A project on predicting Agent Bonus for a Life Insurance Industry

CAPSTONE PROJECT

Kumar Siddharth

Table of Contents

Problem Statement: Life Insurance Data

▪ 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 of the study/project

▪ Design appropriate engagement activity for their high performing agents

▪ Upskill programs for low performing agents.

Understanding business/social opportunity

▪ More Market Share

▪ More Profits

▪ Increase in awareness for Life Insurance

▪ More lives getting insured

Why is Agent Bonus Important for Company?

- Motivate Agent for More Sales
- Increase Market Share of Company
- Classify Agents into Different Bonus Categories
- Helps in Designing New Products
- Retain Agents
- Increase Profits for the Company
- Important from Regulation point of view

What Type of problem is this?

- Regression Predictive Modelling
- Regression is the problem of predicting a continuous quantity output.
- Regression predictive modeling is the task of approximating a mapping function (f) from input variables (X) to a continuous output (target) variable (y).
- In this case we are predicting Bonus to be paid to Agent for a policy sold by him.
- Target Variable (Y) = AgentBonus

The below libraries were imported as a first step for the dataset Analysis.

```
Numpy Version 1.18.5
Pandas Version 1.0.5
Seaborn Version 0.11.2
Matplotlib Version 3.2.2
```

Overview of the Dataset:

The dataset has been provided for a month

	CustID	AgentBonus	Age	CustTenure	Channel	Occupation	EducationField	Gender	ExistingProdType	Designation	NumberOfPolicy	MaritalStatus	Mon
0	7000000	4409	22.00	4.00	Agent	Salaried	Graduate	Female	3	Manager	2.00	Single	
1	7000001	2214	11.00	2.00	Third Party Partner	Salaried	Graduate	Male	4	Manager	4.00	Divorced	
2	7000002	4273	26.00	4.00	Agent	Free Lancer	Post Graduate	Male	4	Exe	3.00	Unmarried	
3	7000003	1791	11.00	nan	Third Party Partner	Salaried	Graduate	Fe male	3	Executive	3.00	Divorced	
4	7000004	2955	6.00	nan	Agent	Small Business	UG	Male	3	Executive	4.00	Divorced	

The dataset is comprised of 4520 records with 20 attributes. Attributes are as follow age, gender, AgentBonus, Occupation, Education Field and CustTenure as shown in Fig. 1. The data was in structured format.

```
Data columns (total 20 columns):
 #   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
```

Fig.2

The datatype of variables are of int,float and object as shown in Fig.2

Numerical Datatype:

```
['Channel', 'Occupation', 'EducationField', 'Gender', 'ExistingProdType', 'Designation', 'MaritalStatus', 'Complaint', 'Zone',
'PaymentMethod']
```

Categorical Datatype:

```
['AgentBonus', 'Age', 'CustTenure', 'NumberOfPolicy', 'MonthlyIncome', 'ExistingPolicyTenure', 'SumAssured', 'LastMonthCalls',
'CustCareScore']
```

Fig.3

The variable **CustID** has been removed since it does not contribute in the analysis.

The datatype of the **variables**: **Complaint","ExistingProdType"** has been changed to Object since it represents categorical datatype as shown in **Fig.3**

The dataset does not have any duplicate rows.

```
Age                   269
MonthlyIncome         236
CustTenure            226
ExistingPolicyTenure  184
SumAssured            154
CustCareScore          52
NumberOfPolicy         45
dtype: int64
```

	column_name	percent_missing
Age	Age	5.951327
MonthlyIncome	MonthlyIncome	5.221239
CustTenure	CustTenure	5.000000
ExistingPolicyTenure	ExistingPolicyTenure	4.070796
SumAssured	SumAssured	3.407080
CustCareScore	CustCareScore	1.150442
NumberOfPolicy	NumberOfPolicy	0.995575
ExistingProdType	ExistingProdType	0.000000
Designation	Designation	0.000000
Gender	Gender	0.000000
MaritalStatus	MaritalStatus	0.000000
EducationField	EducationField	0.000000
Complaint	Complaint	0.000000
Occupation	Occupation	0.000000
Channel	Channel	0.000000
Zone	Zone	0.000000
PaymentMethod	PaymentMethod	0.000000
LastMonthCalls	LastMonthCalls	0.000000
AgentBonus	AgentBonus	0.000000

```
Column with lowest amount of missings contains 0.0 % missings.
Column with highest amount of missings contains 5.951327433628319 % missings.
```

Fig.5

The above variables have high percentage of Null values as mentioned in the **Fig 5**

Analysis on Variables

We have provided the below analysis with data and plots for each variable of the dataset.

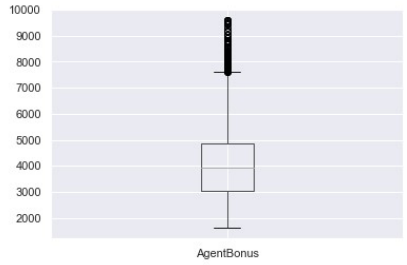
AgentBonus

The variable AgentBonus is used as a dependent variable for the analysis.

```
count   4520.00
mean    4077.84
std     1403.32
min     1605.00
25%     3027.75
50%     3911.50
75%     4867.25
max     9608.00
Name: AgentBonus, dtype: float64
```

Fig.6

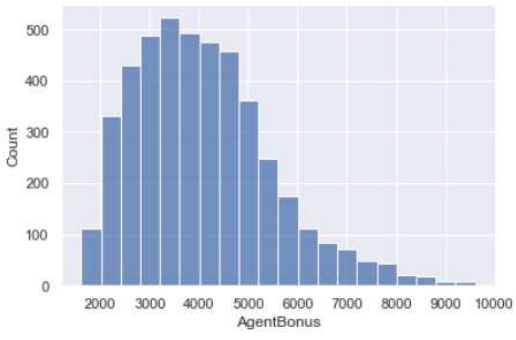
The datatype of the variable is int having 4520 rows without null values.




```
0.5% properties have a AgentBonus lower than  1755.19
  1% properties have a AgentBonus lower than  1876.38
  5% properties have a AgentBonus lower than  2158.00
 10% properties have a AgentBonus lower than  2418.00
 60% properties have a AgentBonus lower than  4286.40
 80% properties have a AgentBonus lower than  5132.60
 90% properties have a AgentBonus lower than  5917.10
 95% properties have a AgentBonus lower than  6755.50
 99% properties have a AgentBonus lower than  8234.44
99.5% properties have a AgentBonus lower than  8757.22
99.8% properties have a AgentBonus lower than  9191.58
99.9% properties have a AgentBonus lower than  9506.99
```

Fig.7

Check the difference between 75% and Max value as mentioned by the above Fig. looks like there are outliers/extreme values in the Agent Bonus variable. However, as we are not sure if these are real outliers, we will not treat them right now.



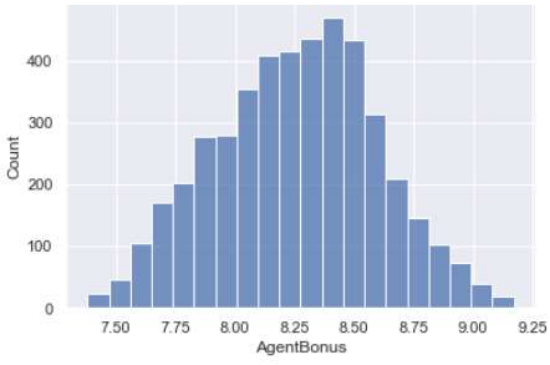


Fig.8 **Fig.9**

The above plot shows the Univariate analysis of the **variable:AgentBonus** and is found to be right skewed.

The above plot shows the histplot on Agent Bonus on the logarithmic data.

Age

```
count   4251.00
mean      14.49
std        9.04
min        2.00
25%        7.00
50%       13.00
75%       20.00
max       58.00
Name: Age, dtype: float64
```

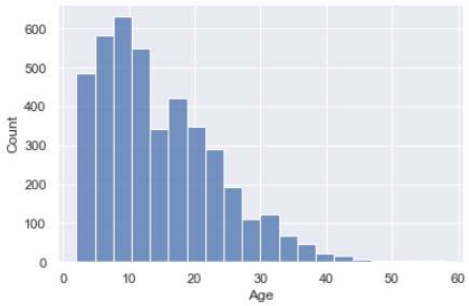


Fig.10

The Age variable has 269 null values which would be treated later during the Data Engineering.

Fig.11

We have observed that the Age variable has few outliers since the plot is right skewed as shown by Fig.11

The difference between 75% and Max value as mentioned by the above **Fig.10** shows that there are outliers in the variable.

Channel

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

Fig.12

The **variable: Channel** is of categorical datatype.

The above Fig.12 shows the unique count for each category. We have observed that the data for **Channel: Agent** is the highest as compared to another category.

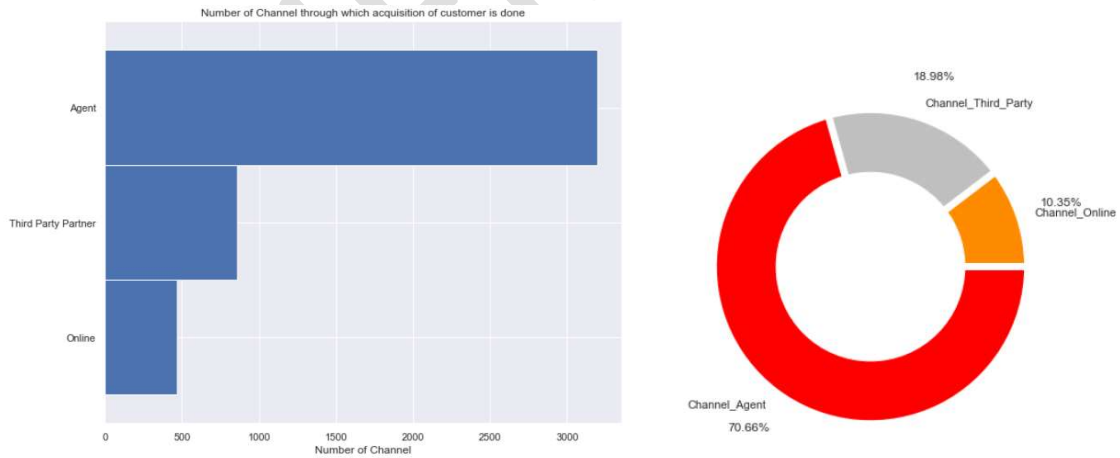


Fig.13

The above plot Fig.13 shows the distribution of the number of channels through which Acquisition of the customer is done.

