Evolution 3

Assessing Data Management

By: Trevor Hofmann

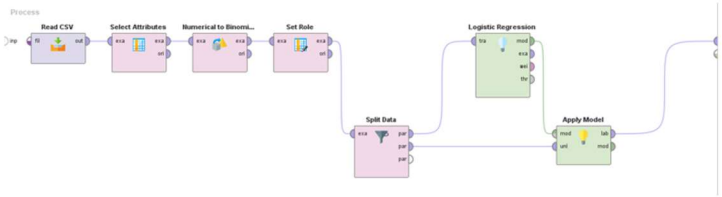
What do you know?

My competencies, skills, and knowledge in terms of Data Management would rank third of the four curriculum areas. I have been working full time in IT for right at 7 years now, since 2017 when I graduated with my bachelors in Business Administration Computer Information Systems for West Texas A&M. During my career I've not worked with any kind of Data Mining or Management beyond Excel and some lite SQL. During my undergrade degree we did utilize Tableau to present data in one class but that was all I had used it for. I had taken classes that went over ERD diagrams, SQL, and business rules.

While working on my graduate degree I've taken two classes around Data Management, CIDM 6350 Data & Information Management and CIDM 6355 Data Mining Methods. For 6355 we performed a group project where we selected a data source and choose 4 data mining methods that we applied to the data utilizing Rapid Miner and RStudio. The four modes we choose where Decision Tree, Logistic Regression, Neural Net, and a Naïve Bayes. Me and one other group member specifically built the models in Rapid Miner and RStudio. Here are a couple screenshots of the models:

Logistic Regression
Rapid Miner

- Import CSV
- Select Attributes
- Convert Numerical to Polynomial
- Set Role for HighValueCar
- Split Data 30/70
- Create Logistic Regression Model
- Apply Model with remaining Split Data



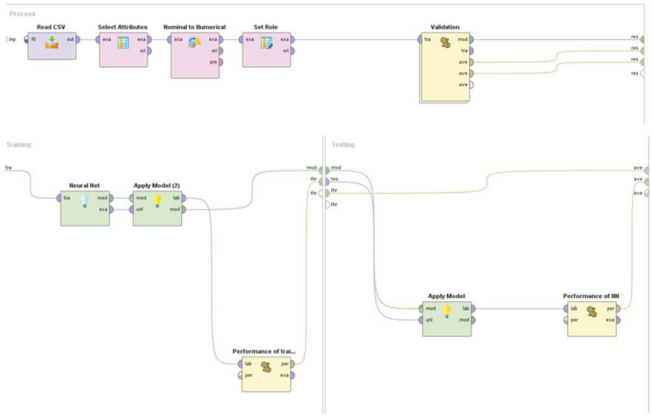
Logistic Regression Rstudio

- Import/Install Packages
- Import CSV
- Create Sample/Train Data
- Create Neural Net Model
- Apply It
- Show Confusion Matrix

```
install.packages("NeuralNetTools")
library(NeuralNetTools)
install.packages("nnet")
library(nnet)
install.packages("e1071")
library(e1071)
install.packages("caret")
library(caret)

GroupData <- read.csv(file.choose(), header = TRUE)

set.seed(123)
smp_size <- floor(0.7 * nrow(GroupData))
train_ind <- sample(seq_len(nrow(GroupData)), size = smp_size)
train <- GroupData[train_ind, ]
test_ind <- setdiff(seq_len(nrow(GroupData)), train_ind)
test <- GroupData[test_ind, ]

# Logistic Regression Model
LRmodel <- glm(HighValueCar ~ ., family = binomial, data = train)
LRp <- predict(LRmodel, test, type = "response")
LRpredict <- ifelse(LRp > 0.20, 1, 0)
table(LRpredict, test$HighValueCar)

# Neural Network Model
GroupNN <- nnet(HighValueCar ~ ., data = train, size = 8, maxit = 100000, decay = 0.01)
GroupNN_probabilities <- predict(GroupNN, newdata = test, type = "raw")
GroupNN_predict <- ifelse(GroupNN_probabilities > 0.5, 1, 0)
GroupNN_factor <- as.factor(GroupNN_predict)

LRconfusionMatrix <- confusionMatrix(as.factor(LRpredict), as.factor(test$HighValueCar))
print("Logistic Regression Confusion Matrix:")
print(LRconfusionMatrix)

NNconfusionMatrix <- confusionMatrix(GroupNN_factor, as.factor(test$HighValueCar))
print("Neural Network Confusion Matrix:")
print(NNconfusionMatrix)
```

Neural Net Rapid Miner

- Import CSV
- Select Attributes
- Convert Numerical to Polynomial
- Set Role for HighValueCar
- Run Validation
- Create Neural Net Model
- Apply Model
- Measure Performance



