**Admiral Data Science Take Home Assignment**

For this assignment you are given a sample of realistic policy data from customers of an insurance company, along with some geographic data on the areas where the customers live.

The assignment is not intended to take more than 3-4 hours to complete, if you find yourself stuck feel free to ask questions (by email: duncan.procter@admiralgroup.co.uk), we will try to respond as soon as is reasonable.

**Section 1 – SQL Querying**

For this section, complete each answer by writing a SQL query to answer the following questions. Some questions require a small degree of interpretation. Do your best to determine what makes sense in the context of performing business analysis. As long as your interpretation of the question is reasonable (defensible) and the SQL is correct, the answer will be considered correct!

*An easy way to test SQL queries for correctness is to use a small SQLite database or a tool similar to the package* ***sqldf*** *in R or the* ***sqlite3*** *package in Python. Any dialect is okay, but sticking to SQLite and T-SQL (Microsoft) are encouraged.*

1. *What is the total number of claims by state?*

#Using ClaimFlag column ‘1’ for identifying if a policy claimed.

﻿ A picture containing table

Description automatically generated

1. *What is the average claim amount by state?*

Table

Description automatically generated

1. *Transform the continuous variable Applicant Age into a categorical variable with roughly equal numbers of policies in each.*

I usually do this type of data transformation inside of R, rather than SQL, hence I’ve attached both methods, SQLite and R versions.

Table

Description automatically generated

Table

Description automatically generated

1. *Join the geographic data to the policy data. Compute the mean for each geographic feature grouped by whether or not the customer currently has insurance.*

#Classifying a customer that currently has insurance as TimeInsured < 1. I’ve made the assumption that policies end when TimeInsured = 1, as this is the maximum value.

Text

Description automatically generated

1. *How many distinct postal codes are seen in the data set?*

Graphical user interface, text, application

Description automatically generated

**Section 2 – Data Exploration**

You have been given this dataset with the instructions to analyze it and present key findings to your team members. Your team members are curious as to whether the dataset is complete, consistent, and potentially worth building a model with. Create a short presentation (3-5 slides, an Excel worksheet, or 1-2 pages word doc) explaining the steps you took to understand the data, along with key findings and insights. Include anything that might be relevant for using the data in modeling.

*The focus here is on the clear communication of the analysis, not necessarily polished graphics or a “client ready” presentation.*

Please go to *High Level Data Overview.pptx*

**Section 3 – Modeling Claim Likelihood**

Build a model to predict the likelihood that a given policy will file a claim. Keep in mind that each customer has not been insured for a uniform period or represents the same amount of risk! You are free to use R or Python, and any model type, though be prepared to defend your choice. Document how and why you made any choices you did to build the model an how the model performs. Make sure to comment on how you think this model could benefit an insurer.

----- ----- -----

My initial thoughts were to use a Decision Tree. The reason for this was because our response variable is categorical (0/1), and also, the results of the model would be interpretable, which is ideal for customers, if they asked why they did not get accepted/offered insurance. I have given comment within the scripts.

* *0\_model\_source.dt.R*
  + Set working directory to Source File location.
  + Select all to run code (given all libraries installed).
    - Script will pull data (cleaning, transforming etc) from SQLite DB, train model using C50, and produce results on test data.

However, the results of the decision tree were not as accurate as I would have liked them to be, hence, I also built a neural network, using Keras. I have given comment within the scripts.

* *0\_model\_source.nn.R*
  + Set working directory to Source File location.
  + Select all to run code (given all libraries installed).
    - Script will pull data (cleaning, transforming etc) from SQLite DB, train model using Keras, tune hyperparameters, and produce results using the optimal model on the test data.