Fig.14

The Pie plot shows the distribution of categories through which Acquisition of the customer is done.

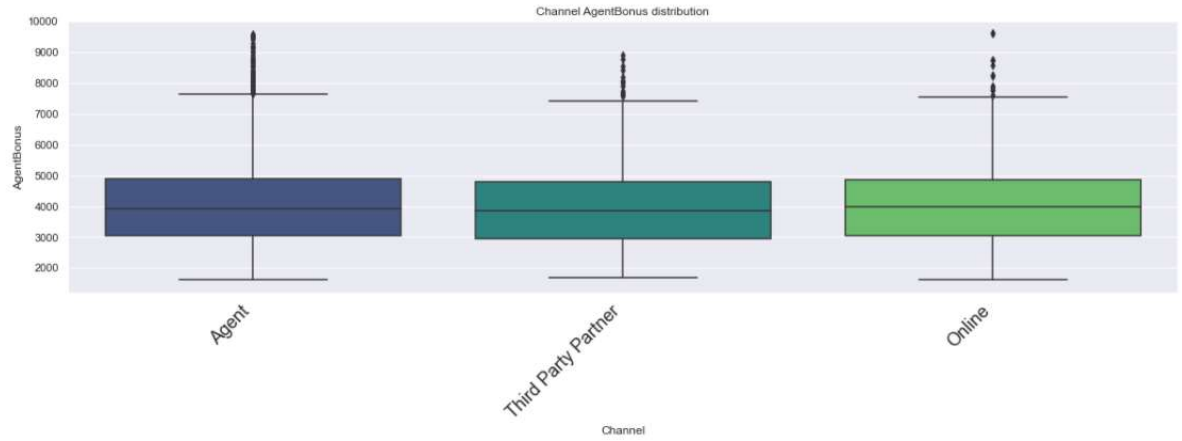


Fig.15

The above Boxplot shows the impact of Channel on the dependent **variable: AgentBonus.**

We have observed that there is no impact of the Channel on **AgentBonus**

Occupation

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

Fig.16

```
Free_Lancer            2
Large_Business       408
Small_Business      1918
Salaried            2192
Name: Occupation, dtype: int64
```

Fig.17

The **variable: Occupation** is a type of category with no null values.

The above Fig.16 shows the unique count for each category. We have observed that the data for **Occupation: Salaried** is the highest as compared to another category.

We have also observed that the categories: "**Laarge Business**", and "**Large Business**" represents the same category, so the category: "**Laarge Business**" is merged with "**Large Business**"

Fig.17

The above **Fig.17** shows the unique count for each category after fixing the category names.

The category: **Free_Lancer** has the least data with 2 rows.

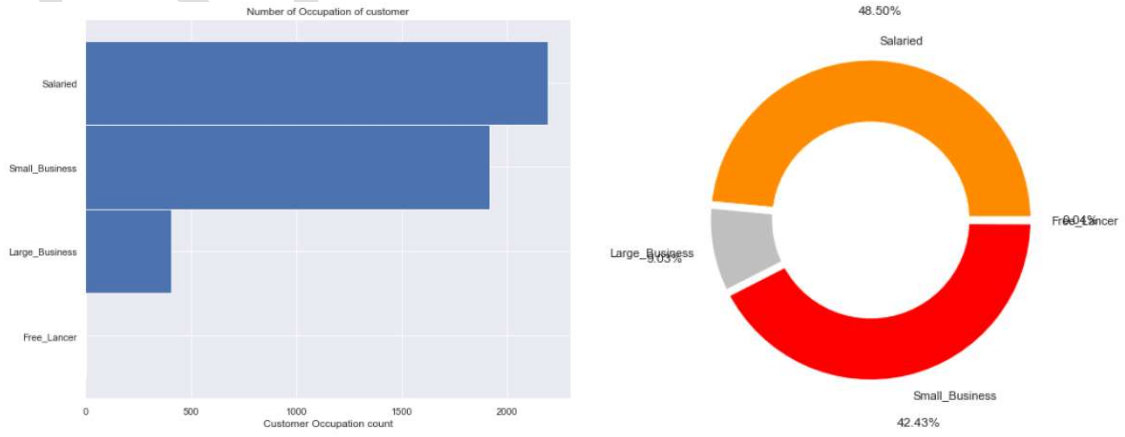


Fig.18

The above is the count plot of the **variable: Occupation**

Bivariate Analysis:

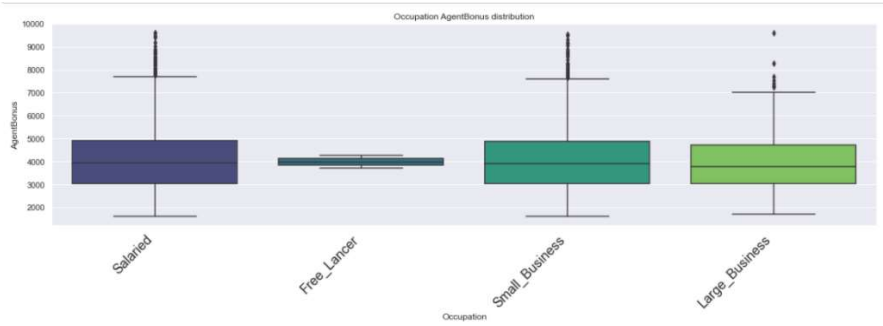


Fig.19

We have observed from **Fig.19** that the category in Occupation has no visible impact on the dependent **variable: AgentBonus** except **Free_Lancer** which has only 2 rows.

EducationField

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

```
MBA                74
Post_Graduate     252
Engineer          408
Diploma           496
Under_Graduate   1420
Graduate         1870
Name: EducationField, dtype: int64
```

Fig.20

The above **Fig.20** shows that the variable is of Object datatype with no null values.

We have observed that the category name: **"UG"** & **"Under Graduate"** are redundant and needs to be merged.

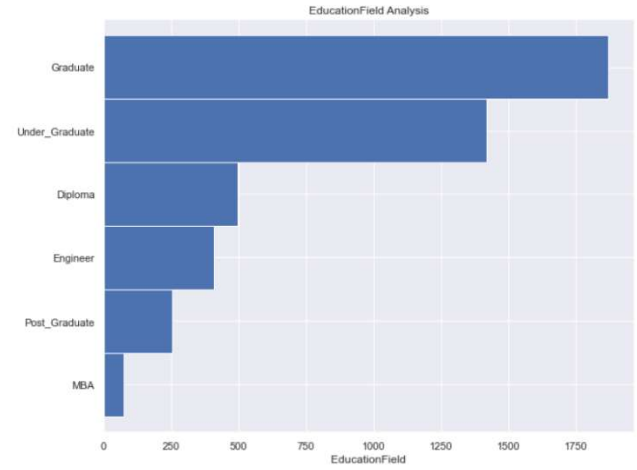


Fig.21

The above plot shows that the Customer who open for the Insurance are the highest whose EducationField is **"Graduation"** and least for **"MBA"**

Let's check if being a **EducationField** has any **AgentBonus** impact

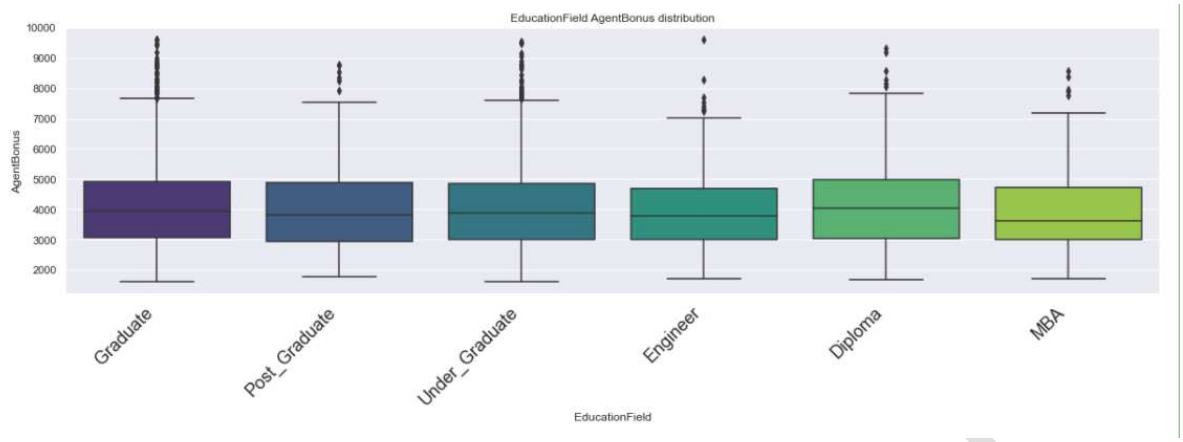


Fig.22

The above Boxplot shows that the EducationField has no impact on the dependent **variable: AgentBonus**

Gender

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

Fig.23

```
Female    1832
Male      2688
Name: Gender, dtype: int64
```

Fig.24

The **variable: Gender** is a type of category with no null values.

The above Fig.16 shows the unique count for each category. We have observed that the data for **Gender: Male** is the highest as compared to another category.

