**Module 8 Critical Thinking: Portfolio Project**

Tyler Ivory

Colorado State University Global

22SA-MIS581-1 Capstone: Business Intelligence and Data Analytics

Dr. Orenthio Goodwin

5/15/2022

**Github Link:**

https://github.com/tivory99/FraudClassification

**SAS Studio Code and screenshots:**

/\* Generated Code (IMPORT) \*/

/\* Source File: Fraud.csv \*/

/\* Source Path: /home/u51638392/Fraud\_Analysis \*/

/\* Code generated on: 5/10/22, 9:41 PM \*/

%web\_drop\_table(WORK.IMPORT);

FILENAME REFFILE '/home/u51638392/Fraud\_Analysis/Fraud.csv';

PROC IMPORT DATAFILE=REFFILE

DBMS=CSV

/\*

\*

\* Task code generated by SAS Studio 3.8

\*

\* Generated on '5/10/22, 9:46 PM'

\* Generated by 'u51638392'

\* Generated on server 'ODAWS04-USW2.ODA.SAS.COM'

\* Generated on SAS platform 'Linux LIN X64 3.10.0-1062.9.1.el7.x86\_64'

\* Generated on SAS version '9.04.01M6P11072018'

\* Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36'

\* Generated on web client 'https://odamid-usw2.oda.sas.com/SASStudio/main?locale=en\_US&zone=GMT-06%253A00&ticket=ST-98093-DfyV3PWzEHf01BbZAbcd-cas'

\*

\*/

ods graphics / reset width=6.4in height=4.8in imagemap;

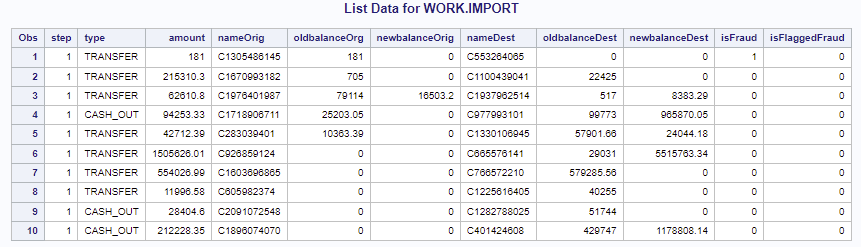
proc sgplot data=WORK.IMPORT1;

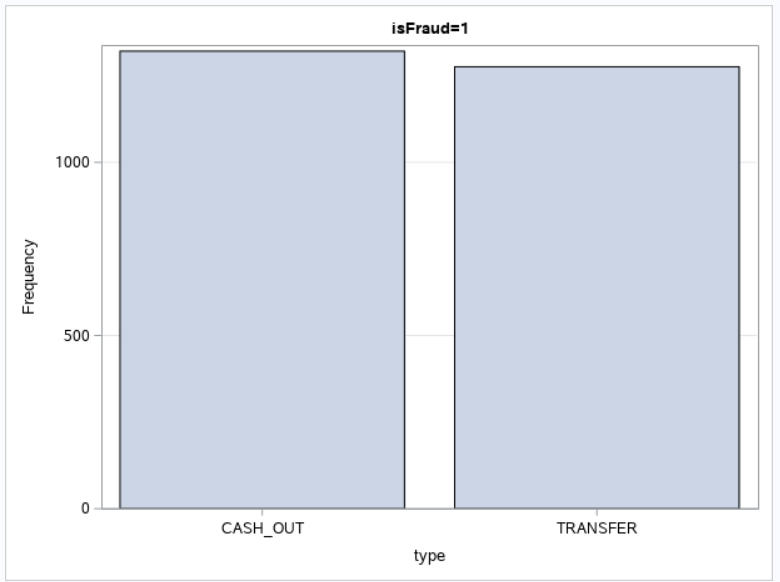
vbar type /;

yaxis grid;

run;

ods graphics / reset;

****

****

**R Studio Code and screenshots:**

#import data and drop columns

CREDITFRAUDDATASET <- read.csv("C:/Users/tivory/Downloads/CREDITFRAUDDATASET.csv")

FraudData <- CREDITFRAUDDATASET[-c(1,2,4,7,11)]

#split data into testing and training subsets

fraud\_split <- sample.split(Y = FraudData$isFraud, SplitRation = 0.7)

train\_set <- subset(x = FraudData, fraud\_split == TRUE)

test\_set <- subset(x = FraudData, fraud\_split == FALSE)

#create logistic regression

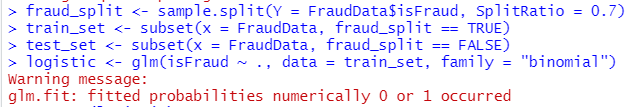
logistic <- glm(isFraud ~ ., data = train\_set, family = "binomial")

