Life Insurance Company

Agent Performance

Business Report

Prepared by :

Sourav Pandey
1928265

Contents

1 Introduction to Business Problem

1.1 Defining the problem statement

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.

1.2 Need Of the Study/Project

As we know the company wants to get a knowledge about the performance of it's agent, so that effective upskill programme can be arranged for low performing agents and various engagement activities for high performers. This study will therefore help the low Performers to enhance their skills and inturn helps the company have more efficient agents. Similarly the study will help the high performers to know about where they can improve more. Overall this study is very essential for the company to boost it's performence.

1.3 Understanding Business Opurtunity

i) Company can understand the market perspective.
ii) Company will identify the high value agents and low value agents
iii) Company will accordingly plan the upskill program and also able to give reward to high value agents
iv)company will understand who will be the target value customer

2: Data Report

2.1 Understanding how data was collected in terms of time, frequency and methodology

The dataset is devided quite evenly among high performers and low performers but the zone wise representation is
High for west and north zone while compared to others.

Variable	Discerption
CustID	Unique customer ID
AgentBonus	Bonus amount given to each agents in last month
Age	Age of customer
CustTenure	Tenure of customer in organization
Channel	Channel through which acquisition of customer is done
Occupation	Occupation of customer
EducationField	Field of education of customer
Gender	Gender of customer
ExistingProdType	Existing product type of customer
Designation	Designation of customer in their organization
NumberOfPolicy	Total number of existing policy of a customer
MaritalStatus	Marital status of customer
MonthlyIncome	Gross monthly income of customer
Complaint	Indicator of complaint registered in last one month by customer
ExistingPolicyTenure	Max tenure in all existing policies of customer
SumAssured	Max of sum assured in all existing policies of customer

Zone	Customer belongs to which zone in India. Like East, West, North and South
PaymentMethod	Frequency of payment selected by customer like Monthly, quarterly, half yearly and yearly
LastMonthCalls	Total calls attempted by company to a customer for cross sell
CustCareScore	Customer satisfaction score given by customer in previous service call

2.2 Visual inspection of data (rows, columns, descriptive details)

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

dtypes: float64(7), int64(5), object(8)
memory usage: 706.4+ KB
```

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
AgentBonus	4520.00	NaN	NaN	NaN	4077.84	1403.32	1605.00	3027.75	3911.50	4867.25	9608.00
Age	4520.00	NaN	NaN	NaN	14.41	8.77	2.00	8.00	13.00	19.00	58.00
CustTenure	4520.00	NaN	NaN	NaN	14.40	8.74	2.00	8.00	13.00	19.00	57.00
Channel	4520	3	Agent	3194	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Occupation	4520	4	Salaried	2192	NaN	NaN	NaN	NaN	NaN	NaN	NaN
EducationField	4520	6	Graduate	1870	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	4520	2	Male	2688	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ExistingProdType	4520.00	NaN	NaN	NaN	3.69	1.02	1.00	3.00	4.00	4.00	6.00
Designation	4520	5	Executive	1662	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NumberOfPolicy	4520.00	NaN	NaN	NaN	3.57	1.45	1.00	2.00	4.00	5.00	6.00
MaritalStatus	4520	4	Married	2268	NaN	NaN	NaN	NaN	NaN	NaN	NaN
MonthlyIncome	4520.00	NaN	NaN	NaN	22823.25	4764.89	16009.00	19858.00	21606.00	24531.75	38456.00
Complaint	4520.00	2.00	0.00	3222.00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ExistingPolicyTenure	4520.00	NaN	NaN	NaN	4.08	3.29	1.00	2.00	3.00	5.00	25.00
SumAssured	4520.00	NaN	NaN	NaN	618602.01	242117.25	168536.00	444476.25	578976.50	750010.50	1838496.00
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	4520.00	NaN	NaN	NaN	4.63	3.62	0.00	2.00	3.00	8.00	18.00
CustCareScore	4520.00	NaN	NaN	NaN	3.07	1.38	1.00	2.00	3.00	4.00	5.00
Result	4520	2	low	2474	NaN	NaN	NaN	NaN	NaN	NaN	NaN

2.3 Understanding of attributes (variable info, renaming if required)

#	**Column**	**Count**	**Dtype**	**Remark**
0	*CustID*	*4520*	*int64*	*Dropped as not important.*
1	*AgentBonus*	*4520*	*int64*	*Numeric, target variable*
2	*Age*	*4251*	*float64*	*Numeric*
3	*CustTenure*	*4294*	*float64*	*Numeric*
4	*Channel*	*4520*	*object*	*Categorical*
5	*Occupation*	*4520*	*object*	*Categorical*
6	*EducationField*	*4520*	*object*	*Categorical*
7	*Gender*	*4520*	*object*	*Categorical*
8	*ExistingProdType*	*4520*	*int64*	*Numeric*
9	*Designation*	*4520*	*object*	*Categorical*
10	*NumberOfPolicy*	*4475*	*float64*	*Numeric*
11	*MaritalStatus*	*4520*	*object*	*Categorical*
12	*MonthlyIncome*	*4284*	*float64*	*Numeric*
13	*Complaint*	*4520*	*int64*	*Converted into categorical*
14	*ExistingPolicyTenure*	*4336*	*float64*	*Numeric*
15	*SumAssured*	*4366*	*float64*	*Numeric*
16	*Zone*	*4520*	*object*	*Categorical*
17	*PaymentMethod*	*4520*	*object*	*Categorical*
18	*LastMonthCalls*	*4520*	*int64*	*Numeric*
19	*CustCareScore*	*4468*	*float64*	*Numeric*

Dropped Column CustID.

```
df.drop(['CustID'],axis=1,inplace=True)
```

The name of the columns seems to be fine with no special characters or spaces between them .

Unique values of various Categories

```
Channel :  3
Online                 468
Third Party Partner     858
Agent                 3194
Name: Channel, dtype: int64


Occupation :  5
Free Lancer          2
Laarge Business     153
Large Business      255
Small Business     1918
Salaried          2192
Name: Occupation, dtype: int64


EducationField :  7
MBA                 74
UG                 230
Post Graduate      252
Engineer           408
Diploma            496
Under Graduate    1190
Graduate          1870
Name: EducationField, dtype: int64


Gender :  3
Fe male     325
Female     1507
Male       2688
Name: Gender, dtype: int64


Designation :  6
Exe                127
VP                 226
AVP                336
Senior Manager     676
Executive         1535
Manager           1620
Name: Designation, dtype: int64


MaritalStatus :  4
Unmarried     194
Divorced      804
Single       1254
Married      2268
Name: MaritalStatus, dtype: int64


Zone :  4
South      6
East      64
North    1884
West     2566
Name: Zone, dtype: int64


PaymentMethod :  4
Quarterly       76
Monthly        354
Yearly        1434
Half Yearly   2656
Name: PaymentMethod, dtype: int64
```

The highlighted data seems to be recorded incorrectly and required replacement and this was done to ensure theright categories are picked up by the model

Post fixing of the data

```
Channel :  3
Online               468
Third Party Partner     858
Agent               3194
Name: Channel, dtype: int64
```

```
Occupation :  4
Free Lancer         2
Large Business     408
Small Business    1918
Salaried         2192
Name: Occupation, dtype: int64
```

```
EducationField :  6
MBA              74
Post Graduate      252
Engineer         408
Diploma          496
Under Graduate    1420
Graduate         1870
Name: EducationField, dtype: int64
```

```
Gender :  2
Female    1832
Male      2688
Name: Gender, dtype: int64
```

```
Designation :  5
VP             226
AVP             336
Senior Manager     676
Manager          1620
Executive        1662
Name: Designation, dtype: int64
```

