysis to understand relationship b/w variables. - Both visual and non-visual understanding of the data.

Let's perform Exploratory Data Analysis (EDA) on the dataset.

Head of the Data:

```python
df1=df.drop('CustID',axis=1)
df1.head()
```

Python

	AgentBonus	Age	CustTenure	Channel	Occupation	EducationField	Gender	ExistingProdType	Designation	NumberOfPolicy	MaritalStatus	MonthlyIncome	Complaint	ExistingPolicyTenure	Su
0	4409	22.0	4.0	Agent	Salaried	Graduate	Female	3	Manager	2.0	Single	20993.0	1	2.0	
1	2214	11.0	2.0	Third Party Partner	Salaried	Graduate	Male	4	Manager	4.0	Divorced	20130.0	0	3.0	
2	4273	26.0	4.0	Agent	Free Lancer	Post Graduate	Male	4	Exe	3.0	Unmarried	17090.0	1	2.0	
3	1791	11.0	NaN	Third Party Partner	Salaried	Graduate	Fe male	3	Executive	3.0	Divorced	17909.0	1	2.0	
4	2955	6.0	NaN	Agent	Small Business	UG	Male	3	Executive	4.0	Divorced	18468.0	0	4.0	

Fig No. 1(Data Head)

- We dropped "CustID" as it is not that useful for agent bonus which is our target variable.
- Above figure gives us the idea of how the dataset looks like. However, complete list of variables is not visible.

Shape of the Dataset:

There are total 4520 rows and 19 columns (after removing CustID) in the dataset.

Descriptive Analysis of the columns:

```
df1.describe(include='all').T
```

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
AgentBonus	4520.0	NaN	NaN	NaN	4077.838274	1403.321711	1605.0	3027.75	3911.5	4867.25	9608.0
Age	4251.0	NaN	NaN	NaN	14.494707	9.037629	2.0	7.0	13.0	20.0	58.0
CustTenure	4294.0	NaN	NaN	NaN	14.469027	8.963671	2.0	7.0	13.0	20.0	57.0
Channel	4520	3	Agent	3194	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Occupation	4520	5	Salaried	2192	NaN	NaN	NaN	NaN	NaN	NaN	NaN
EducationField	4520	7	Graduate	1870	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	4520	3	Male	2688	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ExistingProdType	4520.0	NaN	NaN	NaN	3.688938	1.015769	1.0	3.0	4.0	4.0	6.0
Designation	4520	6	Manager	1620	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NumberOfPolicy	4475.0	NaN	NaN	NaN	3.565363	1.455926	1.0	2.0	4.0	5.0	6.0
MaritalStatus	4520	4	Married	2268	NaN	NaN	NaN	NaN	NaN	NaN	NaN
MonthlyIncome	4284.0	NaN	NaN	NaN	22890.309991	4885.600757	16009.0	19683.5	21606.0	24725.0	38456.0
Complaint	4520.0	NaN	NaN	NaN	0.287168	0.452491	0.0	0.0	0.0	1.0	1.0
ExistingPolicyTenure	4336.0	NaN	NaN	NaN	4.130074	3.346386	1.0	2.0	3.0	6.0	25.0
SumAssured	4366.0	NaN	NaN	NaN	619999.699267	246234.82214	168536.0	439443.25	578976.5	758236.0	1838496.0
Zone	4520	4	West	2566	NaN	NaN	NaN	NaN	NaN	NaN	NaN
PaymentMethod	4520	4	Half Yearly	2656	NaN	NaN	NaN	NaN	NaN	NaN	NaN
LastMonthCalls	4520.0	NaN	NaN	NaN	4.626991	3.620132	0.0	2.0	3.0	8.0	18.0
CustCareScore	4468.0	NaN	NaN	NaN	3.067592	1.382968	1.0	2.0	3.0	4.0	5.0

Fig No. 2(Data description)

- The above table has all the description for all the variables along with categorical variables.
- The description involves variable count, unique values, top frequently occurring categories like Agent 3194, mean, standard deviation, minimum, 25%, 50% (which is median) , 75% and maximum values are present in the respective variables.
- We may change it by encoding the data in the future if needed.
- We can see the missing values present in the dataset.
- The unique is only present for the categorical variables which hold a specific category.

- Example: Gender has male and female hence it should hold unique value of 2 but later we see some subcategories needs to be renamed.

Info of the data:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4520 entries, 0 to 4519
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   AgentBonus            4520 non-null   int64
 1   Age                   4251 non-null   float64
 2   CustTenure            4294 non-null   float64
 3   Channel               4520 non-null   object
 4   Occupation            4520 non-null   object
 5   EducationField        4520 non-null   object
 6   Gender                4520 non-null   object
 7   ExistingProdType      4520 non-null   int64
 8   Designation           4520 non-null   object
 9   NumberOfPolicy        4475 non-null   float64
 10  MaritalStatus         4520 non-null   object
 11  MonthlyIncome         4284 non-null   float64
 12  Complaint             4520 non-null   int64
 13  ExistingPolicyTenure  4336 non-null   float64
 14  SumAssured            4366 non-null   float64
 15  Zone                  4520 non-null   object
 16  PaymentMethod         4520 non-null   object
 17  LastMonthCalls        4520 non-null   int64
 18  CustCareScore         4468 non-null   float64
dtypes: float64(7), int64(4), object(8)
memory usage: 671.1+ KB
```

Fig No. 3(Info of data)

- We have 7 ‘float’ data type.
- We have 4 ‘integer’ data type.
- We have 8 ‘ object’ data type.

- Age is shown as float, however we will later find out that if it needs to be changed into integer or not , it won't make any difference in our observations as such.
- We can clearly notice the missing values present in the dataset.

Count of missing values is given below.

```
df1.isnull().sum()
```

```
AgentBonus                0
Age                     269
CustTenure              226
Channel                   0
Occupation                0
EducationField            0
Gender                    0
ExistingProdType          0
Designation               0
NumberOfPolicy           45
MaritalStatus             0
MonthlyIncome           236
Complaint                 0
ExistingPolicyTenure    184
SumAssured              154
Zone                      0
PaymentMethod             0
LastMonthCalls            0
CustCareScore            52
dtype: int64
```

Fig No. 4(Checking missing values)

- There are no duplicate rows found.
- The missing values may affect the predictions. So we need to treat them. Hence the missing values are imputed using **median values** in the respective columns.

Checking for Unique Categorical values.

CHANNEL has 3 Unique values:

```
print(df1['Channel'].value_counts())
```

```
Channel
Agent                  3194
Third Party Partner     858
Online                  468
Name: count, dtype: int64
```

OCCUPATION has 5 Unique values:

```
print(df1['Occupation'].value_counts())
```

```
Occupation
Salaried          2192
Small Business    1918
Large Business     255
Laarge Business    153
Free Lancer          2
Name: count, dtype: int64
```

EDUCATIONFIELD has 7 Unique values:

```
print(df1['EducationField'].value_counts())
```

```
EducationField
Graduate          1870
Under Graduate    1190
Diploma            496
Engineer           408
Post Graduate      252
UG                 230
MBA                 74
Name: count, dtype: int64
```

GENDER has 3 Unique values:

```
print(df1['Gender'].value_counts())
```

```
Gender
Male       2688
Female     1507
Fe male     325
Name: count, dtype: int64
```

DESIGNATION has 6 Unique values:

```
print(df1['Designation'].value_counts())
```

```
Designation
Manager           1620
Executive         1535
Senior Manager     676
AVP                336
VP                 226
Exe                127
Name: count, dtype: int64
```

MARITALSTATUS has 4 Unique values:

```
print(df1['MaritalStatus'].value_counts())
```

```
MaritalStatus
Married      2268
Single       1254
Divorced      804
Unmarried     194
Name: count, dtype: int64
```

ZONE has 4 Unique values:

```
print(df1['Zone'].value_counts())
```

```
Zone
West     2566
North    1884
East       64
South       6
Name: count, dtype: int64
```

PAYMENTMETHOD has 4 Unique values:

```
print(df1['PaymentMethod'].value_counts())
```

```
PaymentMethod
Half Yearly    2656
Yearly         1434
Monthly         354
Quarterly        76
Name: count, dtype: int64
```

- Here we can observe that subcategories highlighted with a different color shows an error in the naming convention hence have to be renamed.
- Example: 'Laarge' and 'Large' Business can be put in the same category, the same for 'UG' and 'Under Graduate', 'Graduate' and 'Post Graduation' , 'Fe male' and 'Female' , and 'Exe' and 'Executive'.

Categorical Variable UNIVARIATE Analysis:

EducationField:

```
sns.countplot(data=df1,x="EducationField")
```

```
<Axes: xlabel='EducationField', ylabel='count'>
```

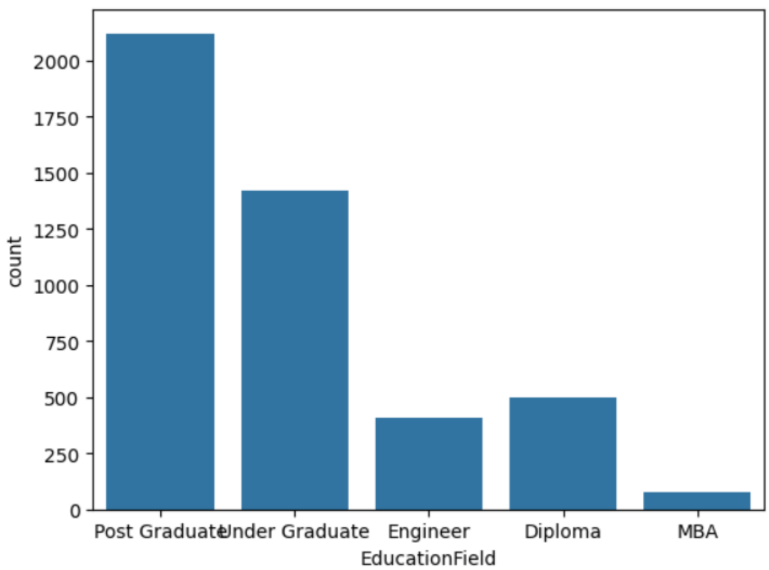

Fig No.5 (Univariate Analysis of 'EducationField')

Post graduation is the most approached Customers, followed by Under graduate customers. MBA being the least of all.

Channel:

```
sns.countplot(data=df1,x="Channel")
```

```
<Axes: xlabel='Channel', ylabel='count'>
```

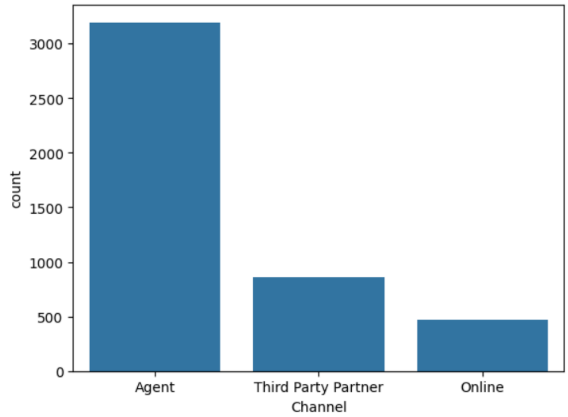

Fig No. 6(Univariate Analysis of ‘Channel)

Investment of a customer is mostly done through an Agent. Least purchase is done online.

Occupation:

```
sns.countplot(data=df1,x="Occupation")
```

```
<Axes: xlabel='Occupation', ylabel='count'>
```

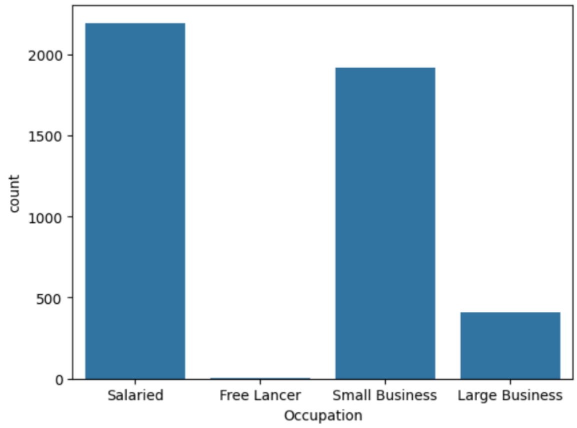

Fig No.7 (Univariate Analysis of Occupation) Around

48% of the Customers are Salaried.

Apparently, freelancers have very less weightage, which is almost negligible.

Gender:

```
sns.countplot(data=df1,x="Gender")
```

```
<Axes: xlabel='Gender', ylabel='count'>
```

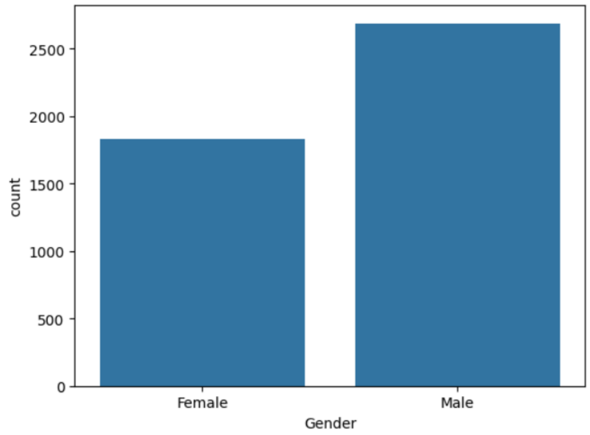

Fig No. 8(Univariate Analysis of ‘Gender’)

Male customers are more as compared to Female customers.

Designation:

```
sns.countplot(data=df1,x="Designation")
```

```
<Axes: xlabel='Designation', ylabel='count'>
```

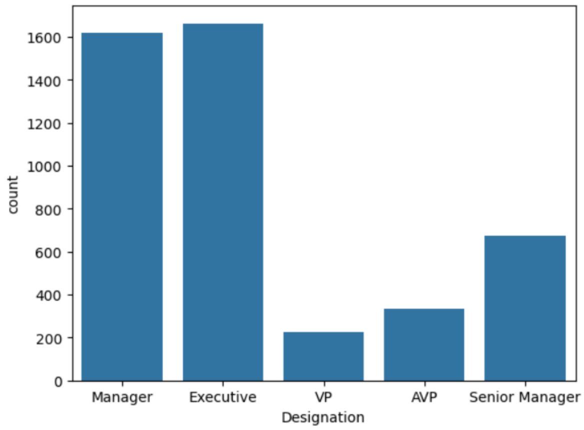

Fig No.9(Univariate Analysis of ‘Designation’)

Most of the Customers are Managers and Executives, followed by Senior Manager, AVP and VP respectively.

Marital Status:

```
sns.countplot(data=df1,x="MaritalStatus")
```

```
<Axes: xlabel='MaritalStatus', ylabel='count'>
```

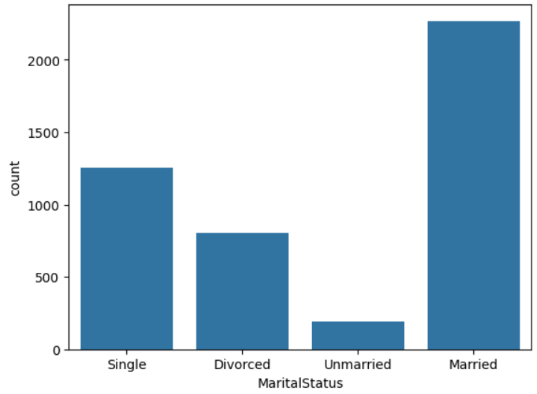

Fig No. 10(Univariate Analysis of ‘Marital Status)

Married Customers are the highest selling customers, least being Unmarried one.

Zone:

```
sns.countplot(data=df1,x="Zone")
```

```
<Axes: xlabel='Zone', ylabel='count'>
```

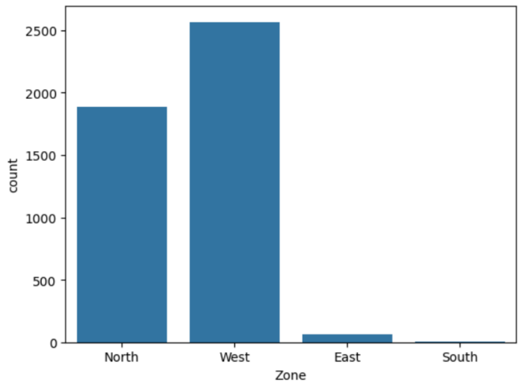

Fig No.11(Univariate Analysis of ‘Zone’)

Most Customers are bought by the West and North zone compared to East and South Zone.

PaymentMonth:

```
sns.countplot(data=df1,x="PaymentMethod")
```

```
<Axes: xlabel='PaymentMethod', ylabel='count'>
```

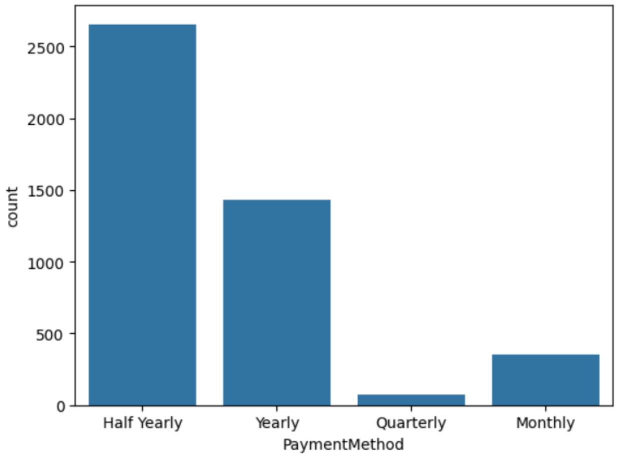

Fig No. 12(Univariate Analysis of 'PaymentMonth')

Most of the Customers have opted for the Half yearly Payment plan.

Categorical Variables Bivariate Analysis w.r.t Agent Bonus Channel:

```
sns.boxplot(x = df1['Channel'],
            y = df1['AgentBonus']
            )
```

```
<Axes: xlabel='Channel', ylabel='AgentBonus'>
```

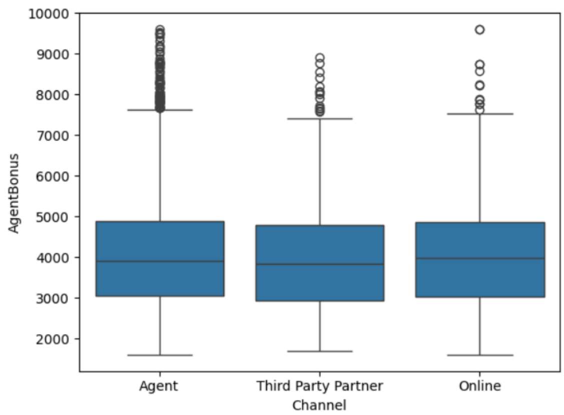

Fig No. 13(Bivariate Analysis of ‘Channel’)

As we can see Agent Bonus has a lot of outliers present for every channel with almost similar mean values for all 3 channels.

Occupation:

```
sns.boxplot(x = df1['Occupation'],
            y = df1['AgentBonus']
            )
```

```
<Axes: xlabel='Occupation', ylabel='AgentBonus'>
```

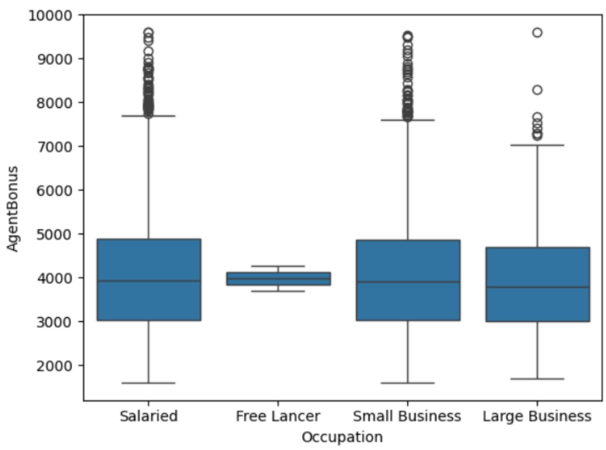

Fig No. 14(Bivariate Analysis of’Occupation’)

It is visible that almost similar mean values are there for all occupations.

No outlier present for Free Lancer could be because we have only 2 data points for Free Lancer.

Gender:

```
sns.boxplot(x = df1['Gender'],
            y = df1['AgentBonus']
           )
```

<Axes: xlabel='Gender', ylabel='AgentBonus'>

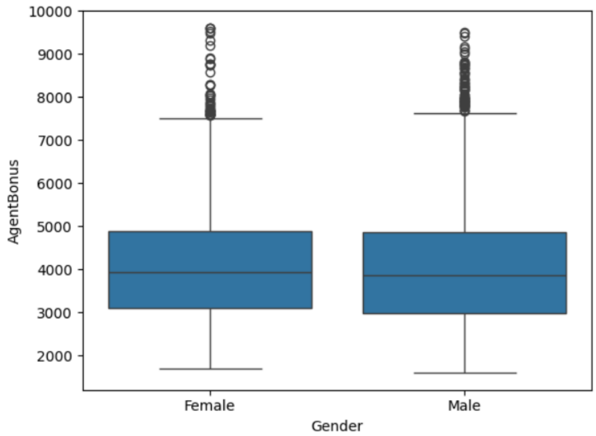

Fig No. 15(Bivariate Analysis of’Gender’)

Agent Bonus contains lots of outlier values for both the genders with almost similar mean values for both male and Female.

Designation:

```
sns.boxplot(x = df1['Designation'],
            y = df1['AgentBonus']
           )
```

<Axes: xlabel='Designation', ylabel='AgentBonus'>

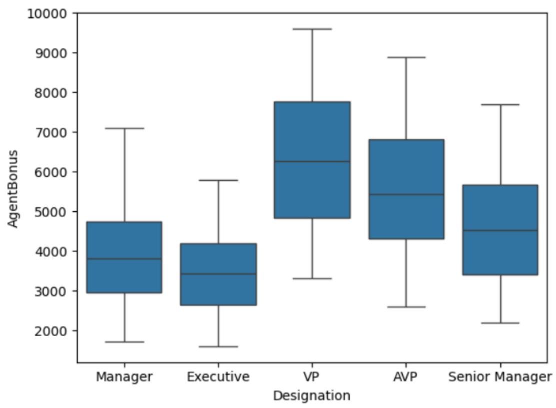

Fig No. 16 (Bivariate Analysis of ’Designation’)

There are no outliers present. VP Designation has the highest mean as compared to other designations.

Marital Status:

```
sns.boxplot(x = df1['MaritalStatus'],
            y = df1['AgentBonus']
            )
```

`<Axes: xlabel='MaritalStatus', ylabel='AgentBonus'>`

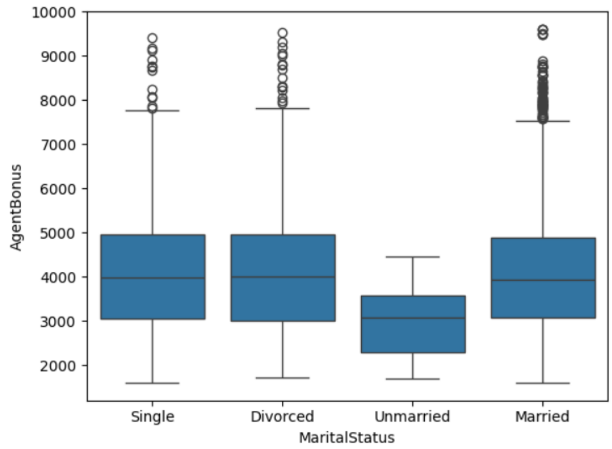

Fig No. 17 (Bivariate Analysis of ’Marital Status’)

Agent Bonus variable has lot of outlier values for all the marital status except for the unmarried customers. With almost similar mean values for all the three customers except unmarried.

Zone:

```
sns.boxplot(x = df1['Zone'],
            y = df1['AgentBonus']
            )
```

`<Axes: xlabel='Zone', ylabel='AgentBonus'>`

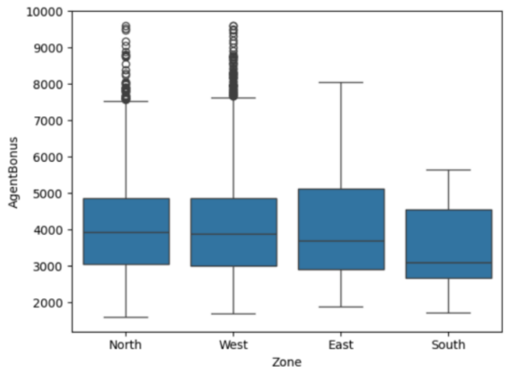

Fig No.18 (Bivariate Analysis of’Zone’)

The outliers are present only in North and West Zones. Both having almost similar means.

There are no outliers present in the East and South Zones may be because of less customer traffic from those Zones.

PaymentMethod:

```
sns.boxplot(x = df1['PaymentMethod'],
            y = df1['AgentBonus']
           )
```

<Axes: xlabel='PaymentMethod', ylabel='AgentBonus'>

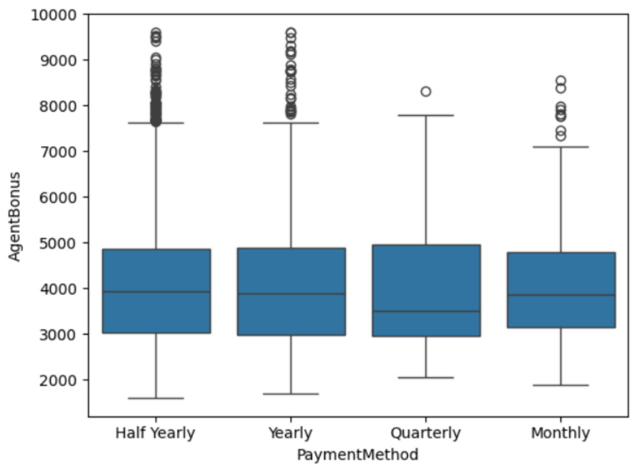

Fig No.19 (Bivariate Analysis of ‘PaymentMethod’)

There are outliers present for all the Payment methods where Quarterly paying customers has the lowest mean.

Let us have a look at the Pairplot:

A Pairplot is used to plot the relationship between the Numeric Variables in the dataset.

```
sns.pairplot(df2, hue='AgentBonus',corner=True)
plt.show()
```

✓ 53.0s

Fig No. 20 (Pairplot)

Heatmap:

Fig No. 21 (Heatmap)

Data Cleaning and Pre-processing - Approach used for identifying and treating missing values and outlier treatment (and why) - Need for variable transformation (if any) - Variables removed or added and why (if any)

Business Insights from EDA:

Removal of Outliers doesn’t seems to be the correct approach as some variables like ‘SumAssured’ are allowed to have some outliers however the model will get affected if outliers are not treated. As we are planning to do Linear Regression for our model, the outliers will produce a biased result with

Linear Regression and to prevent that from happening we will go with the outlier removement method.

We may add the new variables like Premium which will come up as these another variable that has direct correlation with AgentBonus and will make it easier to observe the high performing and the low performing agents as the ones who bring in more premium and good for the firm and performing well and those incurring low premium needs to be focused on.

However, adding new variables are not as simple as it sounds as here we have 4520 rows that needs to have a value which will add to the prediction and if we are not careful enough, the new variable introduced will add more variance to our predictions and can be biased too, which ultimately can affect the model, hence it is not recommended unless you have extreme and thorough domain knowledge.

So here, we have completed the EDA . In the coming exercises we will build the model as this is a Classification problem, Regression Techniques for model building will be our go-to approach.

The data from the EDA can be said to be highly unbalanced e.g Zone, South has less weightage similar for Occupation-Freelancer, more data is needed or upscale the data, similar can be the case with EducationField- MBA where we need to have enough data to not make biased decisions which can be done by upscaling the data which will add another problem where the data would

Model building - Clear on why was a particular model(s) chosen. - Effort to improve model performance.

- Regression needs numerical values.
- But in the dataset we have lots of categorical variables.
- And because most of most of the categorical variables have categories more than 2, we need to apply one-hot encoding.
- One-Hot encoding takes every level of the category and turns it into a variable with two level (yes/no).

The data after one-hot encoding looks like this.