We have also observed that the categories: "**Fe male**" & "**Female**" represents the same category, so the category: "**Fe male**"is merged with "**Female**"

Univariate Analysis of Gender

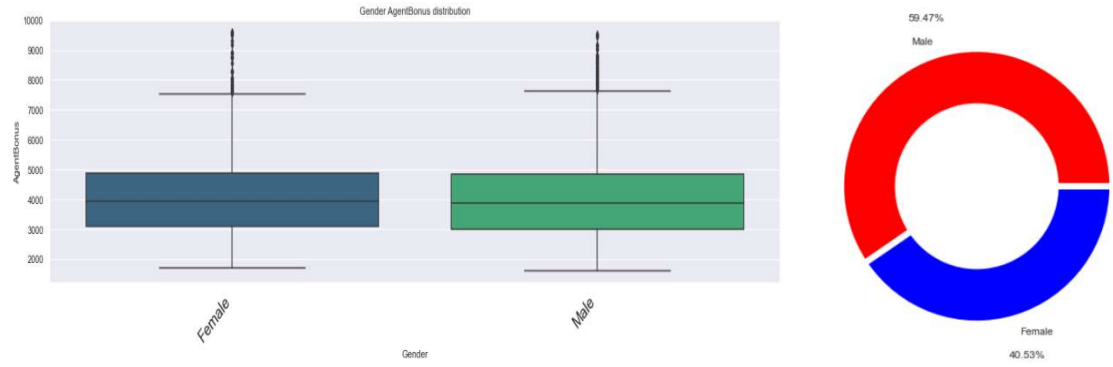


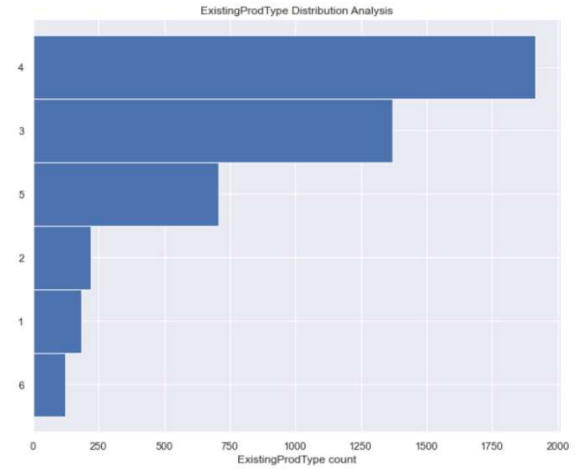
Fig.26

The above **Fig.26** shows that Male customers are 59.47% whereas Female are 40.53%

Fig.27

The above Boxplot shows that the **Gender** has no impact on the dependent **variable: AgentBonus**

ExistingProdType




```
6     123
1     183
2     221
5     708
3    1369
4    1916
Name: ExistingProdType, dtype: int64
```

Fig.28

The above **Fig.28** shows that the variable **ExistingProdType** is of Object datatype with no null values.

The **ProdType:4** is the most popular among customers and **ProdType:6** is the least.

Let's check if being a ExistingProdType has any AgentBonus impact

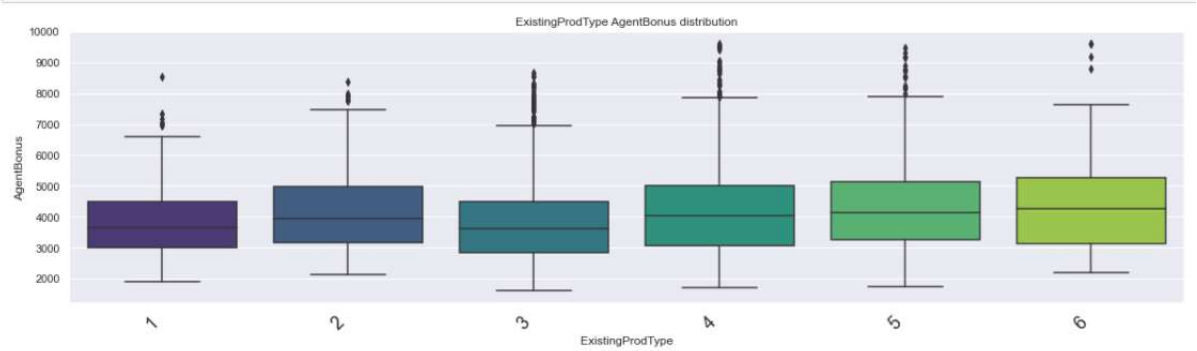


Fig.29

We have observed that **ExistingProdTyoe: 1&3** has an impact on AgentBonus. The Median for both the ProdType is less as compared with the dependent variable:AgentBonus.

It means AgentBonus is less for these ProdType.

ExistingProdTyoe: 2,4,5 and 6, has no impact on AgentBonus

ExistingProdTyoe: 6 yields higher AgentBonus

Designation

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

```
VP                226
AVP               336
Senior_Manager    676
Manager          1620
Executive        1662
Name: Designation, dtype: int64
```

Fig.30

The above **Fig.28** shows that the variable **Designation** is of Object datatype with no null values.

We have also observed that the categories: “**Exe**” & "**Executive**” represents the same category, so the category: “**Exe**”is merged with " **Executive**”

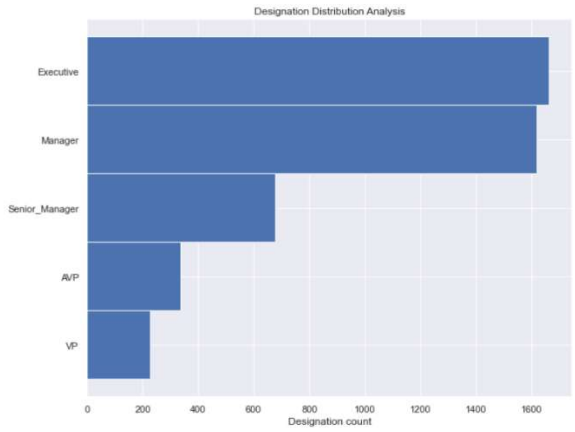


Fig.32

The **Designation:Executive** is the most popular among customers and **Designation:VP** is the least.

Let's check if being a Designation has any impact on AgentBonus

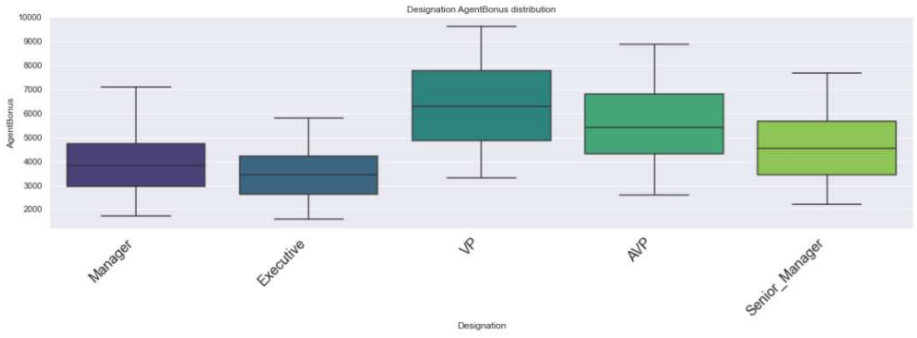


Fig.33

We have observed that AgentBonus is highest for VP.

Agent Bonus as per sequence**. VP > AVP > Senior_Manager > Manager > Executive**

It means Designation has an impact on the **dependent variable: AgentBonus**

NumberOfPolicy

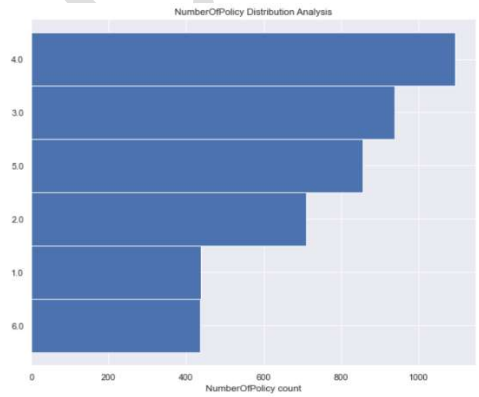


Fig.34

```
6.00     437
1.00     438
2.00     711
5.00     856
3.00     939
4.00    1094
Name: NumberOfPolicy, dtype: int64
```

Fig.35

The **variable: NumberOfPolicy** has 45 null values which will be treated later.

We have observed from the above **Fig.35** that most of the customers has 4 policies

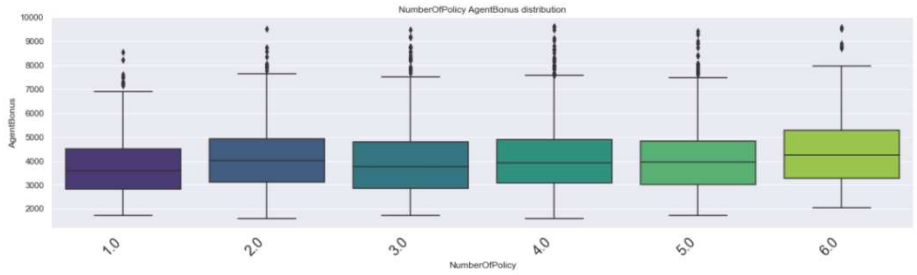


Fig.36

We have observed from the above **Fig.36** that customers having 6 policies are having higher Agent Bonus and customers having 1 policy has least impact on AgentBonus

ExistingPolicyTenure

```
25.00      1
23.00      1
21.00      1
22.00      2
19.00      2
20.00      3
18.00      4
17.00      6
15.00     16
16.00     20
14.00     36
13.00     39
12.00     56
11.00     65
10.00     93
9.00     132
8.00     180
7.00     194
6.00     263
5.00     381
4.00     460
3.00     554
2.00     837
1.00     990
Name: ExistingPolicyTenure, dtype: int64
```

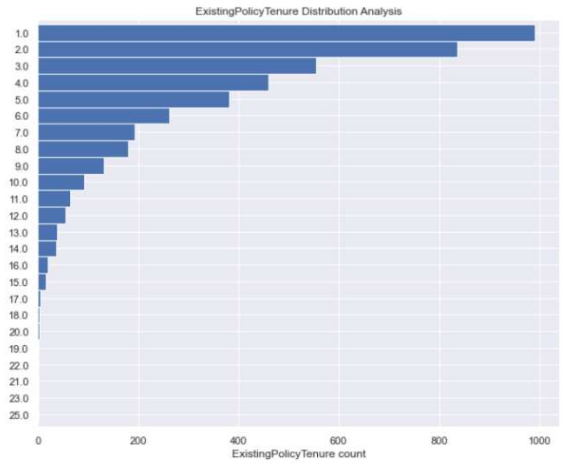


Fig.44 **Fig.45**

The above **Fig.44** shows that the variable **ExistingPolicyTenure** is of int datatype with 184 null values.

