

Azure PipeLine / Shiny R app :   
Credit Card Fraud Modeling and Deployment

**Project Proposal**

Higher Diploma in Science in Data Analytics

Word Count: 1500

# Contents

[Contents 1](#_Toc43927267)

[1. Introduction 2](#_Toc43927268)

[2. Project Scope and Objectives 3](#_Toc43927269)

[3. Student’s Learning Objectives 4](#_Toc43927270)

[4. Technical Specification of the Project 5](#_Toc43927271)

[5. Project Plan 8](#_Toc43927272)

[6. Conclusion 0](#_Toc43927273)

[7. References / Bibliography 1](#_Toc43927274)

10524150 / Ciaran Finnegan

10524150@mydbs.ie / ciaran@feefinnegan.com

24/06/2020

# Introduction

Over the last decade my career has centred on the management of software delivery projects in the financial crime prevention domain. In my opinion, it is clear that the industry is undergoing an evolution in software solutions for regulatory compliance and fraud prevention, moving away from ‘rules-engine’ technologies and towards machine learning techniques.

A key motivator for me in joining the DBS Higher Diploma in Data Analytics course was therefore to gain greater insight into this field of learning.

Another industry shift, which is obvious across multiple business domains, is the trend away from on-premises project deployments towards ‘cloud-based’ offerings.

My project proposal is therefore a topic that I believe compliments these characteristics of the FinTech world by delivering a basic cloud hosted credit card fraud detection model. The model will be built and deployed via an Azure portal, but I will create a hosted Shiny R application to access the model in real time.

# Project Scope and Objectives

The scope of this project can be summarised in the following points;

* An analysis of a Credit Card Fraud dataset (based on internal company data used for a discontinued product line).
* The creation of an optimal Credit Card Fraud Detection model, built through an Azure Machine Learning workspace.
* The deployment of a Rest Endpoint within an Azure cluster to access the model.
* A Shiny R application, hosted within a separate Internet service to invoke the deployed model in real time with ‘new’ credit card data (this is data that is excluded from the training/testing of the model in Azure).
* I also intend to use the Shiny R App to perform a graphical analysis of the Credit Card dataset.

Project objectives can be summarised as;

* The successful training and testing of a one million row dataset in Azure Machine Learning Studio, with a resultant model that demonstrate a high degree of accuracy in credit card fraud detection.
* The deployment of an endpoint in Azure for which I can write R code routines to invoke a single API call to the model from an external application.
* This ‘external application’ will be an R Shiny application written by me for purposes of demonstrating the working deployment of the model.
* The Shiny R application will be hosted on-line and be the centre piece of the project demonstration. The application will also incorporate a visual analysis of the credit card dataset to exploit the richness of R libraries to display data characteristics.

# Student’s Learning Objectives

I have determined the following personal objectives for this project;

* Credit card fraud detection through machine learning techniques is not an area with which I have worked previously. I wish to gain insight into how the process is performed.
* My company is investing heavily in Azure based services. The Machine Learning Studio provided by Azure is a technology that greatly interests me and I want to use this project to dive deeper into some of its capabilities.
* In the first CA for the *Advanced Data Analytics* module (B8IT109) I experimented with using Shiny R applications and was impressed with the options to build visuals within browser based dashboards. I would like to continue to extend my knowledge of this technology through implementing a UI for this project.

# Technical Specification of the Project

The project proposal has two major components to deliver;

* A deployed model for credit card fraud detection, built in Azure Machine Learning Studio.
* A Shiny R application to act as both a UI for the model and also to provide additional visual information on the structure of the Credit Card dataset used to build the model.

**Azure**

*The Azure Portal*

The Machine Learning Studio is accessed through the Azure portal and, obviously, requires an Azure account in order to build and deploy the ML pipelines.

Azure services are not free but I have used a free educational account, via DBS, to date. If that becomes an issue I will revert to my company account, where I have a training allowance to complete the project.

*Azure Pipelines*

Model development will be through an ‘Azure Pipeline’. Work is likely to use the guidance of automated ML model generation utilities, followed by a visual model generation process.

It is difficult to determine at the moment how much embedded R or Python code will also be used, but the Azure Machine Learning studio is very flexible in terms of SDK integration.

*Rest Endpoints*

The Azure Machine Learning Studio assists with code generation to invoke the deployed model through R, Python, or C# code. I have conducted some background research (Microsoft.com, 2020)**1** on this process but I still expect the actual implementation of the code within my Shiny App to be challenging.

