

Credit Card Fraud Modeling and Deployment

**Interim Report**

Higher Diploma in Science in Data Analytics

Word Count: 1500

Author: Ciaran Finnegan / 10524150

E-mail: [10524150@mydbs.ie](mailto:10524150@mydbs.ie) / [ciaran@feefinnegan.com](mailto:ciaran@feefinnegan.com)

Supervisor: Dr Shahram Azizi Sazi

Report Submission Date: 14th August 2020

**Acknowledgments**

A brief paragraph or two acknowledging professional advice and help in submitting the report.

# Contents

[Contents 2](#_Toc47558831)

[1. Introduction 3](#_Toc47558832)

[2. Background 4](#_Toc47558833)

[3. Requirements: Specification and Design 5](#_Toc47558834)

[3.1. Project Mission Statement 6](#_Toc47558835)

[3.2. Project Requirements 7](#_Toc47558836)

[3.3. Project Design 8](#_Toc47558837)

[3.3.1. User Story Structure 9](#_Toc47558842)

[3.3.2. User Story Roles 10](#_Toc47558843)

[3.3.3. User Story 4 – Initial Data Modelling and User Interface ‘Shell’. 11](#_Toc47558844)

[3.3.4. User Story 5 – Shiny App Prototype 13](#_Toc47558845)

[3.3.5. User Story 6 – Hosted Prototype 15](#_Toc47558846)

[3.3.6. User Story 7 – Feature Enhancements 1 17](#_Toc47558847)

[3.3.7. User Story 8 – Feature Enhancements 2 19](#_Toc47558848)

[3.3.8. User Story 9 – Final Project Refinements 21](#_Toc47558849)

[4. Project Testing and Evaluation 22](#_Toc47558850)

[4.1. Evaluation of User Stories 22](#_Toc47558851)

[4.2. Project Testing 23](#_Toc47558852)

[5. Demonstration of Progress 24](#_Toc47558853)

[5.1. Credit Card Fraud Application: User Guide 24](#_Toc47558854)

[5.2. Project Plan 2020 : Current Status 0](#_Toc47558855)

[6. Project Re-Design Considerations 0](#_Toc47558856)

[6.1. Initial Proposal 0](#_Toc47558857)

[6.2. Possible Future Design/Deployment Considerations 0](#_Toc47558858)

[7. Appendices 2](#_Toc47558859)

[7.1. Azure Generated Code Segments 2](#_Toc47558860)

[7.2. Shiny R Application Code Files 3](#_Toc47558861)

[7.3. Azure Machine Learning Classic Studio Experiments 4](#_Toc47558862)

[8. References / Bibliography 5](#_Toc47558863)

# Introduction

The introduction provides the reader with an overview of the project.

A good introduction will inform the reader what the project is about without assuming any specialist knowledge and without detail that may obscure the overview. The reader is assumed to be knowledgeable but not necessarily an expert in the field of the project.

The introduction should anticipate and combine main points described in more detail in the rest of the report.

The Introduction contains:

• the aims of the project;

• the scope of the project;

• the approach used in carrying out the project;

• assumptions, if any, on which the work is based.

Strengthen the aim/objective -

# Background

The purpose of the background is to provide the reader with the information that they may not know but which they will need in order to fully understand and appreciate the rest of the report.

The following is an indicative list of items that should be included in the background section:

* + the context of the project;
  + the anticipated benefits of the system; typical users of the project product;
  + any theory associated with the project;
  + the analytics methods/theories/algorithms used; any relevant/similar existing software/hardware

# Requirements: Specification and Design

The requirements specification must include the following:

Project/Business requirements

* What is the business/project need or problem? What business questions do we need to answer?
* Information requirements
* What data is necessary to answer those questions? Functional requirements
* How do we need to use the resulting information to answer those questions? Detailed report / usage req.
* Detailed layout etc. Other requirements
* All other non-functional requirements, etc.

The Design should include the high level design to meet the requirements.

In brief:

* Requirements Specification
* Design: The top-level details to meet the specification.
* Tools and Techniques: Evaluation and research to apply theories, models, methodologies and tools for the data analytics project.
* Use of diagrams, such as entity-relationship and UML diagrams

Strengthen – project objective Machine learning application…

Review ML papers for research… and related works…Chapter 2…

Refer to the paper and which model is best for CC classification…

Chapter 3…methodology on ML approach – which algorithm…and why picked them…

Add..Data understanding…CRISP-DM..framework and decision making process..