```
MaritalStatus :  4
Unmarried     194
Divorced      804
Single       1254
Married      2268
Name: MaritalStatus, dtype: int64
```

```
Complaint :  2
1    1298
0    3222
Name: Complaint, dtype: int64
```

```
Zone :  4
South      6
```

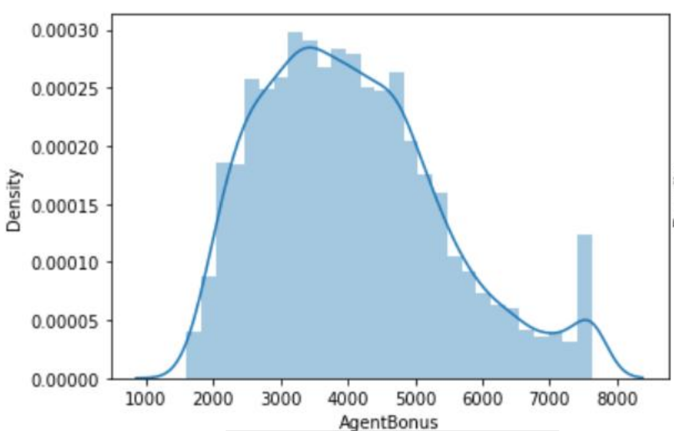
East 64
North 1884
West 2566
Name: Zone, dtype: int64

PaymentMethod : 4
Quarterly 76
Monthly 354
Yearly 1434
Half Yearly 2656
Name: PaymentMethod, dtype: int64

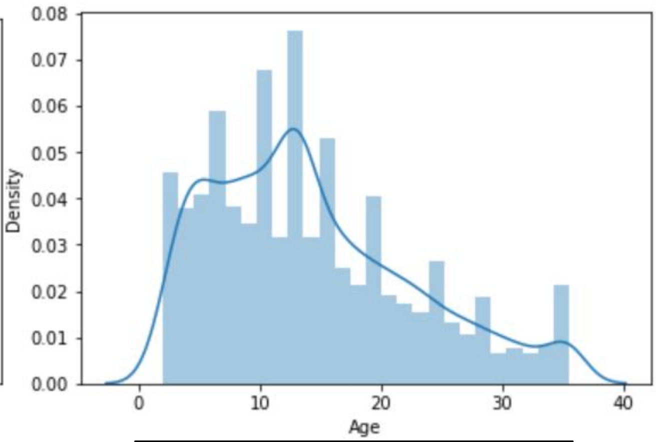
The complaint column was actually categorical columns but perceived as numarical because of incorrect data capture .. Fixing the inconsistencies fixed the type of the variable as well.

3. Expliratory Data Analysis

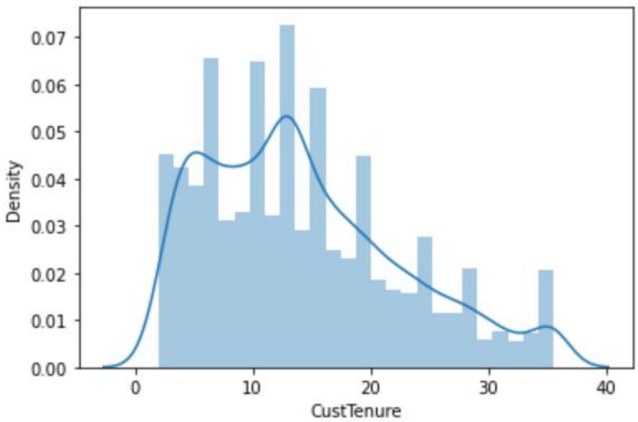
3.1 Univariate analysis (distribution and spread for every continuous attribute, distribution ofdata in categories for categorical ones)



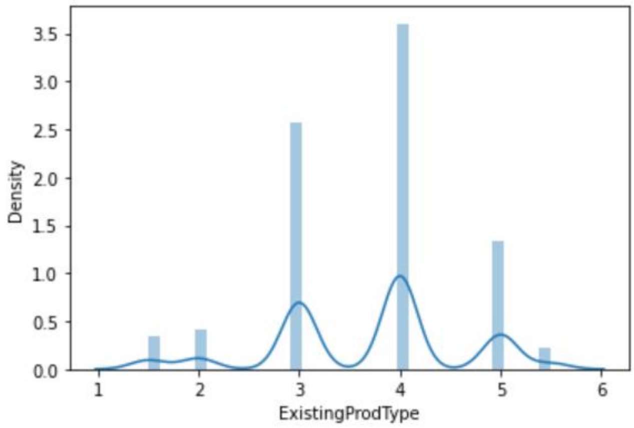
Continuous in a range



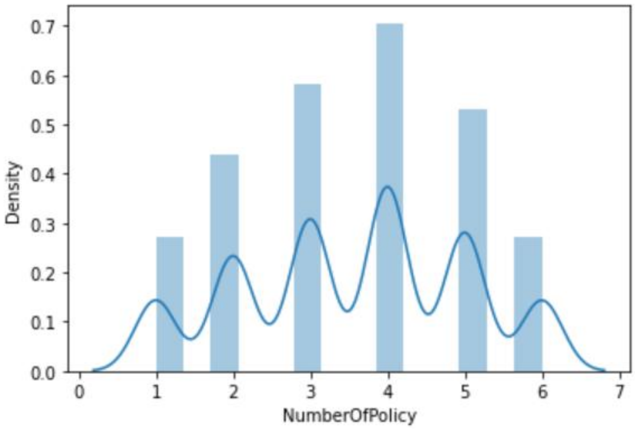
Continuous and Right Skewed



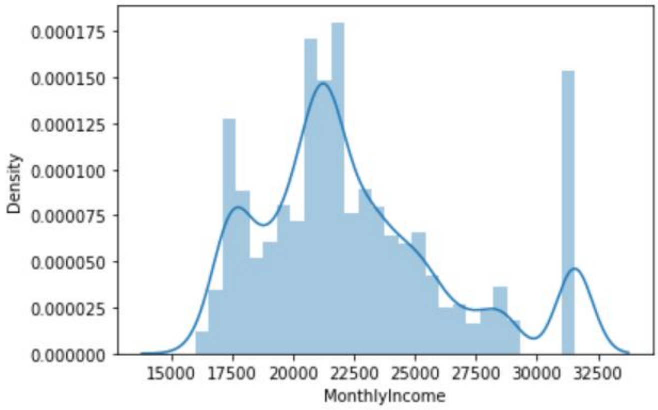
Right Skewed



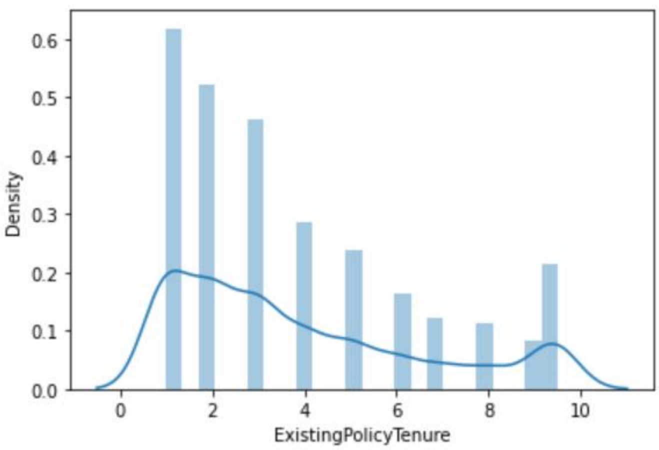
Discrete values with 4 most frequent



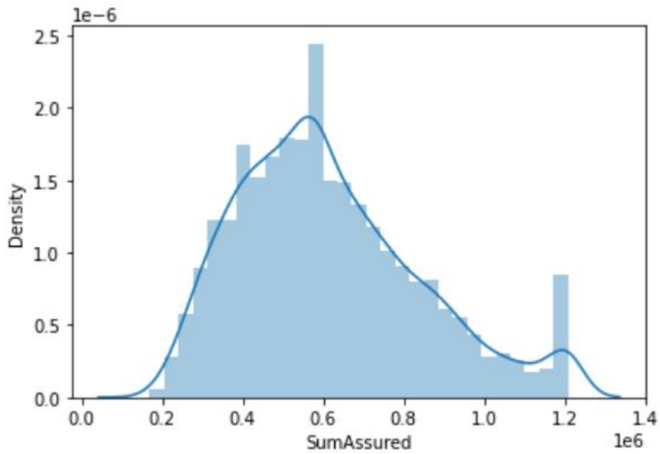
Discrete values with 4 most frequent



Continuous with some peaks

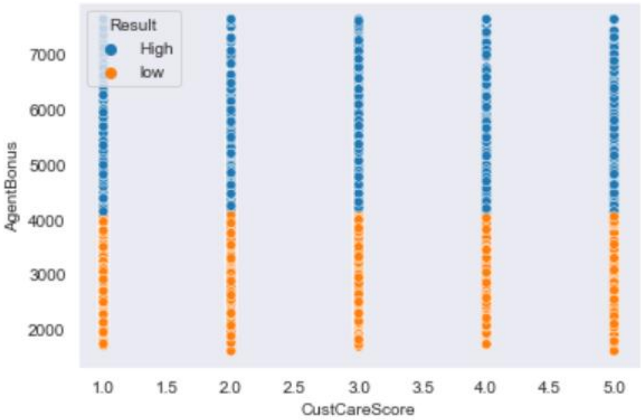


Discrete with 0-2 range highest

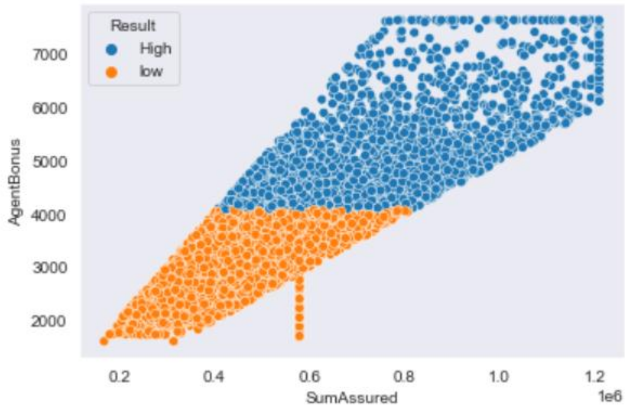