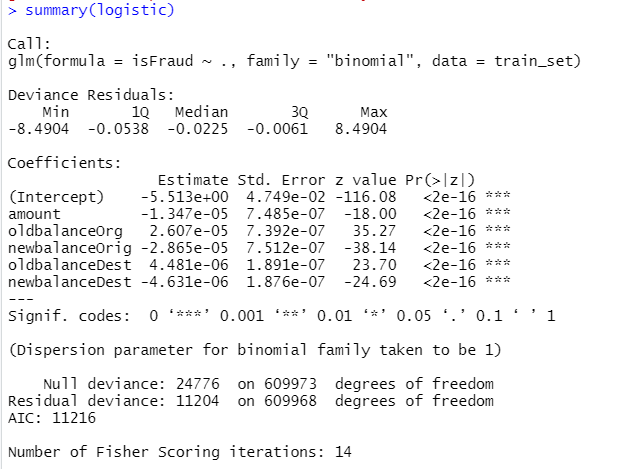
#evaluate model

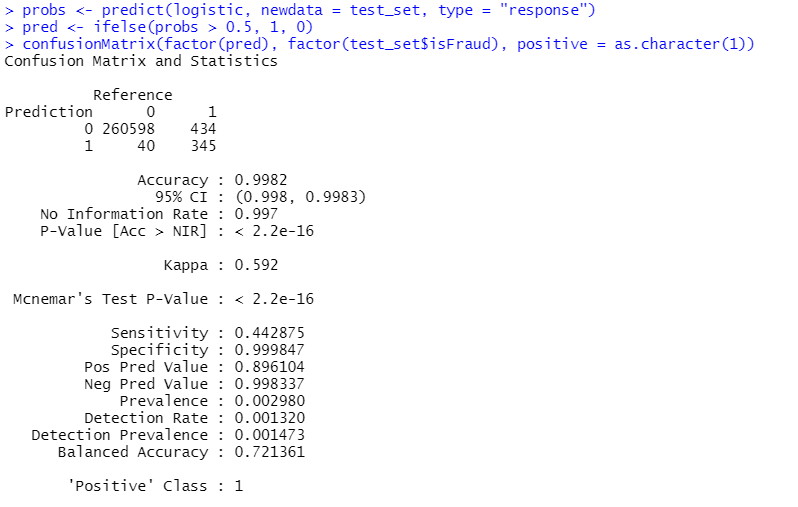
probs <- predict(logistic, newdata = test\_set, type = "response")

pred <- ifelse(probs > 0.5, 1, 0)

confusionMatrix(factor(pred), factor(test\_set$isFraud), positive = as.character(1))







**Python Code and Screenshots:**

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

#import csv into dataset

dataset = pd.read\_csv('CREDITFRAUDDATASET.csv')

#create input data and classifier data

X= dataset.drop(columns=['step', 'nameOrig', 'type', 'nameDest', 'isFlaggedFraud', 'isFraud'])

y= dataset.drop(columns=['step', 'amount', 'nameOrig', 'type', 'oldbalanceOrg', 'newbalanceOrig', 'newbalanceDest', 'nameDest', 'oldbalanceDest', 'isFlaggedFraud'])

#set shape to be equal

X= dataset.iloc[:,0:8]

y= dataset.iloc[:,8]

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X = sc.fit\_transform(X)

#split data into testing and training datasets

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3)

#create model

from keras import Sequential

from keras.layers import Dense

classifier = Sequential()

#First Hidden Layer

classifier.add(Dense(4, activation='relu', kernel\_initializer='random\_normal', input\_dim=5))

#Second Hidden Layer

classifier.add(Dense(4, activation='relu', kernel\_initializer='random\_normal'))

#Output Layer

classifier.add(Dense(1, activation='sigmoid', kernel\_initializer='random\_normal'))

#compile neural network

classifier.compile(optimizer ='adam',loss='binary\_crossentropy', metrics =['accuracy'])

#run model

classifier.fit(X\_train,y\_train, batch\_size=10, epochs=100)

#evaluate model accuracy

eval\_model=classifier.evaluate(X\_train, y\_train)

eval\_model

#create confusion matrix

y\_pred=classifier.predict(X\_test)

y\_pred =(y\_pred>0.5)

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

#create visualization of neural network

from ann\_visualizer.visualize import ann\_viz

import os

os.environ["PATH"] += os.pathsep + "C:/Program Files/Graphviz/bin/"

ann\_viz(classifier, title="Fraud Detection Neural Network")