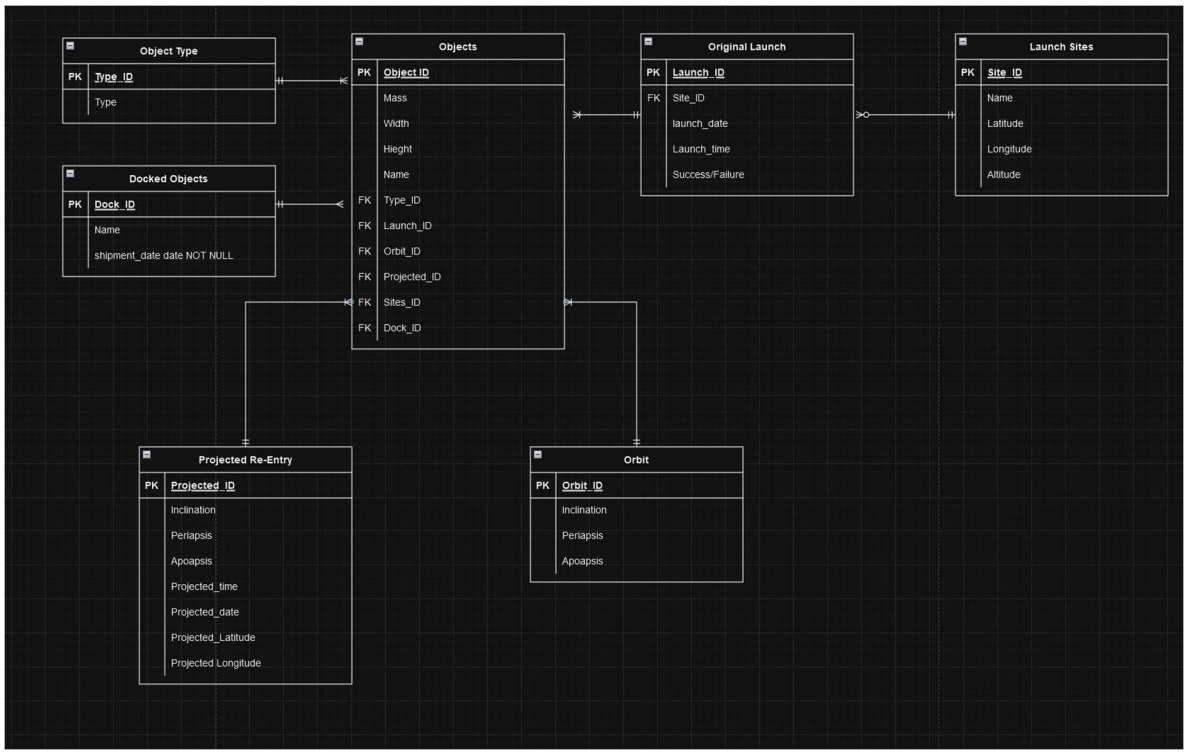
Neural Net Rstudio

- Import/Install Packages
- Import CSV
- Create Sample/Train Data
- Create Neural Net Model
- Apply It
- Show Confusion Matrix