Continuous

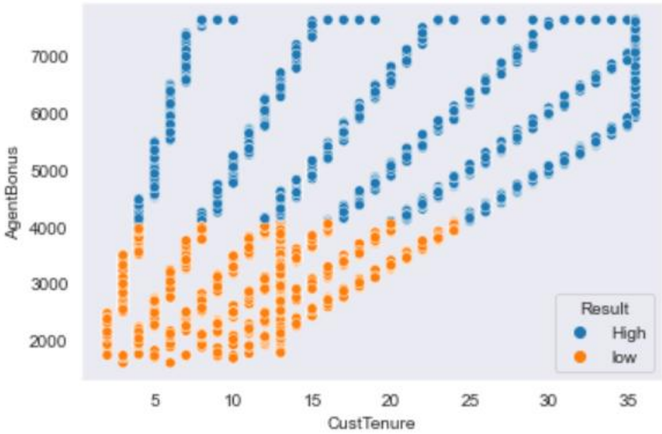
3.2 Bivariate analysis (relationship between different variables , correlations)



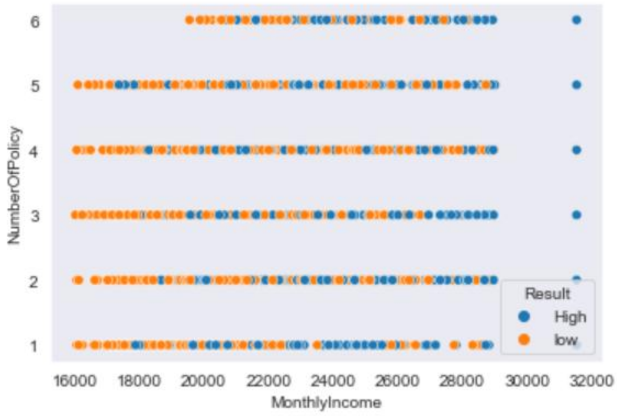
No relation found



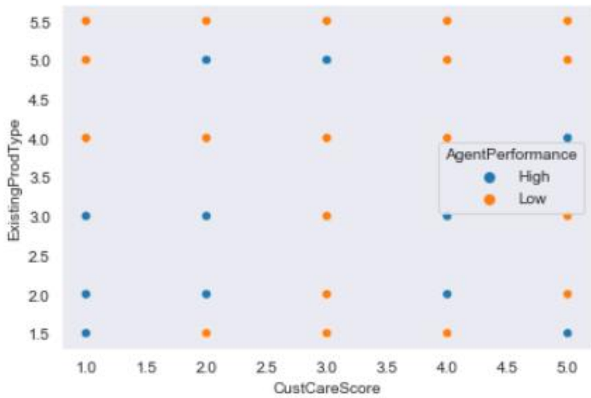
Positive Correlation



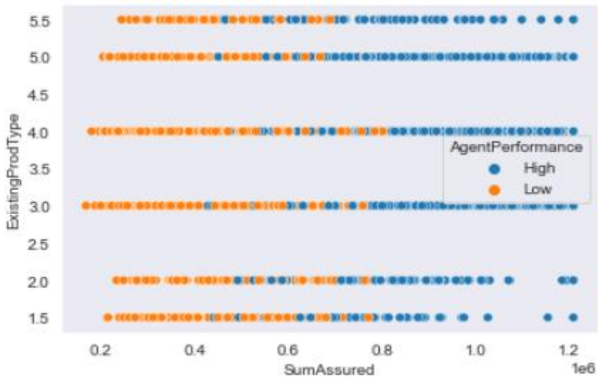
Positive Correlation



No relation found.



No relation found

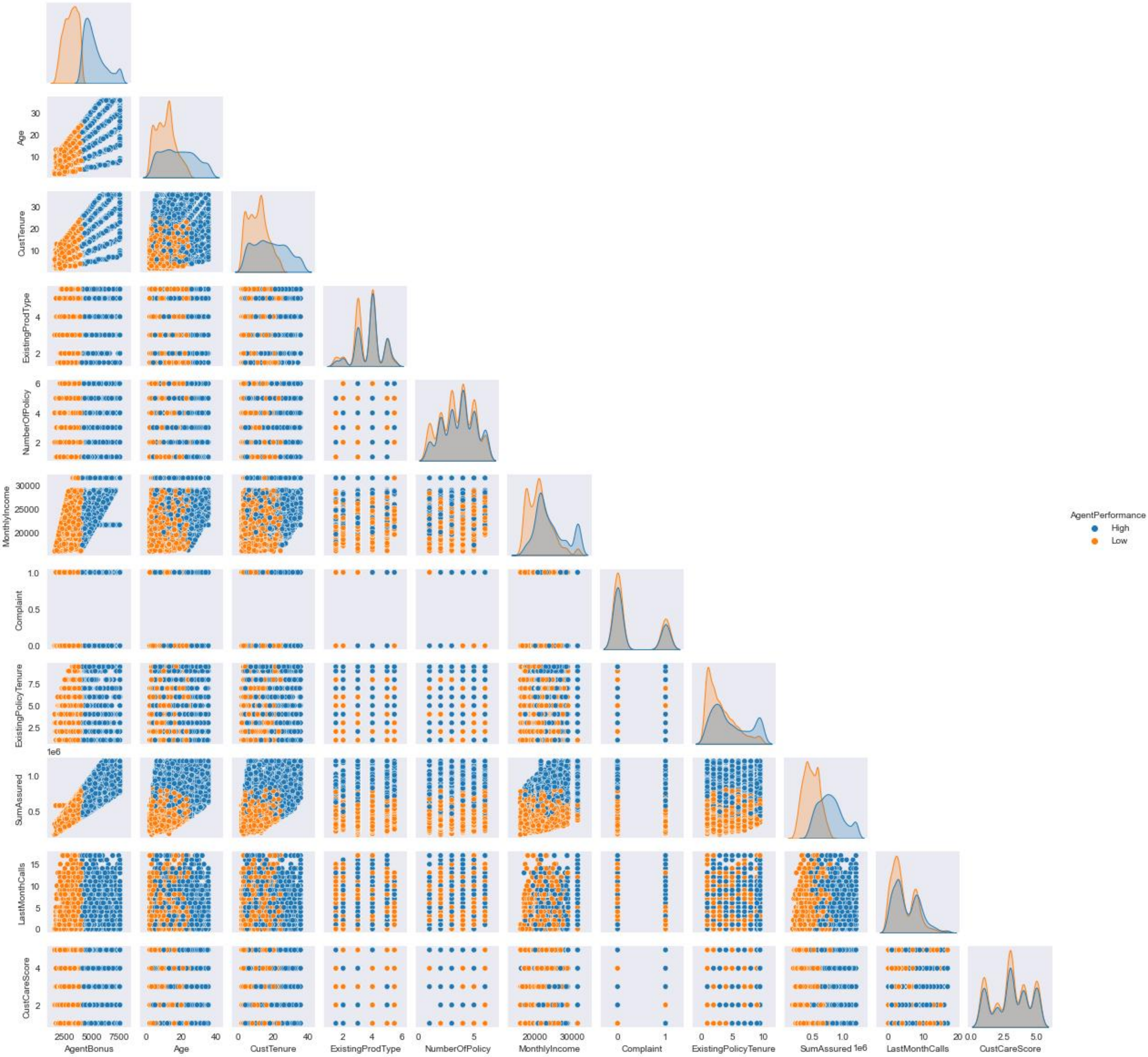


No relation found

Most of the variables don't seem to be related closely to each other which means there is low multi -collinearity in the data and each feature would have its importance in building the right model . because of this we have not dropped any columns other than CustId and would want to build the model to see the variable importance.

The pair plot also seems to suggest the same thing . But due to the huge number of columns pair plot was not providing very clear insight and hence resorted to bi variate plots with every combination possible.

Pair Plot



AgentPerformance
High
Low
AgentBonus
Age
CustTenure
ExistingProdType
NumberOfPolicy
MonthlyIncome
Complaint
ExistingPolicyTenure
SumAssured
LastMonthCalls
CustCareScore

3.3 Removal of unwanted variables (if applicable)

CustID is a redundant column and has been removed. Chose not to remove any other columns and left to the model phase where the variable importance would be judged.