```
df1
```

	AgentBonus	Age	CustTenure	ExistingProdType	NumberOfPolicy	MonthlyIncome	Complaint	ExistingPolicyTenure	SumAssured	LastMonthCalls	CustCareScore
0	4409.0	22.0	4.0	3.0	2.0	20993.0	1	2.0	806761.0	5.0	2.0
1	2214.0	11.0	2.0	4.0	4.0	20130.0	0	3.0	294502.0	7.0	3.0
2	4273.0	26.0	4.0	4.0	3.0	17090.0	1	2.0	578976.5	0.0	3.0
3	1791.0	11.0	13.0	3.0	3.0	17909.0	1	2.0	268635.0	0.0	5.0
4	2955.0	6.0	13.0	3.0	4.0	18468.0	0	4.0	366405.0	2.0	5.0
...	...	...	...	...	...	...	...	...	...	...	...
4515	3953.0	4.0	8.0	4.0	2.0	26355.0	0	2.0	636473.0	9.0	1.0
4516	2939.0	9.0	9.0	2.0	2.0	20991.0	0	3.0	296813.0	1.0	3.0
4517	3792.0	23.0	23.0	5.0	5.0	21606.0	0	2.0	667371.0	4.0	1.0
4518	4816.0	10.0	10.0	4.0	2.0	20068.0	0	6.0	943999.0	1.0	5.0
4519	4764.0	14.0	10.0	5.0	2.0	23820.0	0	3.0	700308.0	1.0	3.0

4520 rows × 11 columns

Fig No. 22 (Head after encoding)

- Building our Linear Regression Model with the unprocessed data above.
- Also, this data has no outliers as they were removed in EDA.(part-1)

Split X and y into training and test set in 75:25 ratio

```
X = df_final.drop("AgentBonus",axis=1)  ## Features
y = df_final["AgentBonus"]  ## Target
# Split X and y into training and test set in 75:25 ratio
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25 , random_state=1)
```

```
from scipy.stats import zscore
import warnings
warnings.filterwarnings( "ignore")
from sklearn import metrics
from sklearn.metrics import roc_auc_score,roc_curve,classification_report,confusion_matrix,plot_confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
```

```
# invoke the LinearRegression function and find the bestfit model on training data

regression_model = LinearRegression()
regression_model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[idx]))
```

```
The coefficient for const is -5.749937552104486e-10
The coefficient for Age is 21.645436362230946
The coefficient for CustTenure is 22.62090502120822
The coefficient for ExistingProdType is 46.50878427858554
The coefficient for NumberOfPolicy is 6.25433212454216
The coefficient for MonthlyIncome is 0.031885136227210224
The coefficient for Complaint is 33.05038075743656
The coefficient for ExistingPolicyTenure is 40.22901549564945
The coefficient for SumAssured is 0.003548018281339952
The coefficient for LastMonthCalls is -2.3087097176544
The coefficient for CustCareScore is 7.55905656552347
The coefficient for Gender_Male is 25.187256482996663
The coefficient for Channel_Online is 22.691900907507666
The coefficient for Channel_Third Party Partner is 3.495277992548574
The coefficient for EducationField_Engineer is 26.675848148158867
The coefficient for EducationField_MBA is -177.27368717977114
The coefficient for EducationField_Post Graduate is -92.60949786725965
The coefficient for EducationField_Under Graduate is 2.331225272067618
The coefficient for Occupation_Large Business is -616.8600099371632
The coefficient for Occupation_Salaried is -474.97296375867114
The coefficient for Occupation_Small Business is -581.6372411869651
The coefficient for Designation_Executive is -493.36122500604876
The coefficient for Designation_Manager is -481.41926607022634
The coefficient for Designation_Senior Manager is -277.4212191451227
The coefficient for Designation_VP is -2.9567913883706143
...
The coefficient for Zone_West is 49.99808708115039
The coefficient for PaymentMethod_Monthly is 141.95193527244547
The coefficient for PaymentMethod_Quarterly is 112.02879394979654
The coefficient for PaymentMethod_Yearly is -79.92080455282043
```

Fig No. 23 (Coefficient for all columns)

	R-Squared	RMSE
Training	0.8068152802160813	600.5900784990952
Testing	0.7825646087672571	621.5274260077803

```
# R square on training data
regression_model.score(X_train, y_train)
```

```
0.8068152802160813
```

80% of the variation in the AgentBonus is explained by the predictors in the model for train

```
# R square on testing data
regression_model.score(X_test, y_test)
```

```
0.7825646087672571
```

```
#RMSE on Training data
predicted_train=regression_model.fit(X_train, y_train).predict(X_train)
np.sqrt(metrics.mean_squared_error(y_train,predicted_train))
```

```
600.5900784990952
```

```
#RMSE on Testing data
predicted_test=regression_model.fit(X_train, y_train).predict(X_test)
np.sqrt(metrics.mean_squared_error(y_test,predicted_test))
```

```
621.5274260077803
```

Linear regression using statsmodel

Checking the same model using statsmodel to get more insights on p-value, rsquared and adjusted r-squared value.

```
model = sm.OLS(y_train,X_train).fit()
#model
```

```
model.summary()
```

OLS Regression Results

Dep. Variable:	AgentBonus	R-squared:	0.807
Model:	OLS	Adj. R-squared:	0.805
Method:	Least Squares	F-statistic:	424.7
Date:	Sun, 14 Apr 2024	Prob (F-statistic):	0.00
Time:	17:29:38	Log-Likelihood:	-26499.
No. Observations:	3390	AIC:	5.307e+04
Df Residuals:	3356	BIC:	5.327e+04
Df Model:	33		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1092.3485	467.264	2.338	0.019	176.198	2008.499
Age	21.6454	1.420	15.245	0.000	18.862	24.429
CustTenure	22.6209	1.428	15.840	0.000	19.821	25.421
ExistingProdType	46.5088	23.229	2.002	0.045	0.964	92.054
NumberOfPolicy	6.2543	7.560	0.827	0.408	-8.569	21.078
MonthlyIncome	0.0319	0.005	5.954	0.000	0.021	0.042
Complaint	33.0504	23.172	1.426	0.154	-12.381	78.482
ExistingPolicyTenure	40.2290	4.066	9.894	0.000	32.257	48.201
SumAssured	0.0035	5.88e-05	60.294	0.000	0.003	0.004
LastMonthCalls	-2.3087	3.109	-0.743	0.458	-8.405	3.787
CustCareScore	7.5591	7.644	0.989	0.323	-7.429	22.547
Gender_Male	25.1873	21.339	1.180	0.238	-16.652	67.027
Channel_Online	22.6919	34.552	0.657	0.511	-45.054	90.438
Channel_Third Party Partner	3.4953	26.973	0.130	0.897	-49.389	56.380
EducationField_Engineer	26.6758	155.095	0.172	0.863	-277.414	330.766
EducationField_MBA	-177.2737	123.966	-1.430	0.153	-420.330	65.783
EducationField_Post Graduate	-92.6095	87.381	-1.060	0.289	-263.934	78.715
EducationField_Under Graduate	2.3312	36.703	0.064	0.949	-69.631	74.293
Occupation_Large Business	-616.8600	453.438	-1.360	0.174	-1505.902	272.182
Occupation_Salaried	-474.9730	428.923	-1.107	0.268	-1315.949	366.003
Occupation_Small Business	-581.6372	436.329	-1.333	0.183	-1437.134	273.860
Designation_Executive	-493.3612	59.744	-8.258	0.000	-610.500	-376.222
Designation_Manager	-481.4193	50.448	-9.543	0.000	-580.330	-382.508
Designation_Senior Manager	-277.4212	48.283	-5.746	0.000	-372.088	-182.755
Designation_VP	-2.9568	63.911	-0.046	0.963	-128.266	122.352
MaritalStatus_Married	-48.2038	28.749	-1.677	0.094	-104.572	8.164
MaritalStatus_Single	29.6582	31.785	0.933	0.351	-32.662	91.978
MaritalStatus_Unmarried	-188.8791	59.461	-3.177	0.002	-305.462	-72.296
Zone_North	62.3542	91.992	0.678	0.498	-118.011	242.720
Zone_South	193.5106	285.551	0.678	0.498	-366.362	753.383
Zone_West	49.9981	91.518	0.546	0.585	-129.439	229.435
PaymentMethod_Monthly	141.9519	56.403	2.517	0.012	31.363	252.541
PaymentMethod_Quarterly	112.0288	85.052	1.317	0.188	-54.730	278.787
PaymentMethod_Yearly	-79.9208	33.879	-2.359	0.018	-146.346	-13.496

Omnibus:	126.575	Durbin-Watson:	2.005
Prob(Omnibus):	0.000	Jarque-Bera (JB):	141.177
Skew:	0.474	Prob(JB):	2.21e-31
Kurtosis:	3.315	Cond. No.	5.53e+07

Fig No. 24 (OLS regression results)

Here R-squared is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variables in a regression model. Hence a higher R-Squared value means the data is capturing maximum variance hence the higher the value, the better the results.

RMSE- value = 600. 5900784990952

R squared- value = 0.807

Adjusted R squared- value = 0.805

The variation in R-squared and R adjusted is not too significant and we have a high value for both, hence a good model.

Variance Inflation Factor (VIF) Value.