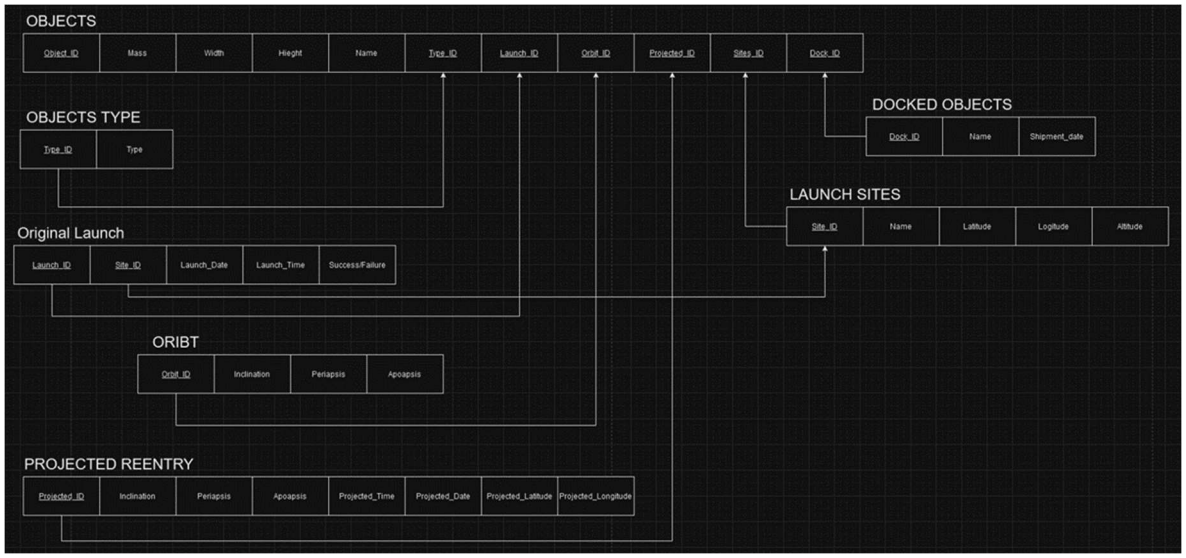
```
install.packages("NeuralNetTools")
library(NeuralNetTools)
install.packages("nnet")
library(nnet)
install.packages("e1071")
library(e1071)
install.packages("caret")
library(caret)

GroupData <- read.csv(file.choose(), header = TRUE)

set.seed(123)
smp_size <- floor(0.7 * nrow(GroupData))
train_ind <- sample(seq_len(nrow(GroupData)), size = smp_size)
train <- GroupData[train_ind, ]
test_ind <- setdiff(seq_len(nrow(GroupData)), train_ind)
test <- GroupData[test_ind, ]

# Logistic Regression Model
LRmodel <- glm(HighValueCar ~ ., family = binomial, data = train)
LRp <- predict(LRmodel, test, type = "response")
LRpredict <- ifelse(LRp > 0.20, 1, 0)
table(LRpredict, test$HighValueCar)

# Neural Network Model
GroupNN <- nnet(HighValueCar ~ ., data = train, size = 8, maxit = 100000, decay = 0.01)
GroupNN_probabilities <- predict(GroupNN, newdata = test, type = "raw")
GroupNN_predict <- ifelse(GroupNN_probabilities > 0.5, 1, 0)
GroupNN_factor <- as.factor(GroupNN_predict)

LRconfusionMatrix <- confusionMatrix(as.factor(LRpredict), as.factor(test$HighValueCar))
print("Logistic Regression Confusion Matrix:")
print(LRconfusionMatrix)

NNconfusionMatrix <- confusionMatrix(GroupNN_factor, as.factor(test$HighValueCar))
print("Neural Network Confusion Matrix:")
print(NNconfusionMatrix)
```

For 6350 Data & Information Management we did a project that focused on all the areas we had learned about that semester. This included creating business rules, EERD, relation diagrams in the 3rd normal form, and SQL code & queries. Here are some examples from the project:

EERD



Relational Diagram



SQL Code

SQL Create Tables

```
--Create the database itself
CREATE SCHEMA Project;
-- Object Type Table
CREATE TABLE Project.Object_Type (
    Type_ID INT PRIMARY KEY,
    Type VARCHAR(255)
);
-- Launch Sites Table
CREATE TABLE Project.Launch_Sites (
    Site_ID INT PRIMARY KEY,
    Name VARCHAR(255),
    Latitude FLOAT,
    Longitude FLOAT,
    Altitude FLOAT
);
-- Original Launch Table
CREATE TABLE Project.Original_Launch (
    Launch_ID INT PRIMARY KEY,
    Site_ID INT,
    Launch_Date DATE,
    Launch_Time TIME,
    Success_Failure VARCHAR(50),
    FOREIGN KEY (Site_ID) REFERENCES Project.Launch_Sites(Site_ID)
);
-- Orbit Table
CREATE TABLE Project.Orbit (
    Orbit_ID INT PRIMARY KEY,
    Inclination FLOAT,
    Periapsis FLOAT,
    Apoapsis FLOAT
);
```

Where are you are weak?

I am more confident at data management and less confident in data mining as a whole. Since I've not worked on either of these areas in my career, they are both on the weaker end of my skills with data mining the lesser of the two. For the data management project, I took more time working on the relational diagram and business rules compared to the EERD and SQL code.

When it comes to data mining, we focused on using Rapid Miner and Rstudio. I found RStudio to be more difficult to jump in and learn especially in comparison to Rapid miner. During my project in 6355 it took me much longer to troubleshoot and create the Rstudio models. Of the models we worked on I felt like Naïve Bayes and Neural Net I struggled more compared to Decision Tree and Logistic Regression.

The Future?

In my career I've found understanding how SQL works and data management in general has been helpful but far from necessary. This is normally caused by vendors being able to manage or

support issues with any SQL databases there applications require. I do not see this changing in the near future. I do see it being helpful being able to understand how to best manage data.

When data management is paired with data mining it changes how I see the future. Being able to process large amounts of data through mining is going to be more and more prevalent, especially in a management role. A manager needs to be able to make effective decisions. I feel like that is stepping into data analytics though so I will discuss that more in that paper.