Configuration Manual

MSc Research Project
MSc in FinTech

Ashwani Teotia
Student ID: x17160715

School of Computing
National College of Ireland

Supervisor: Noel Cosgrave
<table>
<thead>
<tr>
<th>Student Name:</th>
<th>Ashwani Teotia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student ID:</td>
<td>x17160715</td>
</tr>
<tr>
<td>Programme:</td>
<td>MSc in FinTech</td>
</tr>
<tr>
<td>Year:</td>
<td>2019</td>
</tr>
<tr>
<td>Module:</td>
<td>MSc Research Project</td>
</tr>
<tr>
<td>Supervisor:</td>
<td>Noel Cosgrave</td>
</tr>
<tr>
<td>Submission Due Date:</td>
<td>12/08/2019</td>
</tr>
<tr>
<td>Project Title:</td>
<td>Configuration Manual</td>
</tr>
<tr>
<td>Word Count:</td>
<td>1000</td>
</tr>
<tr>
<td>Page Count:</td>
<td>8</td>
</tr>
</tbody>
</table>

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author’s written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

- Attach a completed copy of this sheet to each project (including multiple copies).
- Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).
- You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

**Office Use Only**

<table>
<thead>
<tr>
<th>Signature:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date:</td>
</tr>
<tr>
<td>Penalty Applied (if applicable):</td>
</tr>
</tbody>
</table>
1 Introduction

This is the configuration manual to assist the user to configure the artifact produced as part of the thesis titled “Prediction of Crowdfunding Project Success Probability using Machine Learning”. This manual will include the details regarding software and hardware used to realize the research artifact. This document will also contain the discussion regarding why the specific decisions are made to design and implement to assist the user for an overview. Code snippets are also provided for insights into logical implementation details. The Methodology used to complete this research project is CRISP-DM (Chapman, Clinton, Kerber, Khabaza, Reinartz, Shearer and Wirth; 2000).

2 Data Collection

Dataset 1 for this work is collected from Kaggle 2 website. Yu, Huang, Yang, Liu, Li and Tsai (2018) have used this dataset in research work. This data corresponds to reward-based crowd markets platform- Kickstarter 3. Dataset has 378,661 observations and 15 features. Further to scope the project feasibility, only the projects related to the United Kingdom were selected. Scoped dataset has 29,453 observations and 6 features. Data is downloaded from Kaggle as “CSV” format and used as it is into R-Studio to prepare and process as required for modeling.

2.1 Dataset Metadata

Collected data have following metadata, extracted using R-language with RStudio.

1 Dataset used: https://www.kaggle.com/kemical/kickstarter-projects/version/7  
2 Kaggle website: http://www.kaggle.com  
3 Kickstarter website: http://www.kickstarter.com
3 System Setup

3.1 Hardware

The research work is implemented and deployed using laptop or a desktop machine with the following description:

- Laptop Machine:
  Processor: Intel(R) Core (TM) i5 2430M CPU @ 2.40 GHz Dual core
  8GB RAM
  500GB HDD
  GPU: Intel HD Graphics 3000

- Desktop Machine: Processor: Intel(R) Xenon(R) CPU E5-1620 v4 @ 3.50 GHz Quad core
  16GB RAM
  512GB SSD
  GPU: NVIDIA Quadro M2000 12GB

The research work is started on a laptop, though with the need to tune models the desktop machine was used with a better configuration. As the data analysis grew with the tuning of models and the project data scope defined the artifacts could develop on either of the hardware.

3.2 Software

The research work is implemented and deployed either on a Windows 7 professional or windows 10 professional version. GitHub[^4] is used as a repository for the artifacts produced in this work to maintain versioning and the reviews feedback.

Zotero[^5] is used as a reference management software to maintain the references and citations used in this research work.

64-bit RStudio version 1.2.1335 is used with R-version 3.5.3 in this work to complete the data analysis to produce research artifact. An issue is observed regarding RStudio version, 64-bit RStudio has a better model run performance with respect to RAM usage.

[^4]: GitHub website: [https://github.com/](https://github.com/)
[^5]: Zotero website: [https://www.zotero.org/](https://www.zotero.org/)
Notice 32-bit RStudio version was using less RAM, once RStudio upgraded to 64-bit the performance of the model was better.

Microsoft Excel 2017 with R language is used to get insight from the crowd markets data.

4 Software/ Libraries

RStudio with R language is the main data analysis software used in this research work. There are six R-code files used in this work, more details in section 6. Following is the version information of R language used:

![R Version](image)

**Figure 2: R Version**

4.1 R library and Windows Software Specification

R:
knitr – This package is used to draw data in tabular form using kable function.
plotly – This package is used to draw plots using ggplot2.
lubridate – This package is used to convert date in text format to date format.
Caret - This package is used as a wrapper to use models used in this research work. This library is used for k-fold cross validation as well as to evaluate models performances.

Windows:
RStudio 1.2.1335
Microsoft Excel and Word 2017
TeXstudio desktop LaTeX software is used as creating the final research paper.
5 High Level Design

Figure 3: High Level Design

k-fold cross validation where k is kept as 10 is used to validate models result in this research. This technique is used to evaluate performance of the models and at the same time to avoid over and under-fitting of the models.

