

# SAS GRADED PROJECT-2

## DOMAIN: INSURANCE

NAME: PANAGAM MOHITHA

/\*Question1: Import dataset in the SAS environment and check top 10 record of import dataset\*/

```
FILENAME REFFILE '/home/u61856037/sasuser.v94/Life+Insurance+Dataset.csv';
PROC IMPORT DATAFILE=REFFILE
    DBMS=CSV
    OUT=Life_Insurance_data;
    GETNAMES=YES;
RUN;
PROC CONTENTS DATA=Life_Insurance_data; RUN;
```

/\*Question2: Check variable type of the import dataset\*/

```
proc contents data=Life_Insurance_data varnum;
run;
```

Inference:

- There are 20 variables in the imported dataset in which 14 are numerical and 6 are character based variables.

/\*Question3: Checks if any variables have missing values, if yes then do treatment?\*/

```
proc means data=Life_Insurance_data nmiss;
run;
```

Inference:

- There are no missing values in any variables. There are 1924 observations for all variables.

/\*Question4: Check summary and percentile distribution of all numerical variables for churners and non-churners?\*/

```
proc means data=Life_Insurance_data n nmiss min p1 p5 p10 p25 p50 p75 p90 p95 p99 max
maxdec=0;
var Age Cust_Tenure Overall_cust_satisfaction_score CC_Satisfaction_score Cust_Income
Agent_Tenure
YTD_contact_cnt Due_date_day_cnt Existing_policy_count Miss_due_date_cnt ;
run;
```

Inference:

- The percentile distribution of all numerical variables are shown in results.

```

/*Question5: Check for outlier, if yes then do treatment*/
proc univariate data=Life_Insurance_data;
var Age Cust_Tenure Overall_cust_satisfaction_score CC_Satisfaction_score Cust_Income
Agent_Tenure
YTD_contact_cnt Due_date_day_cnt Existing_policy_count Miss_due_date_cnt;
run;
/*we have some outlier and below is the flooring and capping for those variables*/
data Life_Insurance_data;
set Life_Insurance_data;
if Cust_Income > 35999 then Cust_Income = 35999;
run;
/*checking distribution after flooring and capping*/
proc univariate data=Life_Insurance_data;
var Age Cust_Tenure Overall_cust_satisfaction_score CC_Satisfaction_score Cust_Income
Agent_Tenure
YTD_contact_cnt Due_date_day_cnt Existing_policy_count Miss_due_date_cnt;
run;

```

Inference:

- There are two outliers i.e., 95000 and 96000 in customer income. These outliers are corrected as 35999.

```

/*Question6: Check the proportion of all categorical variables and
extract percentage contribution of each class in respective variables*/
proc freq data=Life_Insurance_data;
table Payment_Period Product EducationField Gender Cust_Designation Cust_MaritalStatus
Complaint/ nocum;
run;

```

Inference:

- Yearly paid policies has major share paid by customers. Customers aligned to choose traditional products.
- Most of the customers are from statistics and CA educational background with executive and manager designation. Half of the customers are married.
- Nearly 71% of customers lodged complaint.

```

/*Question7: Customer service management want you to create a macro where they will just
put mobile number
and they will get all the important information like Age, Education, Gender, Income and
CustID*/
/*Created Macro*/
%MACRO Customer_info();
DATA output (keep = Age EducationField Gender Cust_Income CustID);
SET Life_Insurance_data;
where Mobile_num in (&Mobile_num.);
RUN;
proc print data=output;
run;
%MEND;
/*Provided input mobile number*/
%let Mobile_num = 9878913773,9898819662,9904978124,9887638137,9882200862;
/*run macro for output*/
%Customer_info;

```

Inference:

- A macro is created in which you get customer's details but inputting the customer mobile number.

```

/*Question8: Check correlation of all numerical variables before building model, because
we cannot add correlated variables in model?*/
proc corr data=Life_Insurance_data NOPROB;
var Age Cust_Tenure Overall_cust_satisfaction_score CC_Satisfaction_score Cust_Income
Agent_Tenure
YTD_contact_cnt Due_date_day_cnt Existing_policy_count Miss_due_date_cnt ;
run;

```

**Inference:**  
There are no missing values.

```

/*Question9: Create train and test (70:30) dataset from the existing data set. Put seed
1234?*/
proc freq data=Life_Insurance_data;
table Churn /nocum;
run;
proc surveyselect data= Life_Insurance_data method = srs rep=1
samsize=600 seed = 1234 out =test;
RUN;
proc contents data=test varnum;
run;
proc freq data=test;
table Churn /nocum;
run;
proc sql;
create table train as select t1.* from Life_Insurance_data as t1
where CustID not in (select CustID from test);
quit;
proc freq data=train;
table Churn /nocum;
run;

```

```

/*Question10: Develop linear regression model first on the target variable to extract VIF
information to check multicollinearity?*/
proc reg data=Life_Insurance_data;
model Churn=Age Cust_Tenure Overall_cust_satisfaction_score CC_Satisfaction_score
Cust_Income Agent_Tenure
YTD_contact_cnt Due_date_day_cnt Existing_policy_count Miss_due_date_cnt / vif tol
collin;
title 'Life_Insurance_data - Multicollinearity Investigation of VIF';
run;
quit;

```

**Inference:**

- Chi square statistic is 0.6695

```

/*Question11: Create clean logistic model on the target variables?*/
%let var = Age Cust_Tenure Overall_cust_satisfaction_score CC_Satisfaction_score
Cust_Income Agent_Tenure
YTD_contact_cnt Due_date_day_cnt Existing_policy_count Miss_due_date_cnt ;
proc logistic data=train descending outmodel=model;
model Churn = &var / lackfit;
output out = train_output xbeta = coeff stdxbeta = stdcoeff predicted = prob;
run;

```

```
/*Question12: Create a macro and take a KS approach to take a cut off on the calculated scores?*/  
proc univariate data=Life_Insurance_data;  
var Overall_cust_satisfaction_score;  
    histogram Overall_cust_satisfaction_score / normal(mu=est sigma=est);  
run;
```

**Inference:**

The Kolmogorov-Smirnov test for overall satisfaction score cutoff is 0.1811

```
/*Question13: Predict test dataset using created model?*/  
proc reg data=Life_Insurance_data outest=test1;  
model Churn=Age Cust_Tenure Overall_cust_satisfaction_score CC_Satisfaction_score  
Cust_Income Agent_Tenure  
YTD_contact_cnt Due_date_day_cnt Existing_policy_count Miss_due_date_cnt ;  
run;  
  
proc score data=Life_Insurance_data score=test1 type=parms predict out=test2;  
var Age Cust_Tenure Overall_cust_satisfaction_score CC_Satisfaction_score Cust_Income  
Agent_Tenure  
YTD_contact_cnt Due_date_day_cnt Existing_policy_count Miss_due_date_cnt ;  
run;
```