```
In [7]:  df.drop(['CustID'],axis=1,inplace=True)
```

3.4 Missing Value treatment (if applicable)

```
In [19]:  print ('There are', df.isnull().sum().sum(),'missing values in the dataset')

There are 1166 missing values in the dataset
```

```
In [20]:  df.isnull().sum()[df.isnull().sum()>0]

Out[20]: Age                     269
         CustTenure              226
         NumberOfPolicy           45
         MonthlyIncome           236
         ExistingPolicyTenure    184
         SumAssured              154
         CustCareScore            52
         dtype: int64
```

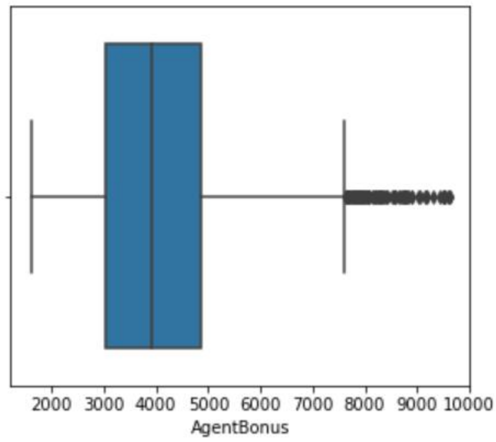
```
In [23]:  median1=df["Age"].median()
          median2=df["CustTenure"].median()
          median3=df["NumberOfPolicy"].median()
          median4=df["MonthlyIncome"].median()
          median5=df["ExistingPolicyTenure"].median()
          median6=df["SumAssured"].median()
          median7=df["CustCareScore"].median()


          df["Age"].replace(np.nan,median1,inplace=True)
          df["CustTenure"].replace(np.nan,median2,inplace=True)
          df["NumberOfPolicy"].replace(np.nan,median3,inplace=True)
          df["MonthlyIncome"].replace(np.nan,median4,inplace=True)
          df["ExistingPolicyTenure"].replace(np.nan,median5,inplace=True)
          df["SumAssured"].replace(np.nan,median6,inplace=True)
          df["CustCareScore"].replace(np.nan,median7,inplace=True)
```

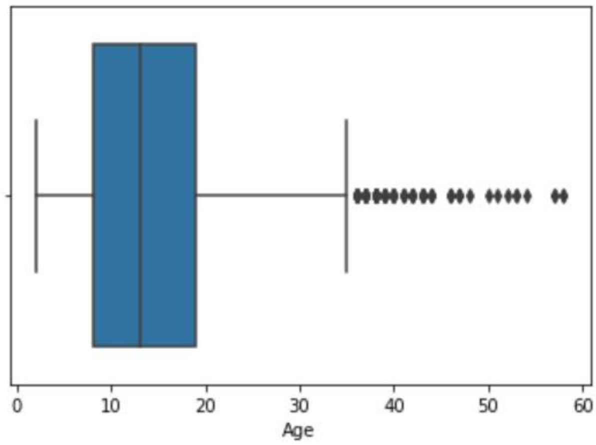
Since all the variables which had missing values were of numeric type .
So we have replaced it with median values.

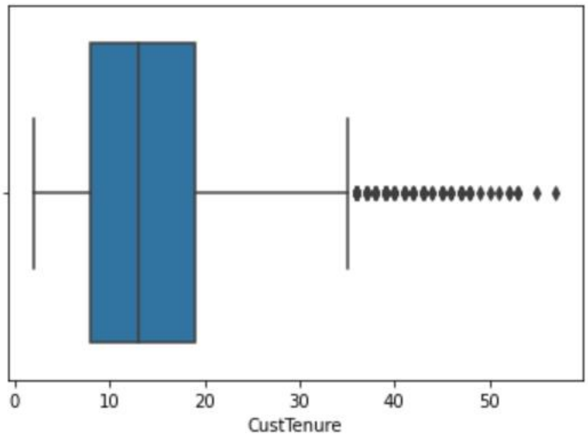
After fixing values:

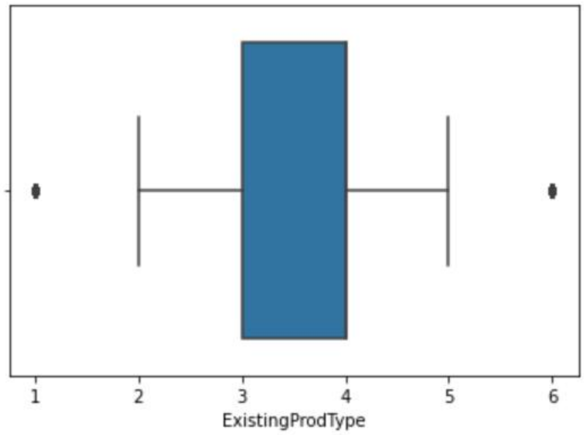
```
In [24]:  df.isnull().sum()