Read up on Fraud classification – find papers…

Look at Boosting methods…to select the most accurate model…

## Project Mission Statement

What is the goal of this project? What business objective does it attempt to achieve?

This project intends to deliver a working system to allow a user assess a given single credit card transaction, and obtain a prediction as to whether the transaction is likely to be fraudulent.

<image?>

## Project Requirements

In order to achieve the objectives of the project mission statement, the following requirements must be met;

* A predictive model for Credit Card fraud detection must be built using an effective Machine Learning workflow process, which produces results that are as accurate as reasonably possible.
* A dataset is provided with sufficient volume and richness of attributes to allow for appropriate data preparation and modelling to be executed.
* All development and system execution is conducted on cloud based platforms. There is no dependency on local PC libraries or IDEs, and so on;
* All model development and deployment is conducted through the cloud based Azure Machine Learning Studio platform.
* The resultant model is accessible by a separate R Shiny application, which is also hosted online.
* The end user will work with the R Shiny application interface and chose a given single credit card fraud transaction to investigate. A real-time prediction of the likelihood of fraud will be provided to the use on screen.
* The R Shiny application can access the source dataset to provide data visualisations as a peripheral service to the end user.

Non-functional requirements for the system can be summarized as;

* Fully cloud based development and deployment, as mentioned above.
* The response time for real-time fraud prediction is within a 2 – 5 second timeframe.

## Project Design

My system design and implementation approach follows general AGILE methodologies, which have been adapted to be practical for a project of this type.

The essence of my implementation approach is an iterative design, deploy, and assess model.

One key tenant of the AGILE Manifesto is ‘*Working software over comprehensive documentation*’.

Therefore my project software is designed, coded, tested, and deployed in small discrete ‘User Stories’. I assess a ‘demonstration’, which is admittedly just to myself, at the end of one or two User Stories and then take the key learnings and observations into the next User Story.

(In the ‘real world’ I would be following a more traditional SCRUM approach of multiple User Stories within a pre-defined ‘SPRINT’, but I am being flexible with my interpretation of AGILE frameworks to fit with a one-person project of this type).

Section 5 of this document describes project progress to date but also concludes with a breakdown of my project plan into the constituent User Stories.

Although each User Story is refined by the preceding one it is still possible to great a general ‘roadmap’ for this project, based on the current expectations for each User Story.

For the purpose of this Interim Report I have provided the following details on the current content of the User Stories for this project.



### User Story Structure

The User Stories for this project follow a consistent format.

Each User Story described a role (or ‘actor’) and how they interface with the application within this User Story.

As each User Story iterates through the project development lifecycle the roles and actions adapt.

Each User Story has a ‘goal’. This is measured by the ‘Acceptance Criteria’ for each role, which is assessed during the demonstration of working software at the end of the completion of each User Story.

**Title:**

**Priority:**

**Estimate:**

**As a** *<type of user>*

**I want to** <*perform some task>*

**so that I can** <*achieve some goal>*

**Acceptance criteria**

**Given** *<some context>*

**When** <*some action is carried out>*

**Then** <*a set of observable outcomes should occur>*

### User Story Roles

Most User Stories in this project reflect the requirements of different roles, or system actors, and how these roles combine to deliver the end system.

The key roles within this project application are;

* Fraud Investigator: the end use who interacts with the Shiny R application to determine if a given credit card transaction is expected to be fraudulent.
* Data Engineer: the person responsible for the set-up of the Machine Learning development environment and the subsequent deployment of a Credit Card fraud model accessible by the Fraud Investigator.
* Data Scientist: the person who follows a Machine Learning workflow to prepare the Credit Card Fraud dataset for modelling, and who implements the most efficient modelling algorithm(s).

(Obviously in this project I shall be occupying all three roles but at different times during build, test and deployment.)

### User Story 4 – Initial Data Modelling and User Interface ‘Shell’.

User Stories 1, 2, and 3 related to the submission of the original proposal report and review with project supervisor. The key project delivery milestones are captured from User Story 4 onwards.