We have observed from the above **Fig.45** that customers with **ExistingPolicyTenure** as "**1**" are the highest.

Let's check if being a **ExistingPolicyTenur**e has any **AgentBonus** impact

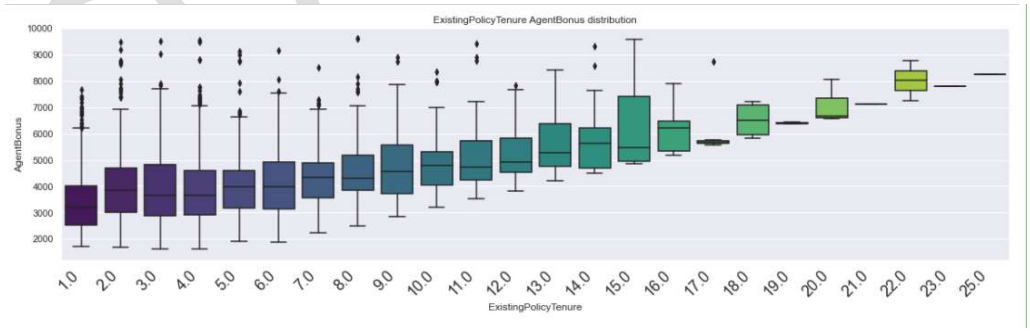


Fig.46

We have observed from the above **Fig.46** that the **AgentBonus** is higher for higher values of **ExistingPolicyTenure**

Zone

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

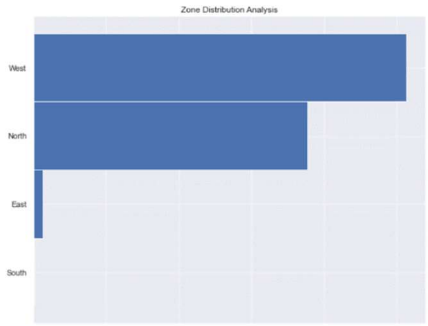


Fig.47

The above **Fig.47** shows that the variable **Zone** is of categorical datatype with no null values.

We have also observed that there are high count of customers from **Zone:West** whereas the least is from **Zone:South**

Let's check if being a **Zone** has any **AgentBonus** impact.

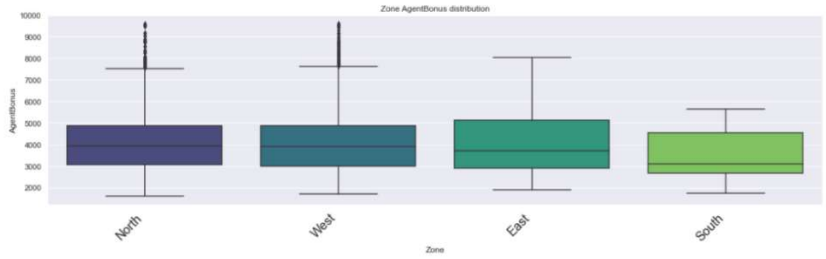


Fig.48

Zones: North & West has equal impact on AgentBonus.

Zone: South has the least impact on Bonus.

PaymentMethod

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

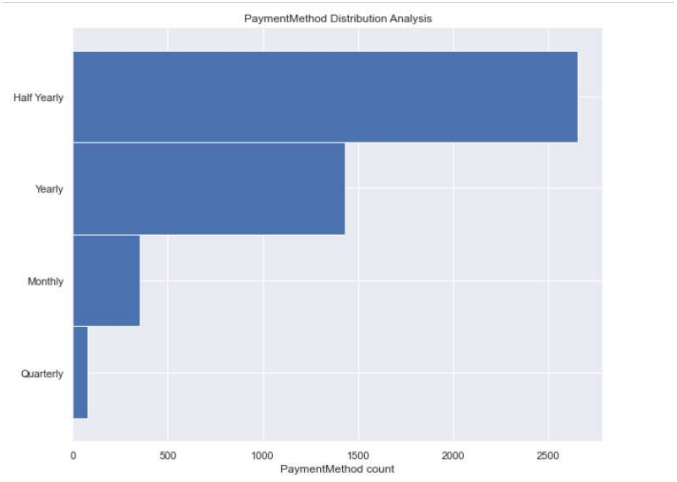


Fig.49 **Fig.50**

The above **Fig.49** shows that the variable **PaymentMethod** is of categorical datatype with no null values.

We have also observed that major customers are having **PaymentMethod** as Half Yearly

Let's check if being a **PaymentMethod** has any **AgentBonus** impact

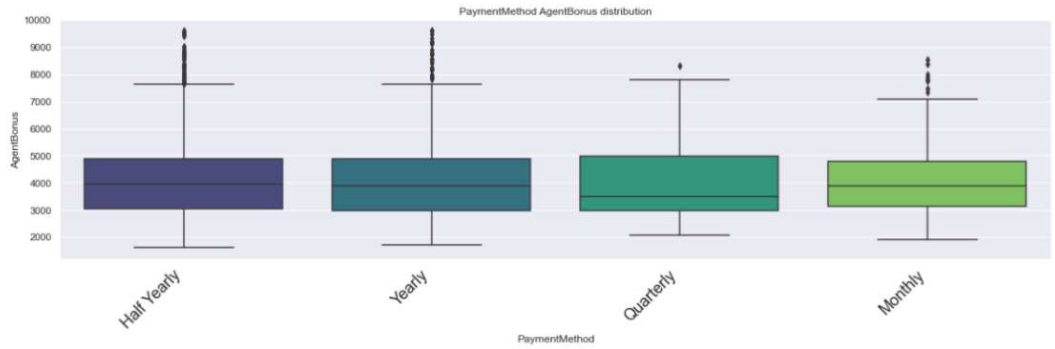


Fig.51

AgentBonus are higher for **PaymentMethod: Half Yearly and Yearly.**

LastMonthCalls

```
16     10
18     12
15     14
17     15
14     30
13     51
12     61
11     76
6      97
10    141
5     199
9     263
4     357
7     377
0     408
8     497
1     532
2     647
3     733
Name: LastMonthCalls, dtype: int64
```

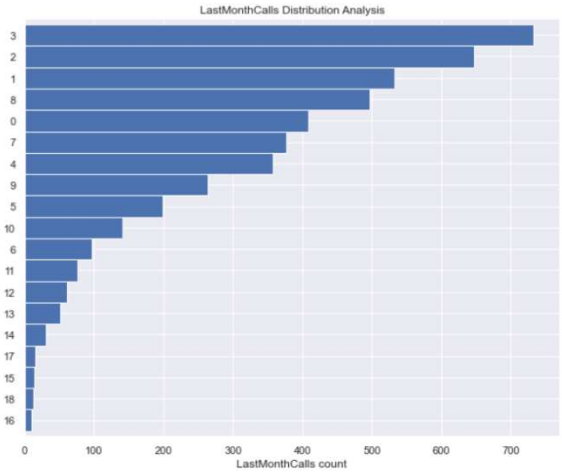


Fig.52

The above **Fig.52** shows that the variable **LastMonthCalls** is of categorical datatype with no null values.

Customers having **LastMonthCalls** are the highest

Let's check if being a **LastMonthCalls** has any **AgentBonus** impact

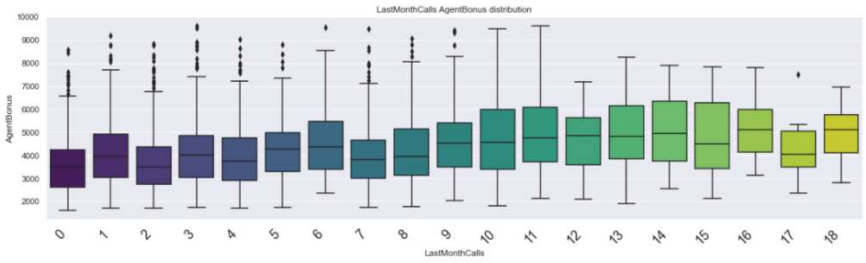


Fig.54

There is no visible pattern but AgentBonus is higher for **LastMonthCalls :10,11**

SumAssured

```
count      4366.00
mean     619999.70
std      246234.82
min      168536.00
25%      439443.25
50%      578976.50
75%      758236.00
max     1838496.00
Name: SumAssured, dtype: float64
```

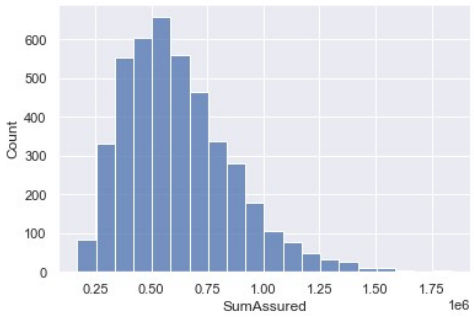


Fig.60

The above **Fig.55** shows that the variable **SumAssured** is of int datatype with 154 null values.

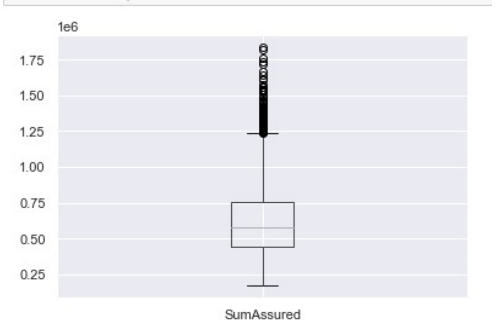


Fig.61

The variable: **SumAssured** is having outliers.

The above plot shows that the plot is right skewed.