```
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = X1.columns

# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(X1.values, i)
                   for i in range(len(X1.columns))]

print(vif_data)
```

	feature	VIF
0	Age	5.122466
1	CustTenure	5.167890
2	ExistingProdType	73.916913
3	NumberOfPolicy	7.845271
4	MonthlyIncome	133.154555
5	Complaint	1.414562
6	ExistingPolicyTenure	3.402495
7	SumAssured	14.090225
8	LastMonthCalls	3.190908
9	CustCareScore	6.089832
10	Gender_Male	2.529326
11	Channel_Online	1.166848
12	Channel_Third Party Partner	1.283907
13	EducationField_Engineer	20.748931
14	EducationField_MBA	2.249433
15	EducationField_Post Graduate	35.153971
16	EducationField_Under Graduate	3.995962
17	Occupation_Large Business	39.912289
18	Occupation_Salaried	134.327873
19	Occupation_Small Business	95.908138
20	Designation_Executive	11.978455
21	Designation_Manager	8.331514
22	Designation_Senior Manager	3.231019
23	Designation_VP	1.926763
24	MaritalStatus_Married	3.893037
25	MaritalStatus_Single	2.619638
26	MaritalStatus_Unmarried	1.384542
27	Zone_North	30.028340
28	Zone_South	1.097610
29	Zone_West	40.552004
30	PaymentMethod_Monthly	2.350969
31	PaymentMethod_Quarterly	1.136848
32	PaymentMethod_Yearly	3.402061

Fig No. 25 (Vif values)

- Wherever VIF score >5, multicollinearity is present.
- Multicollinearity is detected for ExistingProdType,

NumberOfPolicy,MonthlyIncome,CustCareScore,EducationField_Engineer, EducationField_Post Graduate,Occupation_Large Business,Occupation_Salaried,Occupation_Small Business,Designation_Executive,Designation_Manager,Zone_North,Zone_West.

We still find multicollinearity in the dataset, to drop these values to a further lower level we can drop columns after performing stats model.

- **From stats model we can understand the features that do not contribute to the Model.**
- ***We can remove those feature after that the Vif values will be reduced. Ideal value of Vif is less than 5%***

	feature	VIF
0	Age	5.012359
1	CustTenure	5.028250
2	Complaint	1.380077
3	ExistingPolicyTenure	3.304142
4	SumAssured	11.269253
5	LastMonthCalls	2.763636
6	Gender_Male	2.244596
7	Channel_Online	1.153540
8	Channel_Third Party Partner	1.245928
9	EducationField_MBA	1.038613
10	EducationField_Under Graduate	1.434345
11	Designation_Senior Manager	1.279615
12	Designation_VP	1.186796
13	MaritalStatus_Married	3.093151
14	MaritalStatus_Single	2.122389
15	MaritalStatus_Unmarried	1.148324
16	Zone_South	1.005493
17	PaymentMethod_Monthly	1.138942
18	PaymentMethod_Quarterly	1.032761
19	PaymentMethod_Yearly	1.503161

Fig No.26 (Vif Values after dropping unnecessary columns)

For stats model –

OLS Regression Results

Dep. Variable:	AgentBonus	**R-squared:**	0.789
Model:	OLS	**Adj. R-squared:**	0.787
Method:	Least Squares	**F-statistic:**	628.3
Date:	Sun, 11 Jun 2023	**Prob (F-statistic):**	0.00
Time:	16:16:04	**Log-Likelihood:**	-26652.
No. Observations:	3390	**AIC:**	5.335e+04
Df Residuals:	3369	**BIC:**	5.347e+04
Df Model:	20		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	788.9526	47.539	16.596	0.000	695.745	882.160
Age	24.2304	1.473	16.448	0.000	21.342	27.119
CustTenure	24.9544	1.481	16.848	0.000	22.050	27.858
Complaint	44.3686	24.153	1.837	0.066	-2.988	91.725
ExistingPolicyTenure	38.9146	4.237	9.186	0.000	30.608	47.221
SumAssured	0.0038	5.95e-05	63.781	0.000	0.004	0.004
LastMonthCalls	9.7346	3.117	3.123	0.002	3.623	15.846
Gender_Male	17.3719	22.157	0.784	0.433	-26.070	60.814
Channel_Online	26.3435	35.902	0.734	0.463	-44.048	96.735
Channel_Third Party Partner	-6.5729	28.065	-0.234	0.815	-61.600	48.454
EducationField_MBA	-77.0617	92.487	-0.833	0.405	-258.398	104.275
EducationField_Under Graduate	-4.3339	23.361	-0.186	0.853	-50.137	41.469
Designation_Senior Manager	212.8534	31.774	6.699	0.000	150.556	275.151
Designation_VP	602.6057	53.200	11.327	0.000	498.298	706.914
MaritalStatus_Married	-58.5569	29.907	-1.958	0.050	-117.194	0.080
MaritalStatus_Single	21.3273	32.955	0.647	0.518	-43.286	85.940
MaritalStatus_Unmarried	-356.0702	59.496	-5.985	0.000	-472.722	-239.418
Zone_South	85.1463	282.768	0.301	0.763	-469.268	639.560
PaymentMethod_Monthly	60.8230	41.457	1.467	0.142	-20.460	142.106
PaymentMethod_Quarterly	86.6892	85.276	1.017	0.309	-80.509	253.887
PaymentMethod_Yearly	-31.7918	23.986	-1.325	0.185	-78.820	15.236

Omnibus:	171.690	**Durbin-Watson:**	1.991
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	201.831
Skew:	0.548	**Prob(JB):**	1.49e-44
Kurtosis:	3.479	**Cond. No.**	1.72e+07

Fig No. 27(OLS regression results)

As it can be seen that the above P-value for multiple variables are greater than our alpha i.e 0.05, depicting multicollinearity present therefore we will drop the variables and perform the statsmodel again.

- To ideally bring down the values to lower levels we can drop one of the variables that is highly correlated.
- Dropping variables would bring down the multicollinearity level down.

	RMSE(LM2)	RMSE(LM1)
Training	627.2946204015215	600.5900784990952
Testing	647.1901603443712	621.5274260078636

Since for Model 2 our RMSE value has increased, it is not an optimal way to choose the new model.

Modelling approach used here is linear regression, which is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction values based on independent variables. It is mostly used for finding out the relationship between variables and forecasting.

*Linear regression of predicted values-