Out[24]:  AgentBonus              0
          Age                     0
          CustTenure              0
          Channel                 0
          Occupation              0
          EducationField          0
          Gender                  0
          ExistingProdType        0
          Designation             0
          NumberOfPolicy          0
          MaritalStatus           0
          MonthlyIncome           0
          Complaint               0
          ExistingPolicyTenure    0
          SumAssured              0
          Zone                    0
          PaymentMethod           0
          LastMonthCalls          0
          CustCareScore           0
          dtype: int64
```

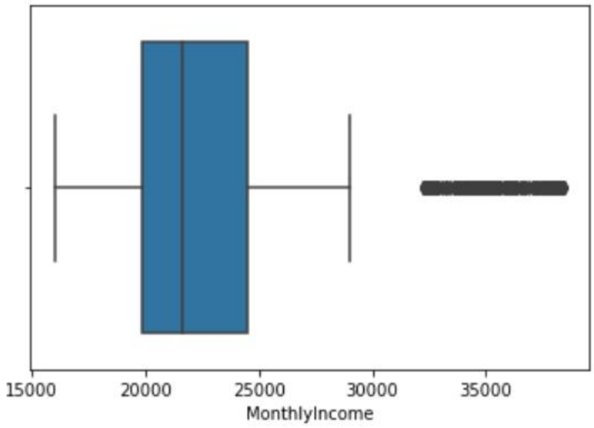
3.5 Outlier treatment (if required)

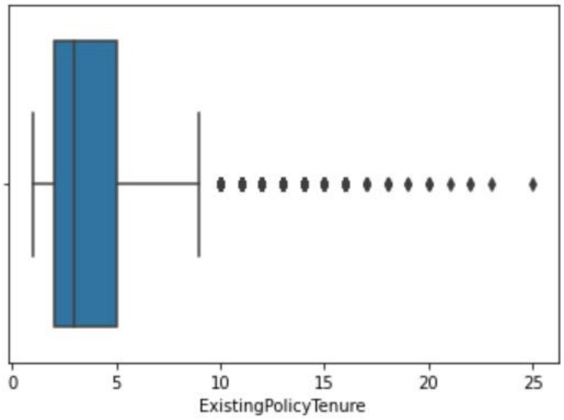


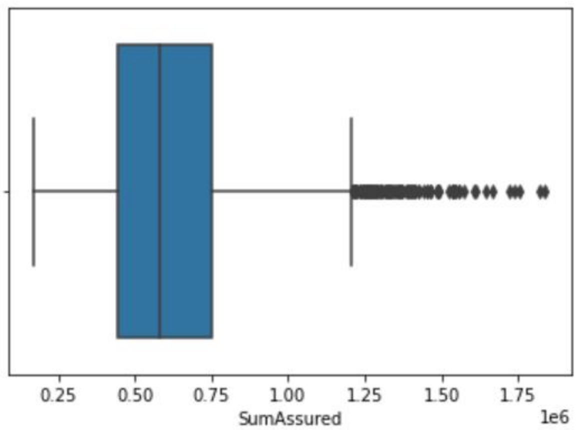


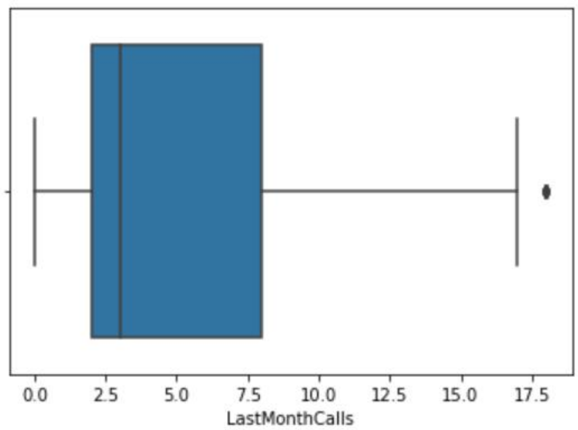






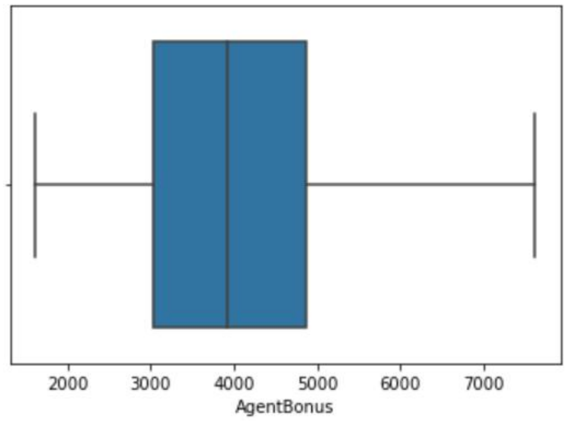


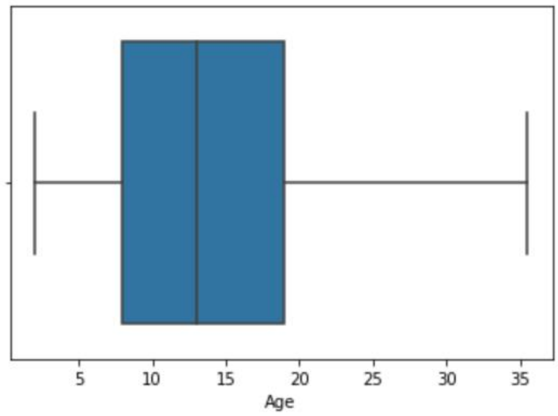


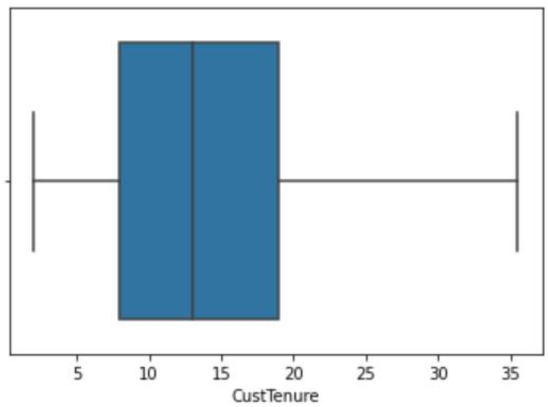


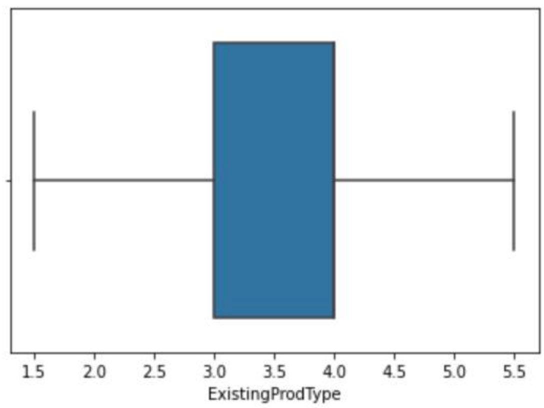
In a favour of doing any outlier treatment as most of the numeric data here has out of bound and hence the outliers might be able add value to the model . More so the numeric data which is continuous has huge outliers. Like the SumAssured or ExistingPolicy tenure has many observation which stands out and most of the others are in the right range.

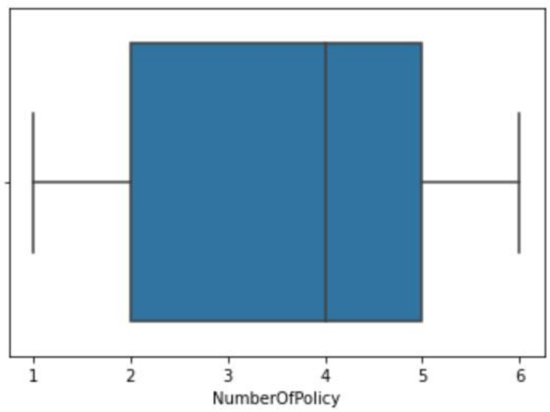
After fixing Outliers:

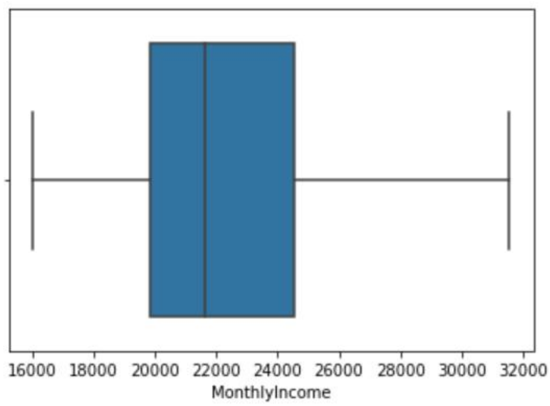


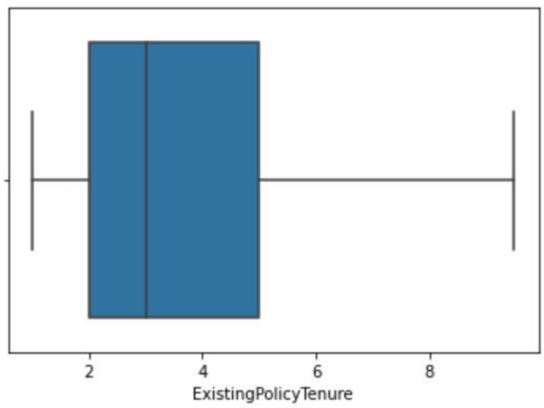


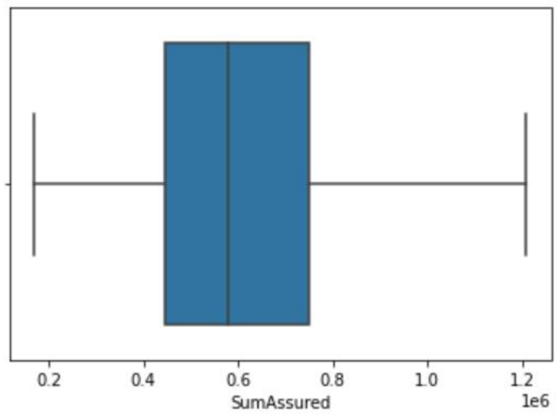


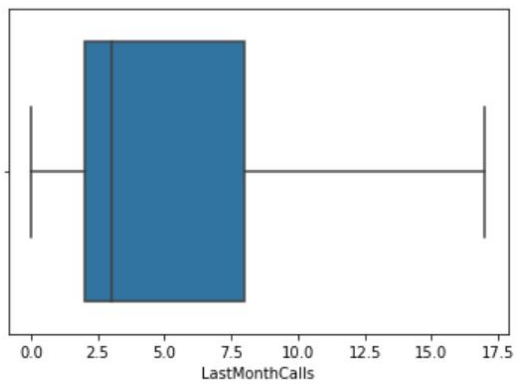


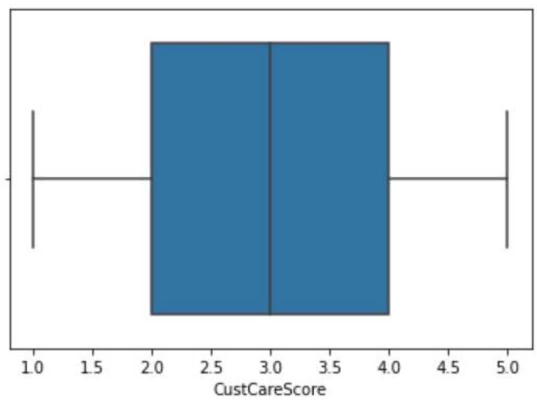





2 4 6 8
ExistingPolicyTenure


0.2 0.4 0.6 0.8 1.0 1.2
SumAssured
1e6


0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
LastMonthCalls


1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
CustCareScore

3.6 Variable Transformation (if required)