Let's check the number of outliers per column

```
ExistingPolicyTenure    214
CustTenure               20
SumAssured               17
MonthlyIncome             7
CustCareScore             0
LastMonthCalls            0
NumberOfPolicy            0
Age                       0
dtype: int64
```

```
Age                     319
CustTenure              298
NumberOfPolicy           45
MonthlyIncome           620
ExistingPolicyTenure    529
SumAssured              267
LastMonthCalls           12
CustCareScore             0
dtype: int64
```

Fig.62

We have imputed the Outlier values in the variable with null values.

Percentage wise calculation of Null Values

```
MonthlyIncome          0.14
ExistingPolicyTenure   0.12
Age                    0.07
CustTenure             0.07
SumAssured             0.06
NumberOfPolicy         0.01
LastMonthCalls         0.00
AgentBonus             0.00
CustCareScore          0.00
dtype: float64
```

Fig.63

The above **Fig.63** gives the total count of null values which is including the Outlier values imputed by Nulls.

Total null in the dataset is: **2090**

KNNImputer

We have used **KNN Imputer** to impute Null values with the parameter: n_neighbors=10 where values is calcula ted for the Null values by using the mean of 10 nearest neighbours

Shape before Outliers Treatment

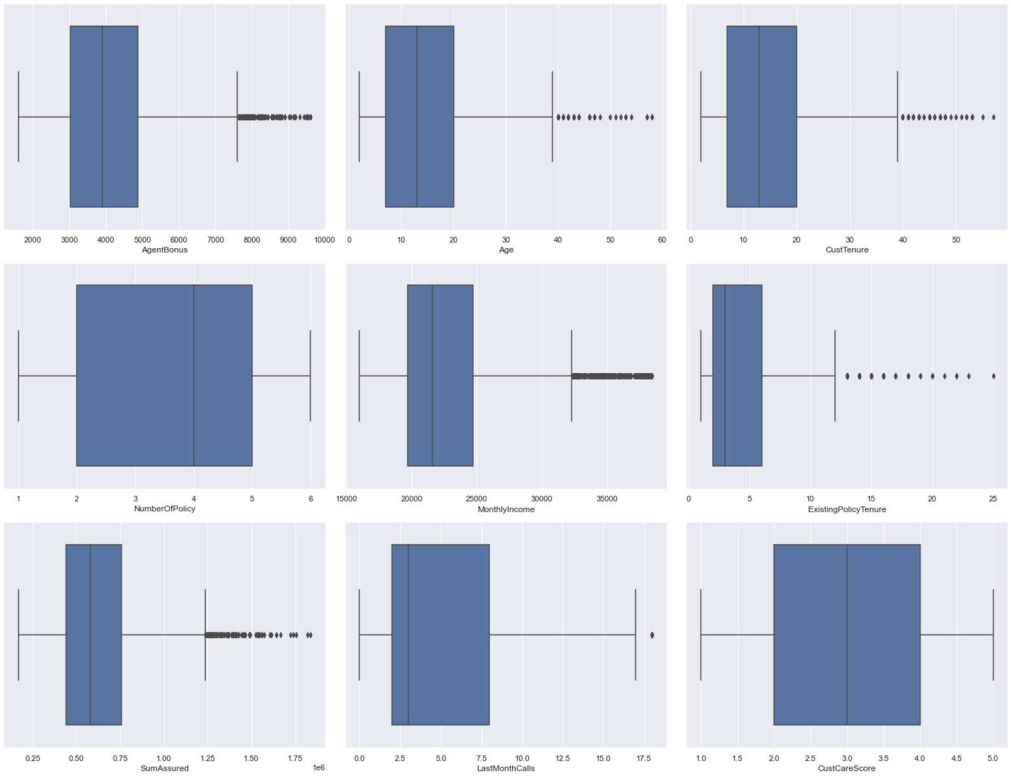


Fig.62

We have observed that all the variables have outliers

Shape after Outliers Treatment

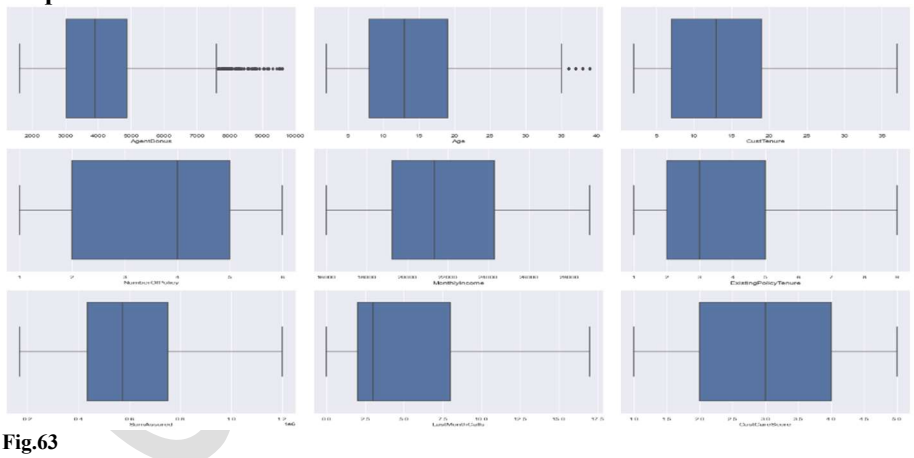


Fig.63

We have observed from **Fig.63** that all the outliers have been treated except the Dependent variable: **AgentBonus** since it will be used for Prediction and does not require any treatment

```
SumAssured              0
NumberOfPolicy          0
MonthlyIncome           0
LastMonthCalls          0
ExistingPolicyTenure    0
CustTenure              0
CustCareScore           0
AgentBonus              0
Age                     0
dtype: int64
```

Fig.64

We have observed from the above figure that all the Null values have been treated.

Correlation Plot

Correlation can tell **if two variables have a linear relationship, and the strength of that relationship**.

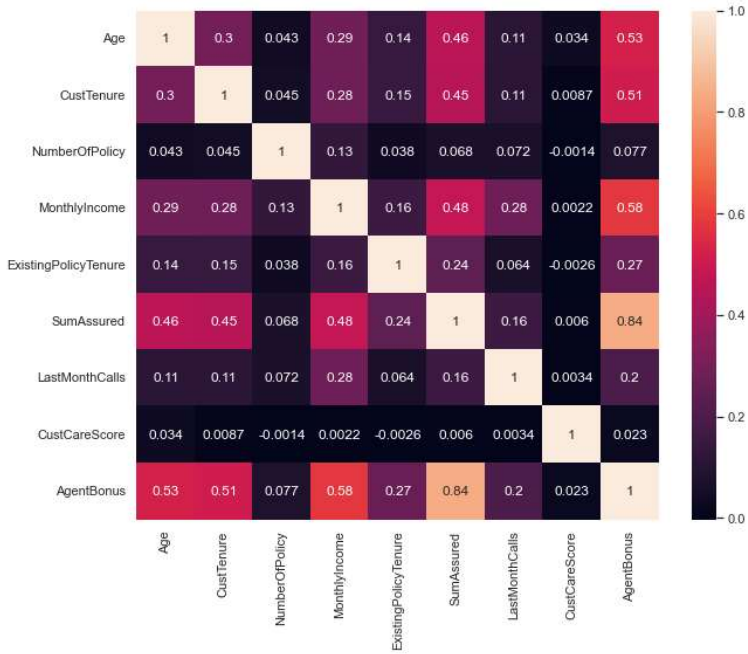


Fig.65

We are interested in knowing the relation of other variables with **dependent variable: AgentBonus**

Age, Cust Tenure and monthly income are the highly correlated variables with AgentBonus.

Age, Cust Tenure and monthly income are the highly correlated variables with Sum Assured.

Pair Plot of the Variables

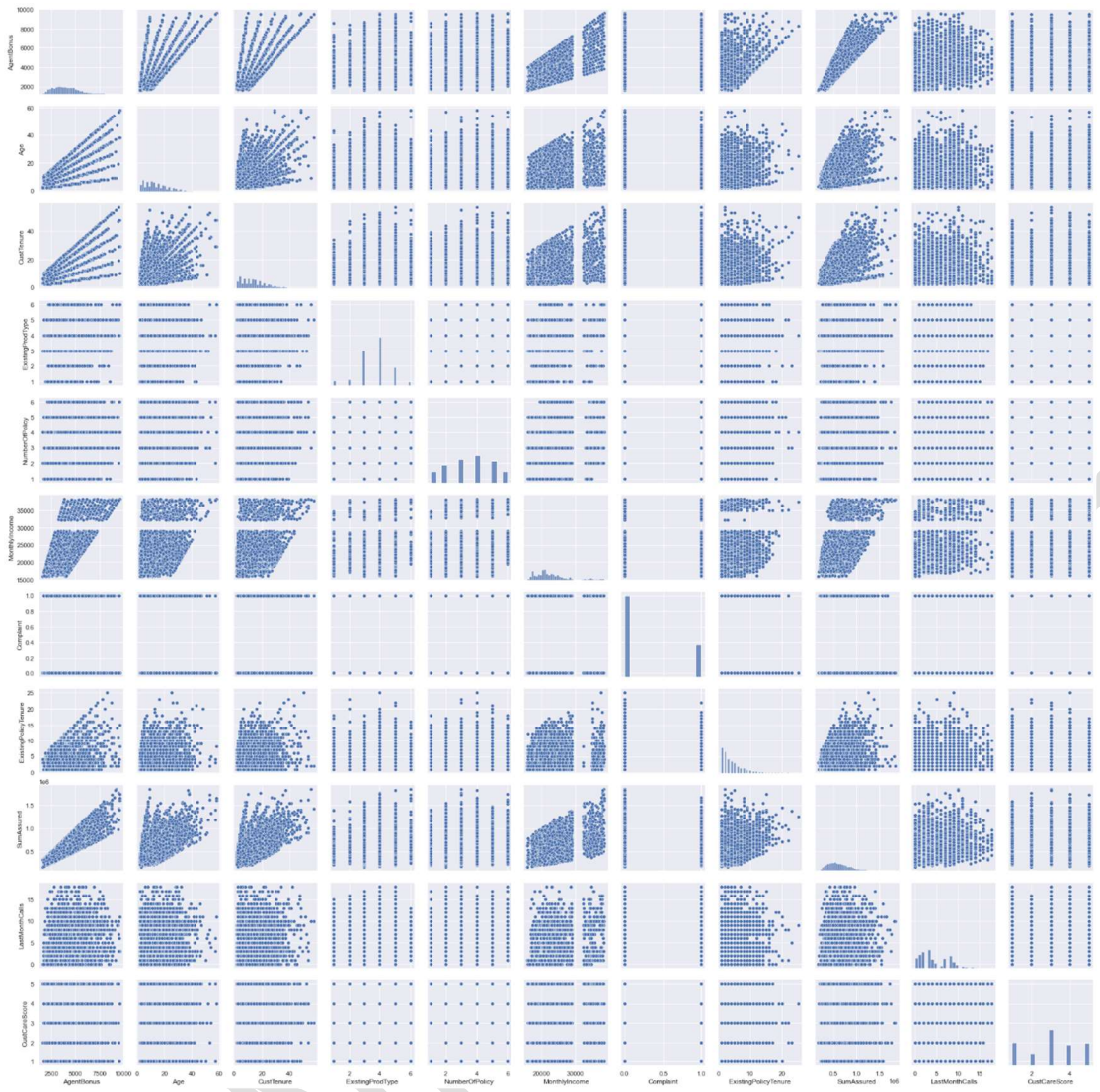


Fig.66

We have observed from **Fig.19** that the category in Occupation has no visible impact on the dependent **variable: AgentBonus** except **Free_Lancer** which has only 2 rows.