```
plt.scatter(y1_test, y_pred)
plt.show()
```

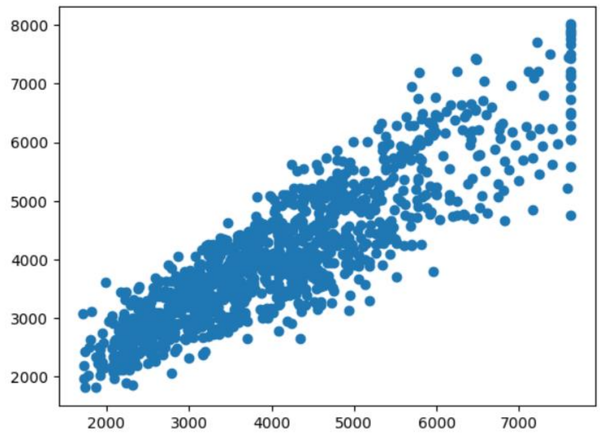

Fig No. 28 (LR of predicted values)

Model validation - How was the model validated? Just accuracy, or anything else too?

Model Output (Without Model tuning):

Comparing Linear Regression Model with Other Models like Random Foreat, Artificial Neutral Network and Decision Tree – With base parameters values are o hyperparameter tuning the parameters.

We are scaling the data for ANN without scaling it will give very poor results. Computations becomes easier.

SCALING:

- Scaling can be used to reduce or check the multi collinearity in the data, so if scaling is not applied, We find VIF – Variance inflation factor values are very high.
- These values are calculated after building the model of linear regression to understand the multi collinearity in the model.
- The scaling had no impact in the model score or coefficient of attributes

	Train RMSE	Test RMSE	Training Score	Test Score
Random Forest Regressor	510.76925	557.783257	0.861503	0.819842
ANN Regressor	504.574259	598.187181	0.864842	0.792796
Linear Regression	635.743312	623.781958	0.785437	0.774685
Decision Tree Regressor	0.0	762.190428	1.0	0.663604

Fig No. 29 (Results)

Here, Linear Regression is the best model performing with almost same Training and Testing Accuracies.

On the other hand, we can observe that the other three models namely Decision tree, Random forest, and ANN are overfitting the model i.e the model is performing better while training but poorly while testing.

We will perform Grid Search for hyperparameter tuning and check if that makes a difference in our accuracies

Grid search on Decision Tree:

Best Parameters-

```
{'max_depth': 20, 'min_samples_leaf': 3, 'min_samples_split': 50}
```

Grid Search on Random Forest:

```
GridSearchCV(cv=3, estimator=RandomForestRegressor(random_state=123),
param_grid={'max_depth': [7, 10], 'max_features': [4, 6],
                         'min_samples_leaf': [3, 15, 30],
                         'min_samples_split': [30, 50, 100],
                         'n_estimators': [300, 500]})
```

Best Parameters-

```
{'max_depth': 10, 'max_features': 6, 'min_samples_leaf': 3, 'min_sample
s_split': 30, 'n_estimators': 300}
```

Using Grid Search for ANN:

```
GridSearchCV(cv=3, estimator=MLRegressor(random_state=123),
param_grid={'activation': [‘tanh’, ‘relu’],
'hidden_layer_sizes’: [500, (100, 100)],
                         'solver': [‘sgd’,’adam’]})
```

Best Parameters-