```
Channel :  3
Online                   468
Third Party Partner      858
Agent                   3194
Name: Channel, dtype: int64


Occupation :  5
Free Lancer            2
Laarge Business      153
Large Business       255
Small Business      1918
Salaried            2192
Name: Occupation, dtype: int64


EducationField :  7
MBA                  74
UG                  230
Post Graduate       252
Engineer            408
Diploma             496
Under Graduate     1190
Graduate           1870
Name: EducationField, dtype: int64


Gender :  3
Fe male      325
Female      1507
Male        2688
Name: Gender, dtype: int64


Designation :  6
Exe                 127
VP                  226
AVP                 336
Senior Manager      676
Executive          1535
Manager            1620
Name: Designation, dtype: int64


MaritalStatus :  4
Unmarried      194
Divorced       804
Single        1254
Married       2268
Name: MaritalStatus, dtype: int64


Zone :  4
South       6
East       64
North    1884
West     2566
Name: Zone, dtype: int64


PaymentMethod :  4
Quarterly        76
Monthly         354
Yearly         1434
Half Yearly    2656
Name: PaymentMethod, dtype: int64
```

The highlighted data seems to be recorded incorrectly and required replacement and this was done to ensure the right categories are picked up by the model

Fixing Column Values

```
In [11]: df['Gender'] = df['Gender'].replace(to_replace='Fe male',value='Female')
```

```
In [12]: df['Occupation'] = df['Occupation'].replace(to_replace='Laarge Business',value='Large Business')
```

```
In [13]: df['EducationField'] = df['EducationField'].replace(to_replace='UG',value='Under Graduate')
```

```
In [14]: df['Designation'] = df['Designation'].replace(to_replace='Exe',value='Executive')
```

Since the complaint column had only values in 0’s and 1’s but was of numaric type .So we have converted it into categorical value.