We have observed that the **dependent variable: Agent Bonus** is linearly**(positively correlated)** related to **Cust Tenure, Sum Assured,Age,Montly Income and Customer Tenure.**

Age is also linearly**(positively correlated)** related to **CustTenure** and **Sum Assured**

Data Engineering

Machine learning models require all input and output variables to be numeric.
This means that if your data contains categorical data, you must encode it to numbers before you can fit and evaluate a mode
Numerical data, as its name suggests, involves features that are only composed of numbers, such as integers or floating-point values.
Categorical data are variables that contain label values rather than numeric values.
The number of possible values is often limited to a fixed set.
Categorical variables are often called nominal.

- **Nominal Variable (Categorical).** Variable comprises a finite set of discrete values with no relationship between values.

'Channel', 'MaritalStatus', 'EducationField', 'Zone', 'PaymentMethod' are nominal variables in the dataset.

We have used One-Hot and Label Encoding for Nominal values.

We have also dropped the first column during One Hot Encoding to avoid Multiicollinearity.

One-Hot Encoding

For categorical variables where no ordinal relationship exists, the integer encoding may not be enough, at best, or misleading to the model at worst.

Ordinal Variable. Variable comprises a finite set of discrete values with a ranked ordering between values.

In ordinal encoding, each unique category value is assigned an integer value.
Occupation and Designation are ordinal variables in the dataset
In Occupation, we have used label encoding
We have given higher values for the category having higher order
Free_Lancer 0
Salaried 1
Small_Business 2
Large_Business 3

In Designation, we have ordered in the below sequence. Executive 0

Manager 1
Senior_Manager 2
AVP 3
VP 4

Fig.69

After Encoding, the data set has 29 columns with 4520 rows. The number of variables increased from 10 to 29.

The continuous variables in the data set are on different scale and would affect the model, so we have done the scaling on the dataset using **StandardScaler**

StandardScaler standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation.**StandardScaler** makes the mean of the distribution 0.

About 68% of the values will lie be between -1 and 1.

Scaled Dataset

executed in 453ms, finished 12:15:24 2021-10-03

	Age	CustTenure	NumberOfPolicy	MonthlyIncome	ExistingPolicyTenure	SumAssured	LastMonthCalls	CustCareScore	Occupation	Gender	ExistingI
0	0.96	-1.24	-1.08	-0.27	-0.68	0.92	0.11	-0.78	1	0	
1	-0.37	-1.49	0.30	-0.52	-0.22	-1.41	0.68	-0.05	1	1	
2	1.44	-1.24	-0.39	-1.40	-0.68	-0.02	-1.29	-0.05	0	1	
3	-0.37	-1.00	-0.39	-1.16	-0.68	-1.53	-1.29	1.40	1	0	
4	-0.98	-0.69	0.30	-1.00	0.24	-1.08	-0.73	1.40	2	1	
...	...	...	...	...	...	...	...	...	...	...	
4515	-1.22	-0.74	-1.08	1.28	-0.68	0.15	1.24	-1.50	2	1	
4516	-0.62	-0.62	-1.08	-0.27	-0.22	-1.40	-1.01	-0.05	1	0	
4517	1.08	1.13	0.99	-1.41	-0.68	0.29	-0.17	-1.50	1	0	
4518	-0.49	-0.49	-1.08	-0.54	1.17	1.55	-1.01	1.40	2	0	
4519	-0.01	-0.49	-1.08	0.55	-0.22	0.44	-1.01	-0.05	1	0	

Fig.70

Clustering

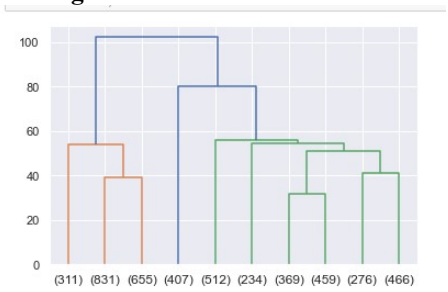
We have applied clustering on the data

Hierarchical Clustering

Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm **that groups similar objects into groups called clusters.**

.

Dendogram for last 10 values.



Cluster 1:

	Age	CustTenure	NumberOfPolicy	MonthlyIncome	ExistingPolicyTenure	SumAssured	LastMonthCalls	CustCareScore	AgentBonus	clusters
count	1797.00	1797.00	1797.00	1797.00	1797.00	1797.00	1797.00	1797.00	1797.00	1797.00
mean	10.71	10.55	3.43	20107.18	2.55	491297.63	3.75	3.31	3297.66	1.00
std	5.76	5.80	1.47	2317.98	1.60	167311.95	3.08	1.26	934.91	0.00
min	2.00	2.00	1.00	16009.00	1.00	168536.00	0.00	1.00	1605.00	1.00
25%	6.00	6.00	2.00	17886.00	1.00	363656.00	2.00	3.00	2531.00	1.00
50%	10.00	10.00	3.00	20159.00	2.00	467478.00	3.00	3.00	3191.00	1.00
75%	14.00	14.00	5.00	21430.00	4.00	595790.00	7.00	4.00	3958.00	1.00
max	35.00	32.00	6.00	28744.00	9.00	1184400.00	17.00	5.00	7468.00	1.00

Cluster 2:

	Age	CustTenure	NumberOfPolicy	MonthlyIncome	ExistingPolicyTenure	SumAssured	LastMonthCalls	CustCareScore	AgentBonus	clusters
count	407.00	407.00	407.00	407.00	407.00	407.00	407.00	407.00	407.00	407.00
mean	22.34	21.84	3.56	27625.55	4.56	925338.18	6.34	3.36	6723.50	2.00
std	9.78	8.64	1.36	1639.95	1.81	157287.50	4.44	1.30	1195.14	0.00
min	4.00	5.00	1.00	17130.20	1.00	432089.00	0.00	1.00	3324.00	2.00
25%	14.00	16.00	3.00	27226.70	3.70	817471.00	2.00	3.00	5874.00	2.00
50%	22.90	22.80	3.60	28403.10	4.50	947087.00	7.00	3.00	6648.00	2.00
75%	31.00	28.00	5.00	28621.10	5.45	1035245.50	10.00	4.00	7591.00	2.00
max	39.00	37.00	6.00	29009.00	9.00	1201184.00	17.00	5.00	9608.00	2.00

Cluster 2: High Earning customers having higher choice of Sum assured,Policy tenure leading to higher Agent Bonus
Cluster 3: Medium Earning customers having medium choice of Sum assured,Policy tenure leading to higher A gent Bonus

Cluster 1: Lowest Earning customers having medium choice of Sum assured,Policy tenure leading to higher Ag ent Bonus

3.1 Model Building

The dataset has been split to Training and Testing Set in 70:30 ratio.

Snapshot of Dataset after splitting:

Training Set:

The Training Set has 3164 rows and 19 columns. We have dropped **custid** column since it is insignificant in the model building

	Age	CustTenure	NumberOfPolicy	MonthlyIncome	ExistingPolicyTenure	SumAssured	LastMonthCalls	CustCareScore	AgentBonus	Channel
2461	12.00	16.00	3.00	20742.00	4.00	480800.00	1.00	4.00	3941.00	Third_Party_Partner
3681	31.00	15.00	5.00	23398.00	4.00	617707.00	2.00	4.00	5148.00	Agent
1309	15.00	6.00	1.00	16232.00	6.00	453360.00	0.00	5.00	3084.00	Agent
4254	5.00	16.00	2.00	23536.00	1.00	282197.00	8.00	3.00	2589.00	Online
1335	8.00	17.00	1.00	17269.00	6.00	750165.00	1.00	5.00	4145.00	Agent
...	...	...	...	...	...	...	...	...	...	...
2895	6.00	10.00	5.00	21658.00	8.00	610756.00	5.00	2.00	3249.00	Agent
2763	14.00	5.00	4.00	20976.00	7.00	316318.00	10.00	5.00	2727.00	Agent
905	12.00	12.00	1.00	19285.00	3.00	493696.00	7.00	1.00	3857.00	Agent
3980	5.00	5.00	5.00	17130.20	2.00	915352.00	9.00	3.00	4948.00	Agent
235	14.00	16.00	2.00	17097.00	2.00	415000.20	4.00	3.00	2736.00	Online

3164 rows × 19 columns

Fig.8

Testing Set:

The Training Set has 1356 rows and 19 columns. We have dropped **custid** column since it is insignificant in the model building

	Age	CustTenure	NumberOfPolicy	MonthlyIncome	ExistingPolicyTenure	SumAssured	LastMonthCalls	CustCareScore	AgentBonus	Channel
610	11.00	23.00	2.00	22756.00	1.00	785082.00	4.00	2.00	5689.00	Third_Party_Partner
1519	20.00	7.00	1.00	27782.60	4.80	1200156.00	6.00	1.00	6558.00	Agent
1620	7.00	15.00	1.00	18697.00	4.00	430218.00	8.00	3.00	2431.00	Agent
2031	13.00	22.00	4.00	21385.00	4.00	884697.00	7.00	5.00	4491.00	Third_Party_Partner
494	12.00	14.00	3.00	17259.00	1.00	253707.00	0.00	2.00	2416.00	Agent
...	...	...	...	...	...	...	...	...	...	...
2124	5.00	31.00	5.00	28758.00	2.00	553879.00	14.00	5.00	5176.00	Third_Party_Partner
3220	12.00	7.00	6.00	23789.00	4.00	421065.00	4.00	1.00	2379.00	Third_Party_Partner
1851	11.00	7.30	3.00	18505.00	1.00	410811.00	2.00	1.00	2221.00	Agent
1065	7.80	10.00	4.00	17760.00	1.00	390365.00	2.00	1.00	2486.00	Agent
462	9.00	18.00	2.00	26836.00	7.00	807495.00	2.00	3.00	4562.00	Agent

1356 rows × 19 columns

Fig.9

3.1.1 Multiple Linear Regression using statsmodels

Ordinary least squares Linear Regression.