Acceptance Criteria become increasingly high level for the later User Stories. At this point I am building a ‘Product Backlog’ of requirements that are less and less granular in the later User Stories. This is done deliberately to allow refinements to be introduced after the assessment of the ‘demo’ as each User Story is completed/delivered.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **User Role / Objective** | | |
| **USER STORY ID** | **As a** *<type of user>* | **I want to** *<perform some task>* | **so that I can** *<achieve some goal>* |
| 1.1 | Fraud Investigator | Access a basic Shiny R Application | See some elementary visualisations on the Credit Card Fraud dataset used for modelling.  See a basic screen that shows a credit card transaction invoking a Fraud prediction algorithm and producing an output. |
| 1.2 | Data Engineer | Set up an Azure Machine Learning Studio Environment. | Provide a Machine Learning development environment for the Data Scientist. |
| 1.3 | Data Engineer | Deploy a basic REST API endpoint for a Fraud prediction model. | Provide access from an external program to the Credit Card Fraud predictive model. |
| 1.4 | Data Scientist | Build a basic Fraud Prediction model with 2.5K rows of Credit Card fraud data. | Quickly validate Azure Machine Learning Studio as means to generate a model for use in the Shiny R application prototype. |
|  | **Acceptance Criteria** | | |
|  | *Given <some context>* | **When** *<some action is carried out>* | **Then** *<a set of observable outcomes should occur>* |
| 1.1 | An RStudio environment | Investigator opens Shiny R App | Basic dataset information is available.  A hard-coded example of the output result of the API calling the Fraud model is displayed. |
| 1.2 | Access to Azure Account / Workspace | Data Scientist creates a new Machine Learning Studio Azure Workspace | All the required data storage, ML Designer, and REST Endpoint tools are available. |
| 1.4 | Access to Azure Machine Learning Studio. | Data Scientist logs into Azure Machine Learning Studio. | Data Scientist can navigate interface and generate initial basic predictive Fraud model. |
| 1.3 | A basic Fraud predictive model has been developed by the Data Scientist. | The model is deployed as a Web Service in Azure. | The API to the Fraud model can be accessed externally from another system (Shiny R App). |

### User Story 5 – Shiny App Prototype

Although User Story 4 provides for a basic Shiny R App, there is relatively little interactive functionality for the user, apart from basic shiny Desktop navigation features.

User Story 5 enriches the Shiny R App user interface.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **User Role / Objective** | | |
| **USER STORY ID** | **As a** *<type of user>* | **I want to** *<perform some task>* | **so that I can** *<achieve some goal>* |
| 1.1 | Fraud Investigator | Select a given ‘new’ Credit Card transaction and assess likelihood of Fraud. | Choose and individual Credit Card transaction from a pre-loaded list and involve the predictive fraud model. |
| 1.2 | Data Engineer | Refine Shiny R app to routines to parse attributes of a user selected CC transaction. | Allow for interactive calls on API for predictive Fraud model. |
| 1.3 | Data Scientist | Refine Shiny R app to routines to display more sophisticated view of data used for modelling. | Give the Investigator (and Scientist) a better view of the source CC transactions used for modelling. |
|  | **Acceptance Criteria** | | |
|  | *Given <some context>* | **When** *<some action is carried out>* | **Then** *<a set of observable outcomes should occur>* |
| 1.1 | A pre-loaded lists of CC transactions are provided for the Investigator. | Investigator : Selects a given CC transaction | API calls returns result of predictive Fraud model : is transaction likely to be fraudulent? |
| 1.2 | Interactive selection of a CC transaction. | API call is made in code. | Parameters in API call are dynamically updated with user CC trxn selection. Involving API returns an on screen ‘score’ to show likelihood of fraud. |
| 1.3 | Access to Azure hosted dataset used for modelling. | R Shiny App launches. | Data Scientist provides routines in R to display meaningful graphical representations of CC dataset. |
| 1.4 | R Shiny App loading dataset for visualisation and invoking Fraud API. | The R Shiny App reads data from Azure datastore and REST Endpoints | Performance is acceptable (2 – 5 seconds) with no on-screen errors. |