**Shiny R App**

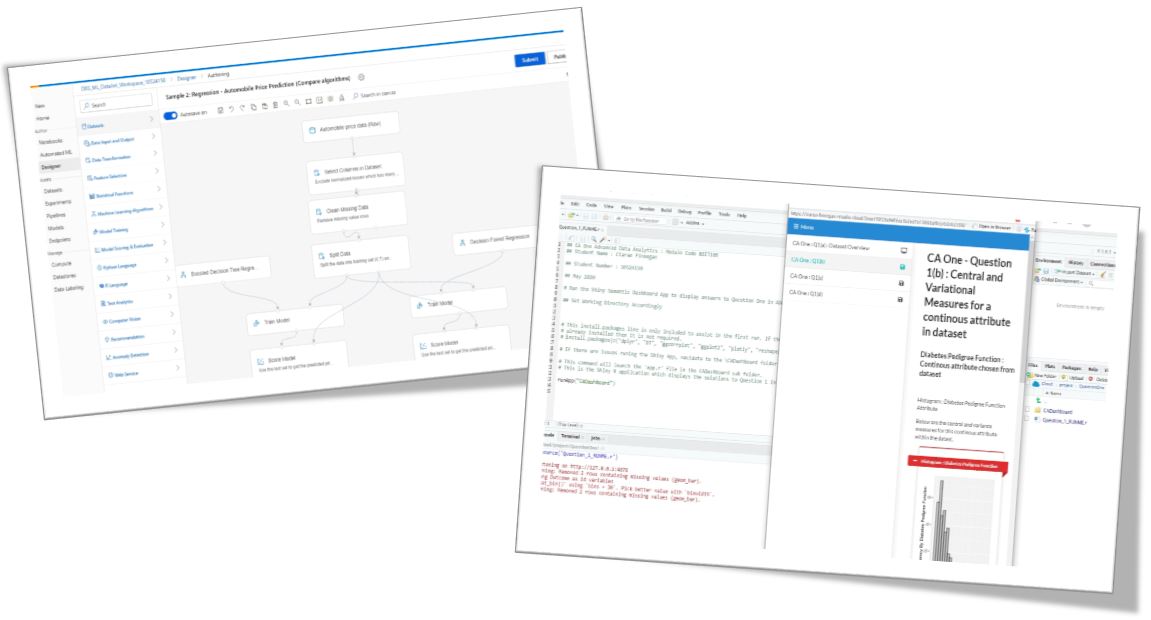
The development on my Shiny R application will be a separate exercise, built in R Studio Cloud.

It will use open source third party libraries, such as *Semantic Dashboard* (Appsilon.com, posted by Dominik Krzeminski, 2018)**2**, to create an interesting visual user interface.

The application will be hosted externally, as opposed to running on a local machine. My intention is to research further into platforms like *Shinyapps.io* to host my project UI. The basis for my approach will be to follow existing guidelines on the RStudio website (RStudio.com, posted by Andy Kipp, 2017)**3**.

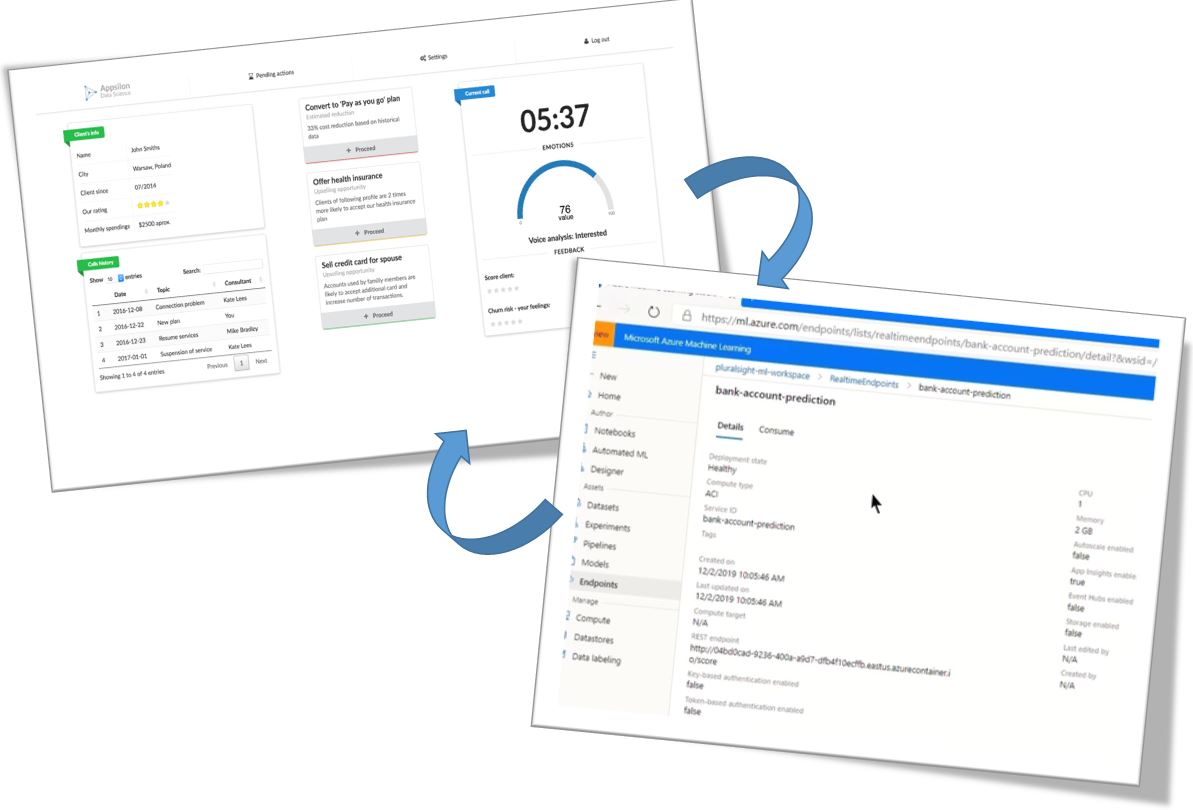
These are screenshots (Figure 1 and Figure 2) from other sample training applications but they illustrate the development and deployment environments in which I will be working.

