

Execution Environment

Author: u61856037
 File: /home/u61856037/sasuser.v94/Insurance Assignment2.sas
 SAS Platform: Linux LIN X64 3.10.0-1062.12.1.el7.x86_64
 SAS Host: ODAWS02-APSE1-2.ODA.SAS.COM
 SAS Version: 9.04.01M6P11072018
 SAS Locale: en_US
 Submission Time: 7/17/2022, 12:32:17 PM
 Browser Host: 157.47.48.2
 User Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36
 Application Server: ODAMID00-APSE1-2.ODA.SAS.COM

Code: Insurance Assignment2.sas

```

/*-----*/
/*-----SAS GRADED PROJECT-----*/
/*-----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;

/*Question3: Checks if any variables have missing values, if yes then do treatment?*/
proc means data=Life_Insurance_data nmiss;
run;

/*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;

/*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;

/*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;

/*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;

/*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;

/*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
sampsiz=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;

/*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;

/*Question13: Predict test dataset using created model?*/
/*Predicting by equation, you can use score statment, in my version of SAS score function is not
present*/
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;

```