```
In [10]: df['Complaint'] = df.Complaint.astype(object)
```

After fixing data:

```
Channel :  3
Online                  468
Third Party Partner     858
Agent                  3194
Name: Channel, dtype: int64


Occupation :  4
Free Lancer          2
Large Business     408
Small Business    1918
Salaried          2192
Name: Occupation, dtype: int64


EducationField :  6
MBA                 74
Post Graduate      252
Engineer           408
Diploma            496
Under Graduate    1420
Graduate          1870
Name: EducationField, dtype: int64


Gender :  2
Female    1832
Male      2688
Name: Gender, dtype: int64


Designation :  5
VP                 226
AVP                336
Senior Manager     676
Manager           1620
Executive         1662
Name: Designation, dtype: int64
```

```
MaritalStatus :  4
Unmarried      194
Divorced       804
Single        1254
Married       2268
Name: MaritalStatus, dtype: int64


Complaint :  2
1    1298
0    3222
Name: Complaint, dtype: int64


Zone :  4
South       6
East       64
North    1884
West     2566
Name: Zone, dtype: int64


PaymentMethod :  4
Quarterly       76
Monthly        354
Yearly        1434
Half Yearly   2656
Name: PaymentMethod, dtype: int64
```

3.7 Addition of new variables (if required)

