1. Use logistic regression model to the “IBM Employee Attrition V2” dataset in CANVAS to uncover the features that can predict employee attrition. This is a subset of a fictional data set created by IBM data scientists.

Sol:

First, classify the variables.

Set the “attrition = 1” if the label of it is “Yes”, otherwise set it to “0”.

Set the “overtime= 1” if the label of it is “Yes”, otherwise set it to “0”.

And the same classify variable NumCompaniesWorked and TotalWorkingYears and MaritalStatus.

Then run the logistic regression model.

Code:

\*-------------------------------------------------------------------------;

\* Project : CS 593: hw4 ;

\* Developer(s) : Rui li ;

\* Comments : code ; ;

\*-------------------------------------------------------------------------;

libname sasdata "C:\Users\Eric\Downloads\SAS\_DATA";

**proc** **import** datafile= 'C:\Users\Eric\Downloads\SAS\_DATA\IBM\_Employee\_Attrition\_V2.csv' out =IBM\_Employee\_Attrition;

**run**;

**data** IBM\_Employee\_Attrition2;

set IBM\_Employee\_Attrition;

if attrition="No"

then V\_attrition=**0**;

else V\_attrition=**1**;

if overtime="Yes" then V\_overtime=**1**;

else V\_overtime=**0**;

if NumCompaniesWorked>**3** then V\_NumCompaniesworked=**1**;

else V\_NumCompaniesworked=**0**;

if TotalWorkingYears > **11** then V\_TotalWorkingYears = **1**;

else V\_TotalWorkingYears = **0**;

if MaritalStatus="Single" or MaritalStatus="Married" then V\_MaritalStatus=**1**;

else V\_MaritalStatus=**0**;

**run**;

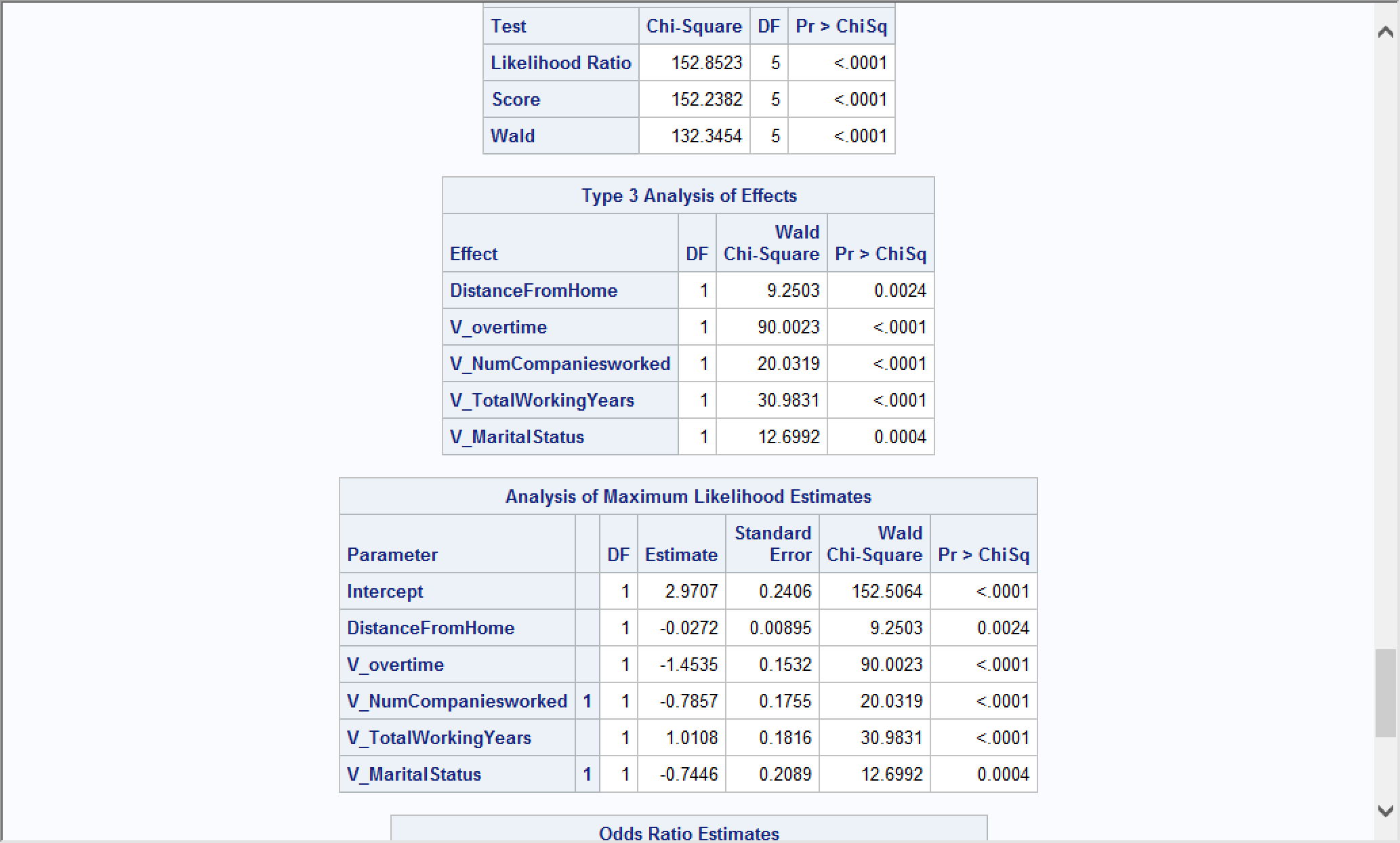
**proc** **logistic** data=IBM\_Employee\_Attrition2 ;