LinearRegression fits a linear model with coefficients w = (w1, …, wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

Regression coefficients

The coefficient value represents the mean change in the response given a one unit change in the predictor. For example, if a coefficient is +3, the mean response value increases by 3 for every one unit change in the predictor.

If Age increases by 1 unit, the **dependent variables: AgentBonus** increases by 20 as per above model.

If sum Assured increases by 1 unit,the **dependent variables: AgentBonus** increases by 20 as per above model.

The column: P>|t| gives the value of probabilities of significance for the independent variable.

We have observed that the **Pvalue** of the **variables: Age,CustTenure,MonthlyIncome,ExistingPolicyTenure and SumAssured etc are 0** and it means that these variables are significant in model building.

In statistics, the **Jarque–Bera** test is a goodness-of-fit test of whether sample data have the skewness and kurtosis matching a normal distribution. The test is named after Carlos Jarque and Anil K.

Prob(JB): 1.58e-17

Inference from Jarque–Bera test

It means that the model is reliable.

Variance inflation factor (VIF) is a measure of the amount of multicollinearity in a set of multiple regression variables. Mathematically, the VIF for a regression model variable is equal to the ratio of the overall model variance to the variance of a model that includes only that single independent variable.

Values of VIF that exceed 5 are often regarded as indicating multicollinearity.

We have excluded the variables having VIF score more than 5 to avoid multicollinearity.

VIF Score of variables before removing

```
Age  VIF =  1.33
CustTenure  VIF =  1.31
NumberOfPolicy  VIF =  1.12
MonthlyIncome  VIF =  2.19
ExistingPolicyTenure  VIF =  1.08
SumAssured  VIF =  1.83
LastMonthCalls  VIF =  1.2
CustCareScore  VIF =  1.03
Channel_Online  VIF =  1.05
Channel_Third_Party_Partner  VIF =  1.04
MaritalStatus_Married  VIF =  1.92
MaritalStatus_Single  VIF =  1.93
EducationField_Engineer  VIF =  16.85
EducationField_Graduate  VIF =  17.75
EducationField_MBA  VIF =  2.01
EducationField_Post_Graduate  VIF =  4.46
EducationField_Under_Graduate  VIF =  2.71
Zone_North  VIF =  18.43
Zone_South  VIF =  1.12
Zone_West  VIF =  18.39
PaymentMethod_Monthly  VIF =  9.53
PaymentMethod_Quarterly  VIF =  1.42
PaymentMethod_Yearly  VIF =  2.37
Occupation_Large_Business  VIF =  143.6
Occupation_Salaried  VIF =  402.8
Occupation_Small_Business  VIF =  411.91
ExistingProdType_2  VIF =  2.23
ExistingProdType_3  VIF =  29.95
ExistingProdType_4  VIF =  35.66
ExistingProdType_5  VIF =  20.52
ExistingProdType_6  VIF =  4.75
Gender_Male  VIF =  1.03
Designation_Executive  VIF =  4.83
Designation_Manager  VIF =  3.98
Designation_Senior_Manager  VIF =  2.68
Designation_VP  VIF =  1.6
Complaint_1  VIF =  1.01
```

Fig.18

We have removed the varibales having VIF score more than 5.

'AgentBonus','EducationField_Engineer','Occupation_Small_Business','ExistingProdType_4','EducationField_Graduate','Zone_North','Zone_West','PaymentMethod_Monthly','Occupation_Salaried','Occupation_Small_Business','ExistingProdType_3','ExistingProdType_4','ExistingProdType_5'

AgentBonus is removed since it is a dependent variable.

```
No. of variables with VIF > 5 and therefore not used for modelling is  19
No. of variables with VIF < 5 and therefore  used for modelling is  18
```

```
Age  VIF =  1.32
CustTenure  VIF =  1.31
NumberOfPolicy  VIF =  1.05
MonthlyIncome  VIF =  1.9
ExistingPolicyTenure  VIF =  1.08
SumAssured  VIF =  1.82
LastMonthCalls  VIF =  1.2
CustCareScore  VIF =  1.02
Channel_Online  VIF =  1.05
Channel_Third_Party_Partner  VIF =  1.04
MaritalStatus_Married  VIF =  1.91
MaritalStatus_Single  VIF =  1.92
EducationField_MBA  VIF =  1.03
EducationField_Post_Graduate  VIF =  1.05
EducationField_Under_Graduate  VIF =  1.1
Zone_South  VIF =  1.01
PaymentMethod_Quarterly  VIF =  1.08
PaymentMethod_Yearly  VIF =  1.09
Occupation_Large_Business  VIF =  1.07
ExistingProdType_2  VIF =  1.09
ExistingProdType_6  VIF =  1.08
Gender_Male  VIF =  1.02
Designation_Executive  VIF =  4.63
Designation_Manager  VIF =  3.93
Designation_Senior_Manager  VIF =  2.66
Designation_VP  VIF =  1.6
Complaint_1  VIF =  1.01
```

Linear Regression using statsmodels with Encoded categorical variables and scaled continuous variables

We have built the OLS Regression model using only significant variable with scaled data.

OLS Regression Results

Dep. Variable:	AgentBonus	R-squared:	0.815
Model:	OLS	Adj. R-squared:	0.814
Method:	Least Squares	F-statistic:	1542.
Date:	Sat, 16 Oct 2021	Prob (F-statistic):	0.00
Time:	16:39:03	Log-Likelihood:	-1821.6
No. Observations:	3164	AIC:	3663.
Df Residuals:	3154	BIC:	3724.
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-8.674e-17	0.008	-1.13e-14	1.000	-0.015	0.015
Age	0.1205	0.009	13.712	0.000	0.103	0.138
CustTenure	0.1177	0.009	13.466	0.000	0.101	0.135
MonthlyIncome	0.2222	0.010	21.513	0.000	0.202	0.242
ExistingPolicyTenure	0.0522	0.008	6.585	0.000	0.037	0.068
SumAssured	0.5544	0.010	53.801	0.000	0.534	0.575
Designation_Executive	-0.1849	0.016	-11.578	0.000	-0.216	-0.154
Designation_Manager	-0.2401	0.015	-15.973	0.000	-0.270	-0.211
Designation_Senior_Manager	-0.1646	0.012	-13.262	0.000	-0.189	-0.140
Designation_VP	0.0540	0.010	5.630	0.000	0.035	0.073

Omnibus:	72.433	Durbin-Watson:	1.994
Prob(Omnibus):	0.000	Jarque-Bera (JB):	77.043
Skew:	0.370	Prob(JB):	1.86e-17
Kurtosis:	3.190	Cond. No.	5.11

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Fig.23

Plot of Predicted data with Actual Test data

 # Since this is regression, plot the predicted y value vs actual y values for the test data

 # A good model's prediction will be close to actual leading to high R and R2 values

This is a good model since model's prediction is close to actual leading to high R and R2 values

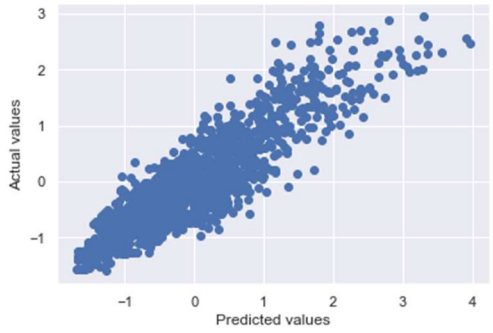


Fig.23

Regression Output:

```
(-0.0) * Intercept + (0.12) * Age + (0.12) * CustTenure + (0.22) * MonthlyIncome +
(0.05) * ExistingPolicyTenure + (0.55) * SumAssured + (-0.18) * Designation_Executi
ve + (-0.24) * Designation_Manager + (-0.16) * Designation_Senior_Manager + (0.05)
* Designation_VP +
```

Inference

When Age increases by 1 unit, AgentBonus increases by 0.12 units, keeping all other predictors constant.

similarly, when Monthly Income increases by 1 unit, AgentBonus increases by 0.22 units, keeping all other predictors constant.

There are also some negative co-efficient values, for instance, **Designation_Executive** has its corresponding co-efficient as `-0.18`. This implies, when Designation is Executive, the **AgentBonus** decreases by 0.18 units, keeping all other predictors constant.

Models	Train RMSE	Test RMSE	Training Score	Test Score

OLS Regression Model(Without Scaling/Encoding)	601.51	625.23		
Linear Regression Model(Without Scaling/With Encoding)	601.51	625.23	0.81	0.79
OLS Regression Model(Without Scaling/With Encoding)	601.51	625.23		
OLS Regression Model(after treating multicollinearity)	604.39	622.48		
Linear Regression Model(With Scaling/With Encoding)	**0.427**	**0.447**	**0.81**	**0.80**
OLS Regression Model(No multicollinearity/with scaling)	0.430	0.444		

We do not see much improvement in the RMSE scores

Train/Test Score(R Square): 81% of the variation in the AgentBonus is explained by the predictors in the model for train set and 80% for Test Set.