**Fig 1: Development Environments:** *Azure Machine Learning Studio and RStudio Cloud*



The Azure Machine Learning designer screenshot references process steps documented in online Microsoft tutorials.

**Fig 2: Deployment Environments:** *Hosted Shiny R App – invoking Rest Endpoint hosted in Azure* (these are images taken from random Google Searches and are intended for illustration only)



The Shiny Dashboard here is a sample Semantic UI application, which will form a part of the basis for my design and development of the project interface (Appsilon.com, posted by Filip Stachura, 2016)**4**.

Project work will follow general practice for building models in Azure ML Studio, based on training guidelines I have researched on the learning website *Pluralsight* (Pluralsight.com, Tim Warner 2019)**5**.

# Project Plan

In the ‘real world’ any new ML project for FinTech software development is likely to managed using an AGILE framework (Wikipedia: Agile Software Development)**6**, such as Scrum or Kanban.

Although such a framework is somewhat overkill for a one-man project, I have created (and will continue to refine) a ***project backlog*** of tasks for this project. Work will be delivered in ***Sprints***, adjusted to align to the key milestone dates set by DBS for the project.

The ***Product Backlog***,representing the expected workload and iterative schedule of design, development, and documentation for my project, is shown in the image overleaf.

Rather than adopt a ‘monolith’ approach to the design and development of my project, where major activities follow in a sequence of ‘waterfall’ steps, I will follow an approach so that **each** Sprint contains the following actions;

* Design
* Development
* Test
* Documentation
* A ‘demo’ of working software.

Some of my ‘Sprints’ contain just one User Story, so I will be applying a very informal adoption of AGILE methods. However, the key point is that my plan is structured around an iterative delivery approach.

**DBS Data Analytics Project Plan 2020 (Produced using the Team Gantt online portal)7**



# Conclusion

I believe this proposal balances new areas of research within an achievable delivery target for the final project in this course.

At the same time, I also expect to draw heavily from knowledge learned during the lectures and CA work of a number of my modules within this Higher Diploma programme.

The challenges within this project work are ones that I am very much looking forward to tackling. This should be a rewarding and satisfying ends to my course work.

# References / Bibliography

1. Microsoft Online Documentation (2020), ‘Tutorial: Predict automobile price with the designer (preview)’. Available at:

<https://docs.microsoft.com/en-gb/azure/machine-learning/tutorial-designer-automobile-price-train-score>

(Accessed 4 June 2020)

1. Dominik Krzeminski (June 2018) ‘Create outstanding R Shiny dashboards with the semantic.dashboard package’. Available at:

<https://appsilon.com/create-outstanding-dashboards-with-the-new-semantic-dashboard-package/>

(Accessed 1 June 2020).

1. Andy Kipp (May 2017), ‘Shinyapps.io – Getting started’. Available at:

<https://shiny.rstudio.com/articles/shinyapps.html>

(Accessed 16 June 2020)

1. Filip Stachura (December 2016) ‘We Have Created a Package to Improve the UI of Shiny Dashboards’. Available at:

<https://appsilon.com/why-have-we-created-package-to-improve-shiny-apps-user-interface/>

(Accessed 23 June 2020)

1. Tim Warner (December 2019). ‘Microsoft Azure AI Engineer: Developing ML Pipelines in Microsoft Azure’. Available at:

<https://app.pluralsight.com/library/courses/microsoft-azure-developing-ml-pipelines/table-of-contents>

(Accessed 10 June 2020)

1. ‘Wikipedia: Agile software development’ (no date). Available at:

<https://en.wikipedia.org/wiki/Agile_software_development>

(Accessed 1 June 2020)

1. ‘Welcome to TeamGantt’ (no date). Available at:

<https://support.teamgantt.com/article/77-welcome-to-teamgantt/>

(Accessed 17 June 2020)