### User Story 6 – Hosted Prototype

User Story 5 improved the richness of the application UI but it still runs as an application through RStudio Cloud. Although it is not necessarily running on a ‘local’ system, the next User Story moves the application onto a true hosted platform where it can be run independently.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **User Role / Objective** | | |
| **USER STORY ID** | **As a** *<type of user>* | **I want to** *<perform some task>* | **so that I can** *<achieve some goal>* |
| 1.1 | Fraud Investigator | Log onto a hosted location for the R Shiny App | Run the application without the need to execute the app within RStudio Cloud |
| 1.2 | Data Engineer | Deploy the application onto a ShinyIO platform | Verify connectivity to the Azure hosted data and REST Endpoint for the Fraud model from outside RStudio Cloud. |
| 1.3 | Data Scientist | Compare output of data modelling between RStudio Cloud and ShinyIO deployment | Confirm that the predictive Fraud model results are consistent, regardless of platform. |
|  | **Acceptance Criteria** | | |
|  | *Given <some context>* | **When** *<some action is carried out>* | **Then** *<a set of observable outcomes should occur>* |
| 1.1 | A ShinyIO web location provided by the Data Engineer. | Investigator : Selects the new web location for the application. | The UI experience is identical to the RStudio Cloud based Shiny R App. |
| 1.2 | Interactive selection of a CC transaction. | API call is made in code. | Results are consistent for any given CC trxn when RStudio Cloud and ShinyIO apps are compared. |
| 1.3 | ShinyIO : R Shiny App loading dataset for visualisation and invoking Fraud API. | The R Shiny App reads data from Azure datastore and REST Endpoints | Performance is acceptable (2 – 5 seconds) with no on-screen errors. |

### User Story 7 – Feature Enhancements 1

Following AGILE Manifesto principles, the User Stories so far have focused on a constant iterative delivery of working software, primarily through the R Shiny Application interface.

User Story 7 returns to the core Machine Learning objectives of the project and the need to refine and improve the predictive Fraud model, which will be used in the final version of the project application.

The predictive model used to date in the project lifecycle was created quickly for the purposes of end-to-end testing and early deployment validations. However, it now necessary to apply more robust and thorough Machine Learning techniques to the final model that I will deploy into Production.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **User Role / Objective** | | |
| **USER STORY ID** | **As a** *<type of user>* | **I want to** *<perform some task>* | **so that I can** *<achieve some goal>* |
| 1.1 | Fraud Investigator | Select CC transactions from external files, not just preloaded data. | Predict Fraud against a more interactive range of file based CC transactions. |
| 1.2 | Data Scientist | Perform data manipulation routines on a larger CC dataset as part of re-modelling iterations. | Build more efficient and accurate predictive Fraud models, using techniques such as feature selection etc. |
| 1.3 | Data Scientist | Perform analysis of the CC dataset with multiple algorithms. | Determine the best model, with supporting documentation, for productive modelling on this CC dataset. |
|  | **Acceptance Criteria** | | |
|  | *Given <some context>* | **When** *<some action is carried out>* | **Then** *<a set of observable outcomes should occur>* |
| 1.1 | An updated ShinyIO web UI provided by the Data Engineer. | Investigator : Selects an external file of CC transactions through the UI. | The Shiny R App can read the transactions and allow for a selective fraud analysis on any given entry. |
| 1.2 | The available Machine Learning Studio Web Designer Data Tools. | A larger CC dataset is loaded. | The dataset output has been manipulated to:  - focus on the most critical features  - handle any non-numeric values  - standardise any numeric data  - any other relevant data transformation, all of which are documented |
| 1.3 | The available Machine Learning Studio Web Designer Modelling Tools. | Data Scientist: iterates through suitable algorithms | A more accurate model is generated backed up by documented comparisons with earlier algorithms. |