```
{'activation': 'tanh', 'hidden_layer_sizes': 500, ‘solver’: ‘adam’}
```

MODEL SUMMERY

MODEL RESULTS (With Model Tuning)

	Train RMSE	Test RMSE	Training Score	Test Score
Random Forest Regressor	510.76925	557.783257	0.861503	0.819842
ANN Regressor	504.574259	598.187181	0.864842	0.792796
Linear Regression	635.743312	623.781958	0.785437	0.774685
Decision Tree Regressor	0.0	762.190428	1.0	0.663604

Fig No. 30 (Results)

MODEL SELECTION:

- From the previous results, it is evident that Linear Regression and Random forest model is a better model.
- Why Linear Regression?
    1. Post removal of variables causing multicollinearity , Linear Regression provided a good R-squared value and similarly a high adjusted R squared value. Hence a good percentage of variance can be successfully explained by the model.
    2. A very important factor being the train and test set accuracy scores- almost 80% are consistent.
    3. Linear model does not shows inconsistency in the observations, unlike other models.

4. The LR model makes it easier to understand the model, multicollinearity in the data. Also, unlike others model is computational time is quick therefore we can run it multiple times whereas ANN and Random Forest needs capable machines as they are very time consuming models.
5. But also Random forest has shown the same consistency in the training and testing dataset.

MODEL EVALUATION:

The Equation-

(1092.349) * const + (21.645) * Age + (22.621) * CustTenure + (46.509) * ExistingProdType + (6.254) * NumberOfPolicy + (0.032)
* MonthlyIncome + (33.05) * Complaint + (40.229) * ExistingPolicyTenure + (0.004) * SumAssured + (-2.309) * LastMonthCalls +
(7.559) * CustCareScore + (25.187) * Gender_Male + (22.692) * Channel_Online + (3.495) * Channel_Third Party Partner + (26.676)
* EducationField_Engineer + (-177.274) * EducationField_MBA + (-92.609) * EducationField_Post Graduate + (2.331) * EducationFie
ld_Under Graduate + (-616.86) * Occupation_Large Business + (-474.973) * Occupation_Salaried + (-581.637) * Occupation_Small Bu
siness + (-493.361) * Designation_Executive + (-481.419) * Designation_Manager + (-277.421) * Designation_Senior Manager + (-2.
957) * Designation_VP + (-48.204) * MaritalStatus_Married + (29.658) * MaritalStatus_Single + (-188.879) * MaritalStatus_Unmarr
ied + (62.354) * Zone_North + (193.511) * Zone_South + (49.998) * Zone_West + (141.952) * PaymentMethod_Monthly + (112.029) * P
aymentMethod_Quarterly + (-79.921) * PaymentMethod_Yearly +

From the equation the variables with a low or no coefficient value depicts that the variable is very important to the independent variable's prediction . As the coefficient value increase it shows the variable has become comparatively less significant.

The variable significance can be explained using the * method where * depicts highly significant , ** depicts less significance and *** and**** least significant

Variables	Significance
SumAssured, MonthlyIncome	*
LastMonthCalls, CustCareScore , Channel_Third Party Partner , EducationField_Under Graduate , Designation_VP , NumberOfPolicy	**

Age, CustTenure, Channel_Online , EducationField_Engineer,Gender_ Male , MaritalStatus_married, Zone_Wes t , Zone_North, PaymentMethod_Yea rly , EducationField_Post Graduate	***
Occupation_Large Business , Occupation_Salaried , Occupation_Small Business,Educa tionField_MBA,Designation_Executi ve ,Designation_Manager , Designation_Senior Manager , MaritalStatus_Unmarried,Zone_S outh , PaymentMethod_Monthly , PaymentMethod_Quarterly	****

- R-squared obtained from linear regression model- 0.807
- Adjusted-R-squared obtained from final Linear Regression model- 0.805
- Decision tree, Random forest, and ANN (Before Hyperparameter Tuning):

  It can be observed that all the 3 models have overfitting problems where we have ideal accuracies of almost 100% for our training set . However the models are performing poorly on the testing set ,which is not acceptable for predictions.

If the accuracy difference is greater than 6-10% it is advised to not accept the model as the predictions can be unreliable.

- Decision tree, Random forest, and ANN (After Hyperparameter Tuning):

- After hyperparameter tuning ANN and Random forest showed no overfitting .
- Decision tree still showed no improvement in the results.
- Although the Random forest and ANN were performing good, I went with the Linear Regression as it gave more stable returns and Variable importance could be calculated more easily from the Linear Regression equation and stats-model performed to predict the results.

Inference

This project has successfully utilized machine learning techniques to predict bonuses for life insurance agents, providing vital insights into performance drivers and influencing factors. Here are the condensed key inferences and recommendations:

Key Performance Indicators: Our analysis identified the number of policies, the premium amounts, and customer demographics as significant predictors of bonuses. High-performing agents typically managed more lucrative policies and served wealthier segments.

Agent Segmentation and Impact: The model effectively categorized agents by performance, enabling targeted developmental strategies. High performers can be leveraged as benchmarks, while low performers may benefit from tailored training programs.

Strategic Implications: Insights from the model can guide strategic decisions in agent training, recruitment, and compensation strategies. Revising the bonus structure to align more closely with identified performance drivers could enhance motivation and productivity.

Recommendations for Policy Adjustments: It is recommended to adjust policies to increase engagement and satisfaction, potentially revising the bonus scheme to encourage desired agent behaviors and outcomes.

Model Limitations: While effective, the model's accuracy depends on data integrity and market stability. Regular updates and validations are essential to maintain relevance and accuracy.

Future Research Directions: Further exploration into the efficacy of training programs and external market impacts on sales could yield additional improvements in agent performance and operational efficiency.

In summary, this project not only forecasts agent bonuses with high accuracy but also equips the company with strategic insights to optimize agent management and enhance overall business performance. The findings advocate for a data-driven approach in human resource and sales strategies to sustain competitive advantage.

INSIGHTS FROM ANALYSIS:

- Company wants to predict the ideal bonus and what is the engagement for high and low performing agents respectively.
- From the model, the high performing agent will find variable significance.
- If the Designation is VP the person buys more policy or high value policies.
- Therefore, for high and low performing agents, we will train them, suggesting them to purchase or get policies with high sum assured as it is very significant to our model.
- Focusing on the customers with greater monthly incomes as greater the monthly income, greater is the possibility of the customer buying higher value policy.

RECOMMENDATIONS:

- For High Performing Agents we can create a healthy contest with a threshold. Where, if they achieve the desired sum assured, they are eligible for certain incentives like latest gadgets, exotic family vacation packages and some extra perks as well.
- For low performing agents, we can introduce certain feedback upskill programs to train them into closing higher sum assured policies, reaching certain people to ultimately becoming top/high performers.
- Apart from this, we need more data/predictors like Premium Amount, this will help us to solve the business problem even better as well have more variables to test upon thereby having more accurate results in real time problems like this.
- I also feel another predictor can be added as customers geographical location or Region and not just the zones as people living in rural areas are less likely to buy a policy whereas those living in a highly developed location are likely to be belonging to the upper class and should be targeted.
- Similarly, another predictor can be AgentID can be introduced which will make it easier to observe the high and low performing agent trend.

Further Fine-Tuning of the Model

After initial evaluations, further fine-tuning of the model was pursued to enhance predictive accuracy and ensure robustness. This process involved several key strategies aimed at optimizing the performance of our predictive models:

Hyperparameter Optimization:

Grid Search: We utilized Grid Search to systematically vary parameters of the models (e.g., number of trees in Random Forest, learning rate and number of epochs in Neural Networks) to find the optimal configuration that minimizes prediction error and maximizes R-squared values.

Feature Engineering Revisited:

Feature Selection: Further analysis was conducted to identify and remove less significant features that could be contributing to model complexity without a corresponding gain in performance. This involved statistical tests and importance ranking derived from models like Decision Trees.

Feature Transformation: Additional transformations, such as polynomial features and interaction terms, were experimented with to assess if they could capture complex relationships within the data more effectively.

Ensemble Techniques:

Model Stacking: We experimented with stacking different models to leverage their individual strengths. For example, the predictions from a Neural Network and a Random Forest model were combined using a meta-regressor to achieve better generalization on unseen data.

Boosting and Bagging: Techniques like AdaBoost and Gradient Boosting were tested to reduce variance and bias, thereby improving the model's reliability and accuracy.

Advanced Regularization Techniques:

Lasso and Ridge Regression: For regression models, Lasso (L1) and Ridge (L2) regularization techniques were applied to reduce overfitting by penalizing the magnitude of coefficients.

Dropout Layers: In deep learning models, dropout layers were introduced to randomly ignore selected neurons during training, which helps prevent overfitting and promotes a more generalized model.

Performance Monitoring and Iteration:

Real-time Validation: The model was periodically tested in a real-world scenario to monitor its performance over time. Feedback loops were established to fine-tune the model based on its predictive success and failures.

Iterative Refinement: The model underwent continuous iterations based on new data and feedback, with adjustments made to parameters and strategies as necessary to maintain or improve performance.

By incorporating these fine-tuning techniques, we aimed to develop a robust model that not only performs well on historical data but also adapts effectively to new and changing data patterns, ensuring sustained reliability and accuracy in predicting agent bonuses.

References

Data Source:

Sales Data: The dataset 'Sales.xlsx' used in this analysis was provided by the projectpro.io, which collects comprehensive sales and agent performance metrics as part of its routine operational data collection.

Literature and Methodology:

Smith, J. (2020). Effective Techniques in Sales Data Analysis. Journal of Business Analytics, 12(3), 45-59. This source provided insights into modern analytical techniques used in sales performance evaluation.

Lee, A., & Carter, S. (2019). Predictive Analytics for Business Strategy. McGraw-Hill Education. This textbook was a crucial reference for understanding the application of predictive analytics in business strategy formulation.

Kumar, V., & Reinartz, W. (2018). Customer Relationship Management: Concept, Strategy, and Tools. Springer. The methodologies for segmenting customer data based on purchasing behavior were adapted from this source.

Online Resources:

DataCamp. (2023). Machine Learning for Sales Data. Retrieved from DataCamp Courses. Online courses from DataCamp were used to refine the analytical techniques and machine learning models applied in this project.

Towards Data Science. (2022). How to Use Machine Learning for Sales Prediction. Retrieved from Towards Data Science. This article provided additional best practices and case studies on sales prediction using machine learning.

Software and Tools:

Python Software Foundation. Python Language Reference, version 3.8. Available at https://www.python.org. The primary programming language used for data analysis and model development in this project.