```
In [24]: df["AgentPerformance"]=np.where(df["AgentBonus"]>=np.mean(df['AgentBonus']),'High','Low')
```

We have added a new column named AgentPerformance where the value of the result column is High if the value of AgentBonous is greater than or equal to it's median else the value as Low.

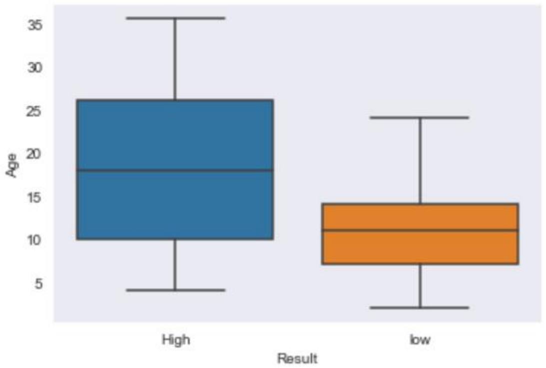
4. Business Insights from EDA

4.1 Is the data unbalanced? If so, what can be done? Please explain in the context of thebusiness

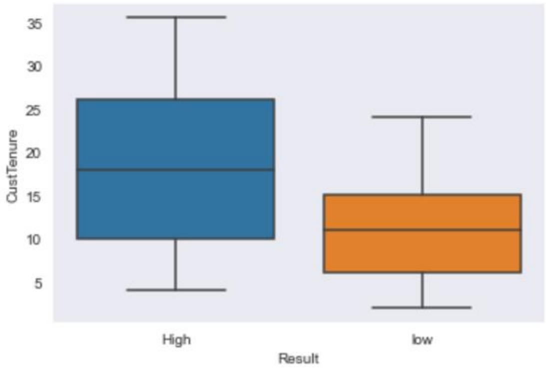
Low 2474
High 2046
Name: Result, dtype: int64

Data is balanced with nearly equal High and Low values. Thus it shows that nearly half of the agent are good performers while the other half are below average.

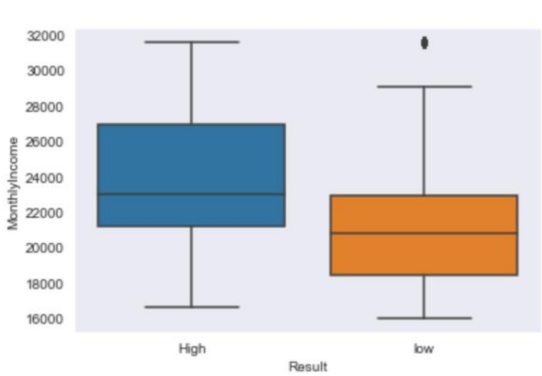
4.2 Any business insights using clustering (if applicable)



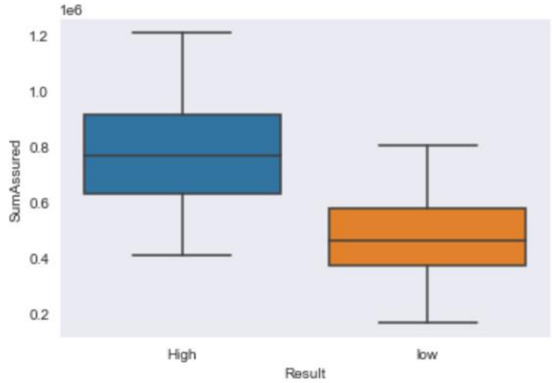
Young customer's Agents have Lower performance



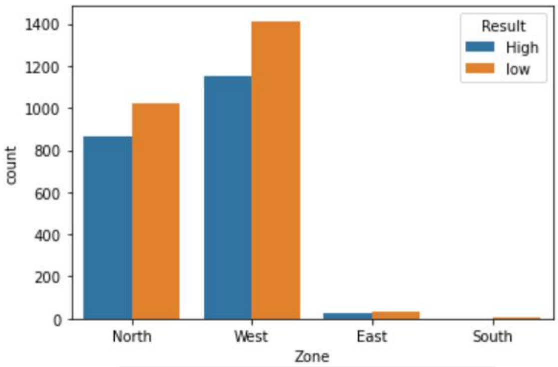
Longer customer tenure results in better Agent performance.



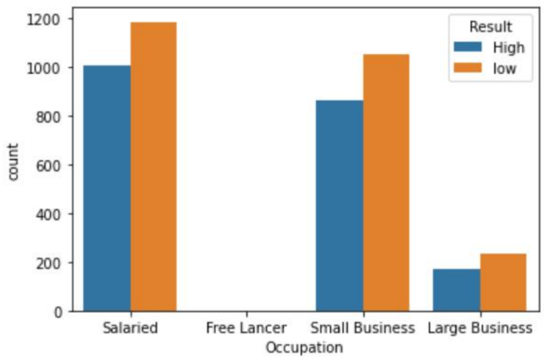
Higher Monthly Income results in better performance



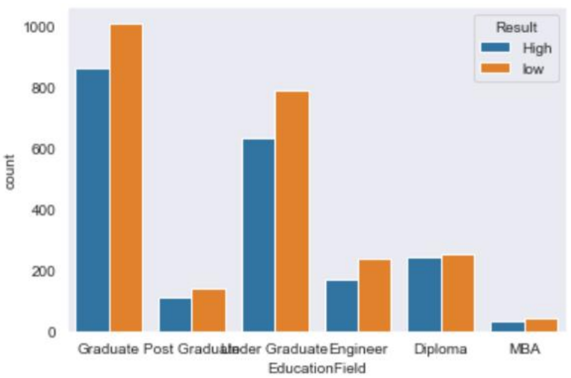
Young customer's Agents have Lower performance



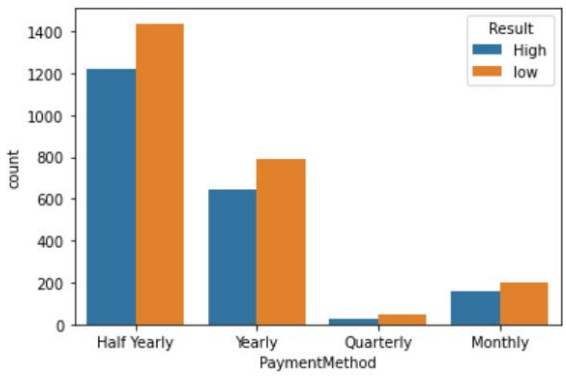
Most customers are from North and West zones



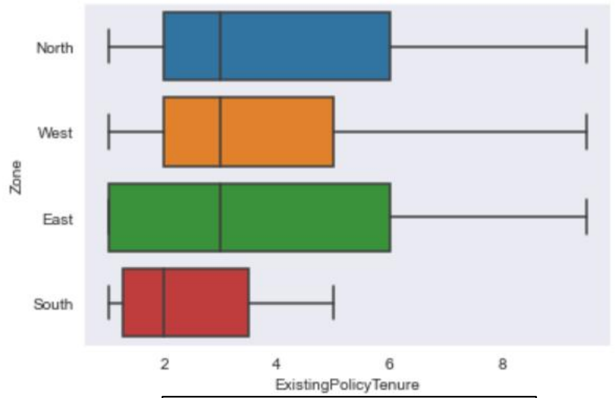
Most customers are Salaried or have small business



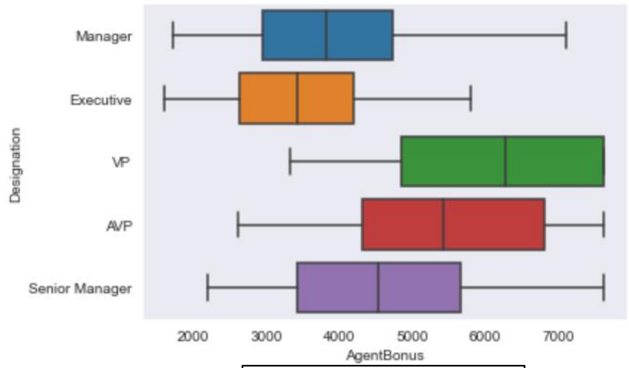
Most customers are Graduate and Under Graduate



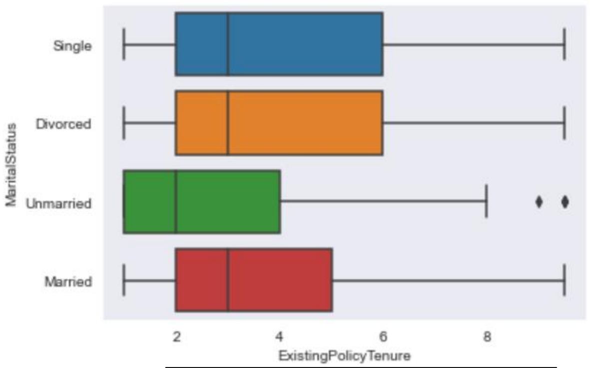
Most customers pay in half yearly method.



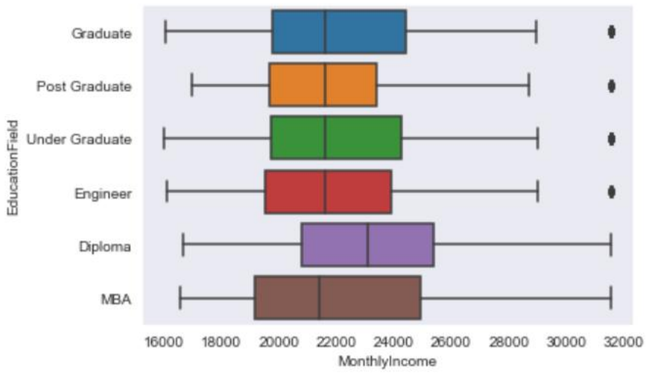
East and North zone has highest customer tenures



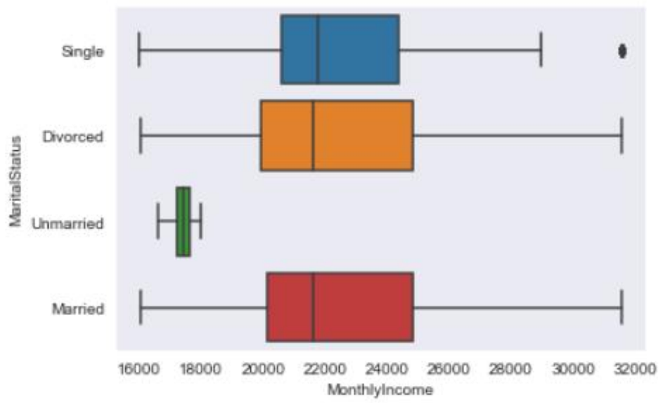
VP’s agents have highest bonus.



Single and Divorced have most existing policy tenure

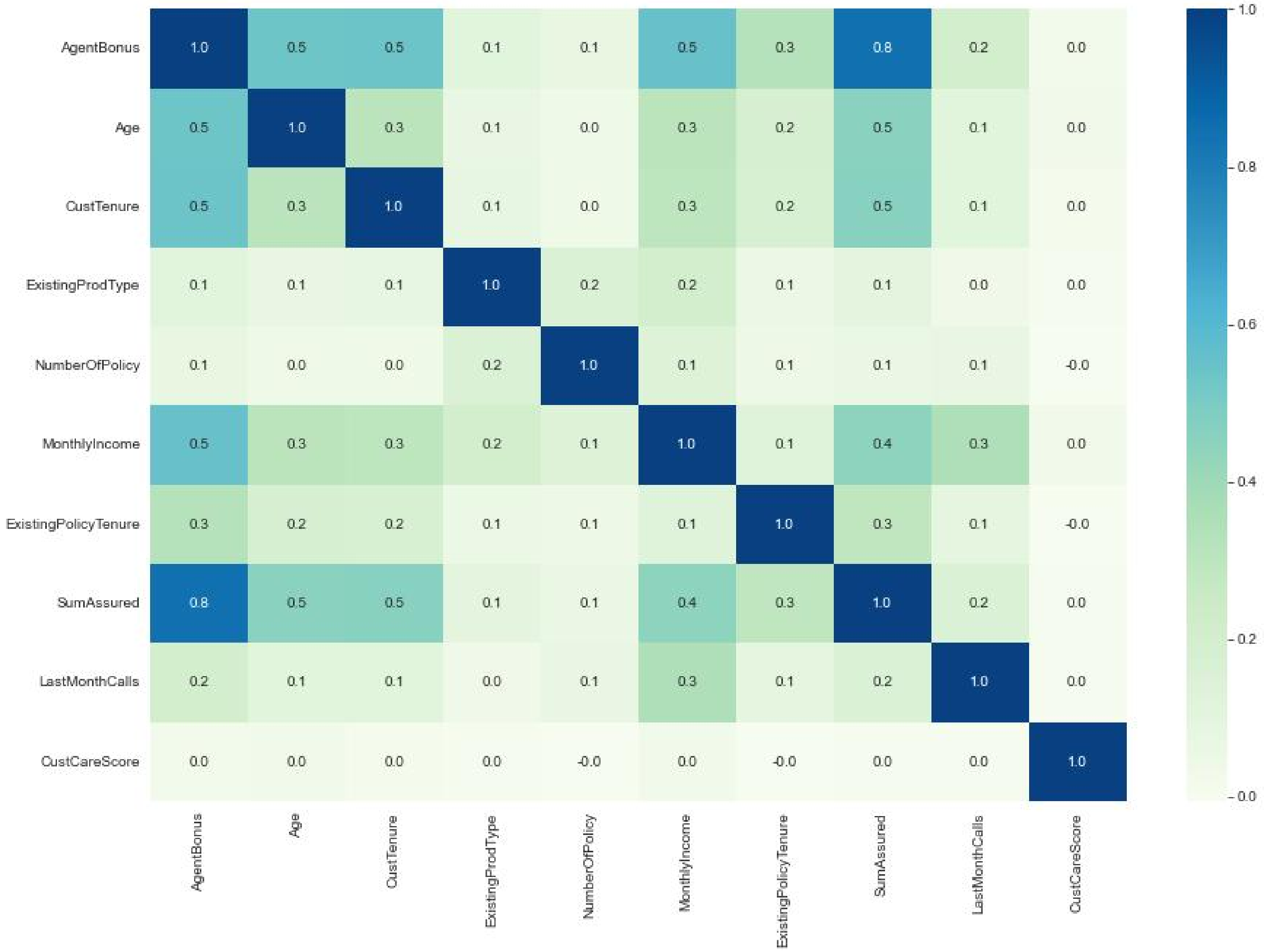


Diploma monthly income is highest



Unmarried customers’ monthly income is less

4.3 Any other business insights



- Age is positively correlated with AgentBonus.
- Cust Tenure is positively correlated with AgentBonus.
- Monthly Income is positively correlated with AgentBonus.
- CustomerCareScore Does not affect any other column.
- NumberOfPolicy has very minimal effect on AgentBonus.
- SumAssured is highly related to AgentBonus.