### User Story 8 – Feature Enhancements 2

User Story 7 refines the quality of the CC Fraud predictive model. User Story 8 redeploys this model through the Shiny R Application and adds improvements to the UI.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **User Role / Objective** | | |
| **USER STORY ID** | **As a** *<type of user>* | **I want to** *<perform some task>* | **so that I can** *<achieve some goal>* |
| 1.1 | Fraud Investigator | See enhanced data visualisations on the source CC dataset (used by the modelling process). | Obtain a more sophisticated view of the key attributes of the CC Fraud trxn data. |
| 1.2 | Fraud Investigator | See an onscreen display of the key data elements of a selected CC transaction before it is submitted to the Fraud model for assessment. | Have a better view of the key data elements being passed to the fraud predictive model. |
| 1.3 | Data Engineer | Re-deploy the revised REST Endpoint API for the revised predictive Fraud model build by the Data Scientist. | Provide the application with access to the improved fraud predictive model. |
|  | **Acceptance Criteria** | | |
|  | *Given <some context>* | **When** *<some action is carried out>* | **Then** *<a set of observable outcomes should occur>* |
| 1.1 | An updated ShinyIO web UI provided by the Data Engineer. | Investigator: Selects initial Shiny R application dashboard tab that contains data visualisations. | The Shiny R App displays an improved set of visualisations, based on recommendations from Data Scientist. |
| 1.2 | An updated ShinyIO web UI provided by the Data Engineer. | Investigator: Selects an external file of CC transactions through the UI. | The Shiny R App displays a description of the key attributes in the CC transaction (based on directions from the Data Scientist). |
| 1.3 | New model provided by the Data Scientist. | Engineer: deploys the new API routines in the R Shiny Application. | The R Shiny Application continues to return predictive outputs for the Fraud model (when compared to test results in the Azure Machine Learning Studio). |

### User Story 9 – Final Project Refinements

This User Story has the least granular set of requirements and is intended as a placeholder for refinements that arise out of the ‘demonstration’ of working software after the delivery of User Stories 7 and 8.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **User Role / Objective** | | |
| **USER STORY ID** | **As a** *<type of user>* | **I want to** *<perform some task>* | **so that I can** *<achieve some goal>* |
| 1.1 | Fraud Investigator | See revisions to the Shiny R Application UI based on feedback from US 7 and US 8. | Declare acceptance of the project as fit for ‘Production’, which is the final presentation of the project in DBS. |
| 1.2 | Data Scientist | Implement revisions to the CC predictive fraud model based on feedback from US 7 and US 8. | Declare acceptance of the predictive Fraud model as fit for ‘Production’, including all supporting documentation. |
| 1.3 | Data Engineer | Implement revisions to the fraud model deployment based on feedback from US 7 and US 8. | Declare acceptance of the deployment of the Fraud model as fit for ‘Production’, including all supporting documentation. |
|  | **Acceptance Criteria** | | |
|  | *Given <some context>* | **When** *<some action is carried out>* | **Then** *<a set of observable outcomes should occur>* |
| 1.1 | An updated ShinyIO web UI provided by the Data Engineer/Data Scientist. | Investigator: Launches the Shiny R application. | The Shiny R App runs and meets all the expected requirements for a final demonstration. |

# Project Testing and Evaluation

*It may include product verification according to the spec and unit testing, etc. Any weaknesses should be discussed.*

*A review of status of the project in terms of the proposed goals and project plan.*

## Evaluation of User Stories

The User Stories in Section 3.3 describe the functional and non-functional requirements of the application.

At the end of each User Story, I will have developed a further iteration of working software for the application.

The ‘Acceptance Criteria’ provided within each User Story provide a series of checkpoints against which I assess the success of the User Story in meeting system requirements.

If requirements are not met, then the following User Story is adjusted to incorporate any changes that must be carried forward into the project development lifecycle.

<image?>

## Project Testing

Is there a formal QA test cycle within this project? Yes, but it is not a ‘Waterfall’ activity at the end of a monolithic phase of development.

Incorporated into each User Story is the assumption that work included design, development, deployment, **and** testing.

What is the criteria, or guidance, for testing? Each User Story has an inherent ‘**Definition of Done**’ that must be met before the User Story is considered fit for purpose and the software can fairly be described as ‘working’.

The Definition of Done (DoD), which I apply to each User Story is;

* Each User Story must generate a new iteration of working software, which is a functional enhancement on the previous User Story.
* All Acceptance Criteria in the User Story are met in full.
* No critical defects remain at the end of the User Story activity.
* Minor or cosmetic defects can be carried over to the next User Story if appropriate but must be addressed in that User Story.
* Screen refreshes when data is loading, or the Fraud API is being invoked, should require no more than 2 – 5 seconds. Usability design should try and minimise data loads when the user first enters a Shiny dashboard tab.
* A documented one-page test report is produced at the end of the User Story. These test reports will be included in the Final Report for the project.