The Linear Regression model is easy to interpret by the users since the coefficients of the Linear Regression output model

XGBRegressor Model:

We have used Grid Search with below parameters to tune the model with multiple values.

The **Grid Search** method is a basic tool for hyperparameter optimization. The Grid Search Method considers several hyperparameter combinations and chooses the one that returns a lower error score.

We have built a base model without tuning.

Model Score/Validation:

Models	Train RMSE	Test RMSE	Training Score	Test Score
XGBRegressor Base Model(Unscaled Data)	**141.65**	**498.43**	**0.98**	**0.87**
XGBRegressor Base Model(Scaled Data)	**141.65**	**498.61**	**0.98**	**0.87**

Fig.20

Feature Importance Ranking:

```
Feature importance ranking


1.SumAssured(0.645731)
2.MonthlyIncome(0.196975)
3.Age(0.051108)
4.CustTenure(0.043030)
5.ExistingPolicyTenure(0.009742)
6.LastMonthCalls(0.007753)
7.Designation_VP(0.007199)
8.NumberOfPolicy(0.005554)
9.CustCareScore(0.004390)
10.Designation_Senior_Manager(0.003897)
11.Gender_Male(0.001549)
12.MaritalStatus_Married(0.001532)
13.ExistingProdType_4(0.001507)
14.PaymentMethod_Yearly(0.001433)
15.Channel_Third_Party_Partner(0.001390)
16.Complaint_1(0.001311)
17.Channel_Online(0.001290)
18.Designation_Manager(0.001284)
19.EducationField_Under_Graduate(0.001098)
20.MaritalStatus_Single(0.001093)
21.ExistingProdType_3(0.001083)
22.ExistingProdType_5(0.001019)
23.Zone_West(0.000989)
24.Zone_North(0.000890)
25.EducationField_Graduate(0.000881)
26.Occupation_Small_Business(0.000860)
27.Designation_Executive(0.000849)
28.Occupation_Salaried(0.000805)
29.EducationField_Post_Graduate(0.000637)
30.EducationField_Engineer(0.000570)
31.ExistingProdType_2(0.000542)
32.PaymentMethod_Monthly(0.000536)
33.Occupation_Large_Business(0.000522)
34.ExistingProdType_6(0.000505)
35.PaymentMethod_Quarterly(0.000280)
36.EducationField_MBA(0.000120)
37.Zone_South(0.000045)
```

Model Validation:

We have observed that RMSE for Training set has reduced drastically.

We have observed that RMSE for Testing set has reduced drastically.

The performance of the model on the training dataset is significantly better than the performance on the test dataset, hence it is overfitting in both the case for the model

Model with Scaled Data:

```
{'colsample_bytree': 0.8, 'learning_rate': 0.03, 'max_depth': 7, 'n_estimators': 200, 'subsample': 0.8}
```

Model Scores:

Models	Train RMSE	Test RMSE	Training Score	Test Score
OLS Regression Model(Without Scaling/Encoding)	601.51	625.23		
Linear Regression Model(Without Scaling/With Encoding)	601.51	625.23	0.81	0.79
OLS Regression Model(Without Scaling/With Encoding)	601.51	625.23		
OLS Regression Model(after treating multicollinearity)	604.39	622.48		
Linear Regression Model(With Scaling/With Encoding)	0.427	0.447	0.81	0.8
OLS Regression Model(No multicollinearity/with scaling)	0.43	0.444		
XGBRegressor with Grid Search(Unscaled Data)	**236.19**	**474.2**	**0.97**	**0.88**
XGBRegressor with Grid Search(scaled Data)	**235.82**	**474.72**	**0.95**	**0.88**

We have observed that the Training Score for XGBRegressor with Grid Search(Unscaled Data) increased by 0.02 but the Testing score remained same.

The performance of the model on the training dataset is significantly better than the performance on the test dataset,hence it is overfitting in both the case for the model

Random Forest Model

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression.. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees.

We have used Grid Search with below parameters to tune the model with multiple values.

Random Forest with Grid Search

Random Forest on Scaled Data

The best parameter chosen by the Grid search is as follows.

```
{'max_depth': 8, 'max_features': 7, 'min_samples_leaf': 30, 'min_sample
s_split': 90, 'n_estimators': 200}
```

Model Scores:

Models	Train RMSE	Test RMSE	Training Score	Test Score
Random forest(Unscaled Data)	**614.94**	**649.81**	**0.80**	**0.78**
Random forest(Scaled Data)	**609.64**	**645.27**	**0.81**	**0.78**

Inference:

The model built by Random Forest has a lower score but it is neither overfitting nor underfitting.

The Training score is marginally higher as compared to the Testing Score. It is a good model.RMSE is higher for the model.

ANN Model with Grid Search

We have used Grid Search with below parameters to tune the model with multiple values.

The **Grid Search** method is a basic tool for hyperparameter optimization. The Grid Search Method considers several hyperparameter combinations and chooses the one that returns a lower error score.

Best Parameter chosen by grid Search. `{'activation': 'relu', 'hidden_layer_sizes': 324, 'max_iter': 1000, 'solver': 'adam', 'tol': 0.01}`

Model Score

Models	Train RMSE	Test RMSE	Training Score	Test Score
ANN Model(Scaled Data)	**478.49**	**610.97**	**0.88**	**0.80**

Inference:

We have observed that the Training score by ANN Model is higher than the Testing set, so the model is overfitting

Model Comparison:

Models	Train RMSE	Test RMSE	Training Score	Test Score
OLS Regression Model(Without Scaling/Encoding)	601.51	625.23		
Linear Regression Model(Without Scaling/With Encoding)	601.51	625.23	0.81	0.79
OLS Regression Model(Without Scaling/With Encoding)	601.51	625.23		
OLS Regression Model(after treating multicollinearity)	604.39	622.48		
Linear Regression Model(With Scaling/With Encoding)	0.427	0.447	0.81	0.80
OLS Regression Model(No multicollinearity/with scaling)	0.43	0.444		
XGBRegressor Base Model(Unscaled Data)	**141.65**	**498.43**	**0.98**	**0.87**
XGBRegressor Base Model(Scaled Data)	**141.65**	**498.61**	**0.98**	**0.87**
XGBRegressor with Grid Search(Unscaled Data)	**236.19**	**474.2**	**0.97**	**0.88**
XGBRegressor with Grid Search(scaled Data)	**235.82**	**474.72**	**0.95**	**0.88**
Random forest(Scaled Data without Grid Search)	173.76	484.91	0.98	0.87
Random forest(Unscaled Data with Grid Search)	614.94	649.81	0.80	0.78
Random forest(Scaled Data)	609.64	645.27	0.81	0.78
ANN Model(Scaled Data without Grid Search)	1063.27	1090.87	0.42	0.39
ANN Model(Scaled Data with Grid Search)	478.49	610.97	0.88	0.80
DecisionTreeRegressor(Unscaled Data)		709.24	1	0.74
DecisionTreeRegressor(Unscaled Data with grid Search)			0.86	0.82
DecisionTreeRegressor(scaled Data with grid Search)			0.86	0.82

Inference:

We have observed that out of all the models **Linear Regression and XGBRegressor has low RMSE value** which means that the observed data points are close to the model's predicted values.

XGBRegressor Base Model has outperformed as compared to other models.

We have observed that RMSE is lowest among the model with higher Training/Testing score.

The model is over-fitting since training score is higher but it is with in allowed limit.

The difference in score is 0.11

XGBRegressor with Grid search has also performed well.

Train/Test Score(R Square): 98% of the variation in the AgentBonus is explained by the predictors in the model for train set and 87% for Test Set.

Linear Regression model has also lower RMSE values and model score of 0.80

End user can understand linear Regression model since it provides the coefficient values which helps in understanding the effect of Independent variables on the **dependent variable:AgentBonus**.

Implication on the business

The above model would help 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.

Recommendations on Dataset

The below recommendations can help in increasing Agent Bonus which would in turn increase the sales of the company.

- Agent should focus more on customers with **Designation as VP/AVP** to gain higher bonus.
- Agents should **sell a greater number of policies** to customers to raise the bonus amount.
- Agents need to get reliable customers who can stay in the policy for a **longer duration**. This can be done by having through verification of customers like customer's past performance, Bank statement etc.
- Agents need to explore **more on South and East zone** since it may help in getting more business opportunity. The company should arrange for campaign in these zones to increase the awareness on importance of having Insurance.
- Since Age is positively correlated with Agent Bonus
- , so Agents should focus on more **Aged customers**.
- Agents should focus on customers having **higher Income**.
- Agents should sell policies of **higher Sum Assured** which can raise the Bonus Amount.
- Agents should acquire **more Freelancer/Large Business** customers since they are in unorganized sector and may need Insurance policies. The company can customize the Insurance plan according to this sector. Like reducing the Sum Assured/Premium amount which can be affordable.
- It has been observed that the **Agent Bonus increases with higher Existing Policy tenure** of the customer.

Recommendations for Under performing Agents

- We have set the benchmark that company has classified Agents as performers whose Agent Bonus value is in Top 50 Percentile and upskill who falls under lower **50 percentile (Agent Bonus<3911).**
- The company should try to encourage Agents to explore more of **South/East zone** by providing **higher rewards**. This can increase Sales productivity of the company.
- Agents should try **to convince Freelancer/Large Business** since they are in unorganized sector. If the Agent can convince large business, then they may sell policies in bulk for all of its employees.
- Agent should try to **sell policies to VP/AVP** and **higher Income groups.**
- Agents needs to target more aged people since it may help in enhancing Agent bonus.
- Agents should have communication/follow ups with its old customers through call/mail since it has been observed that **Customer's tenure yields higher Bonus.** The company can also take the feedback of the existing customers to design new products, so that the existing customers can be retained and Agent Bonus could be maximised.