class V\_NumCompaniesworked(ref='0') TotalWorkingYears(ref='0') V\_MaritalStatus(ref='0')/param=ref;

model V\_attrition=DistanceFromHome V\_overtime V\_NumCompaniesworked V\_TotalWorkingYears V\_MaritalStatus;

**quit**;

Result:



According to the result, all the variables are satisfied by the threshold of P value which “Pr > ChiSq” < 0.05. Therefore DistanceFromHome, Overtime, NumCompaniesworked, MaritalStatus, TotalWorkingYears can predict employee attrition.

1. Calculate the page rank for the following network.

Sol:

Code:

/\*PageRank Soluion\*/

**data** Arcs;

infile datalines;

input Node $ A B C D E F G ;

datalines;

A 0 1 1 1 0 0 0

B 1 0 0 1 0 0 1

C 0 0 0 0 1 0 0

D 0 1 1 0 0 0 0

E 0 0 0 0 0 1 0

F 0 0 1 0 0 0 0

G 0 1 0 0 0 0 1

;

**run**;

/\*get the transition matrix\*/

**proc** **sql**;

create table matrix\_1 as

select a/sum(a) as x1

,b/sum(b) as x2

,c/sum(c) as x3

,d/sum(d) as x4

,e/sum(e) as x5

,f/sum(f) as x6

,g/sum(g) as x7

from Arcs

;

**quit**;

/\*Since there are 7 nodes, the initial vector v0 has 7 components, each 1/7\*/

**data** rank\_p;

x1=**1**/**7**;

x2=**1**/**7**;

x3=**1**/**7**;

x4=**1**/**7**;

x5=**1**/**7**;

x6=**1**/**7**;

x7=**1**/**7**;

output;

**run**;

**proc** **iml**;

use matrix\_1;

read all var{x1 x2 x3 x4 x5 x6 x7} into M;

print M;

use rank\_p;

read all var{x1 x2 x3 x4 x5 x6 x7} into rank\_p1;

print rank\_p1;

rank\_p=t(rank\_p1);

print rank\_p;

rank\_p2=M\*rank\_p;

print rank\_p2;

rank\_p50=(M\*\***50**)\*rank\_p;

print rank\_p50;

**quit**;