A User Story must pass the DoD. If it does not, then the next User Story cannot start. Work on the User Story is extended until the DoD are met.

<image?>

# Demonstration of Progress

Ability to illustrate and demonstrate how the artefact will work and key features it will have.

(This has to be done by means of slides, screenshots, mock--‐up, diagrams, models, sample code, prototype of working software, etc. )

## Credit Card Fraud Application: User Guide

This is a provisional guide that provides a walkthrough of the key functional aspects of the project system

<image?>

<embedded PPT>

## Project Plan 2020 : Current Status

**(Produced using the Team Gantt online portal)7**



# Project Re-Design Considerations

*Suggestions on refinements or changes in direction from original project proposal should be made here. These must be justified.*

## Initial Proposal

The User Stories described in Section 3.3 of this Interim Report are an extrapolation of the system described in the initial project proposal, as submitted on Friday June 26th.

In terms of high-level functionality, I have no plans to radically deviate from these proposals at this point in the project lifecycle.

## Possible Future Design/Deployment Considerations

Certain implementation options were left open for consideration in terms of the build of this project. All such options would be dependent on remaining timeframe and complexity.

My objective is to deliver the project as described in Section 3 of this report but I have some flexibility to adapt the project in the following areas;

* The Azure Machine Learning ***Classic*** Studio is the development environment of choice for this project. I have experimented with the more up to date Azure Machine Learning workspace with Pipeline but this is notably more complex to work within and more expensive on which to deploy. If I encounter functional limitations in the Azure Classic workspace then I may need to consider switching development environments.

<image?>

* Data manipulation and modelling will be implemented through the Azure Machine Learning ***Classic Designer***. This does allow for embedded Python and/or R routines but is essentially a graphical ‘no-code’ tool, not unlike RapidMiner or other such products. If this proves too limiting, or I have time to experiment further, I may deploy the revised predictive Fraud model through a Python Jypter Notebook. I have run a small number of trial Python Notebook experiments and, as above, the issue is mainly with complexity and cost to deploy a REST Endpoint.

<image?>

* The UI for my project is being built as a Shiny R Application using the open source Semantic Dashboard libraries. This is unlikely to change. However, should I encounter issues with a full hosting arrangement I may consider switching to a Python based web application written using the *Streamlit* libraries. This Streamlit option would avoid the complexities of considering *Django* and *Flask* but would only be considered if I encounter serious issues with the deployment of my Shiny R Application for the system UI.

<image?>

# Appendices

*Any code, specifications which should be included in the report should be included in appendices. User manual can also be included here.*

## Azure Generated Code Segments

The Azure Machine Learning Studio auto-generate codes segments in C#, Python 3.6, and R to access both Azure hosted datastores and invoke APIs to Azure hosted Rest Endpoints for deployed models.

Below are examples of code snippets which have been incorporated into my project.

This code segments reads a subset of the Credit Card dataset and uses the output to generate data visualisations in the R Shiny App.

This code segment invokes the API to pass attributes to the Fraud detection model, hosted in Azure, and return a prediction score.

## Shiny R Application Code Files

Attached is a zip file that contains the current bespoke R code that I have written for my project. (This is obviously a work in progress).

The code is modularised and broken into sub folders based on function.

<image?>

<zip file>

The final report for the project will contain a comprehensive view of the application code structure.

## Azure Machine Learning Classic Studio Experiments

The final versions of the model used to provide the predicted fraud score for ‘new’ transactions will be built during the User Stories described in Sections 3.3.6 and 3.3.8 of this document.

In order to quickly build an end-to-end Fraud detection prototype the initial modelling exercise has been relatively straightforward, and the experiment is represented in the diagram below.

<image>

The Feature Selection module was built based on external advice from a company colleague who has prior industry experience in the area of payment fraud.

User Story 7 will involve an exploration of Machine Learning techniques to perform a more formal, and automated, feature selection process.

# 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)