Seeding is used at the time of model execution, sampling and to one-hot encode variable. This is to maintain reproducibility of this research work.

6 R Code Files

R programming language is mainly used for data analysis and specifically have packages related to various statistical and machine learning tasks. Six R code files are developed in this research work as ICT artifact. R code files implemented in this research work are described as follows:

- Explore.R: This file contains the code related to the exploration of the dataset. Code in this R-file prepares data quality reports (DQR) [Kelleher, Namee and D’Arcy (2015)], does data cleanup and explores data using correlation and graphs. This file realizes data understanding and data preparation phases of CRISP-DM. Following outputs are prepared in this R-code file:

**DQR before Data processing:**

DQR for numerical features:
DQR for categorical features:

DQR after Data processing:

DQR for numerical features:
• ABT.R: This file contains code related to creating the analysis base table for modelling. Following analysis base table is produced in this R-code file; ABT for metadata feature:

Figure 7: DQR:Categorical Feature

<table>
<thead>
<tr>
<th>% Missing</th>
<th>% Unique</th>
<th>Mode1</th>
<th>Mode2</th>
<th>Mode1Freq</th>
<th>Mode2Freq</th>
<th>Mode1Percent</th>
<th>Mode2Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
<td>Film &amp; Video</td>
<td>Games</td>
<td>5216</td>
<td>3245</td>
<td>17.70957</td>
<td>11.01755</td>
</tr>
</tbody>
</table>

Figure 8: Analysis Base Table Metadata

• k-NN.R: This file contains code related to tuning the k-NN model. Following is the code snippet used to perform model tuning:

```r
set.seed(457)
training.Control <- trainControl(method = "cv", number = 10)
model_Fit <- train(status ~ goal + backers + duration + prj_name_length,
data = training, method = "knn",
trControl=training.Control,
preProcess = c("center", "scale"),
tuneLength=30,
na.action=na.exclude)
```

• RF.R: This file contains code related to tuning the random forest model. Following is the code snippet used to perform model tuning:

```r
set.seed(4541)

# Random Search
control <- trainControl(method="cv", number=10, search="random")
mtry <- sqrt(ncol(Training))
rf_random <- train(status~., data=Training, method="rf",
metric="metric", tuneLength=15, trControl=control)
print(rf_random)
plot(rf_random)
set.seed(4541)
control <- trainControl(method="cv", number=10, search="grid")
tunegrid <- expand.grid(.mtry=c(12:19))
rf_gridsearch <- train(status~., data=Training, method="rf",
metric="metric", tuneGrid=tunegrid, trControl=control)
```
• SVM.R: This file contains code related to tuning the support vector machine model. Following is the code snippet used to perform model tuning;

```r
set.seed(4581)
trctrl <- trainControl(method = "cv", number = 10, classProbs = TRUE)

# svm_Linear <- train(status ~ ., data = Training, method = "svmLinear",
#                      trControl = trctrl,
#                      preProcess = c("center", "scale"),
#                      tuneLength = 10)

# grid <- expand.grid(C = c(0, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 5))
# svm_Linear_Grid <- train(status ~ ., data = Training, method = "svmLinear",
#                          trControl = trctrl,
#                          preProcess = c("center", "scale"),
#                          tuneGrid = grid,
#                          tuneLength = 10)

svm_Radial <- train(status ~ ., data = Training, method = "svmRadial",
                     trControl = trctrl,
                     preProcess = c("center", "scale"),
                     tuneLength = 10)

grid_radial <- expand.grid(sigma = c(0, 0.01, 0.02, 0.025, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.25, 0.5, 0.75, 0.9),
                           C = c(120, 123, 125, 128, 130, 132, 135))

svm_Radial_Grid <- train(status ~ ., data = Training, method = "svmRadial",
                          trControl = trctrl,
                          preProcess = c("center", "scale"),
                          tuneGrid = grid_radial,
                          tuneLength = 10)
```

• Model.R: This file contains code to include all the models with hyperparameter tuned. This file also contains implementation of sorting smoothing method. Code related to performance matrices of the model accuracy is also included in this file. Further this file contains linear regression code to calculate predictive accuracy of the probability, generated as the output from the models.
```r
probability.failed <- prediction.probability[,1]
prediction.success <- prediction.probability[,2]

d.f <- data.frame(probability.success, Y=predictedBool)
d.f <- d.f[order(probability.success),]

names(d.f)[1] <- "V1predprob"
names(d.f)[2] <- "Yclass"

smoothing.valid.range <- (1: dim(d.f)[1])

for (probability.single in 1:dim(d.f)[1]) {
  smoothing.required.range <- (probability.single-50): (probability.single+50)
  smoothing.selected.range <- smoothing.valid.range[(smoothing.valid.range
%in% smoothing.required.range)]
  d.f[probability.single, "V2"] <-
    sum(d.f$Yclass[smoothing.selected.range]) / (2*50 +1)
}

plot( d.f[,"V1predprob"], d.f[,"V2"], main="k-NN",
xlab="Predicted probability", ylab="Estimated Actual probability", pch=1)

abline(lm(d.f[,"V2"]~d.f[,"V1predprob"]), col="blue",lty=1) # regression
line (y~x)
lines(lowess(d.f[,"V1predprob"],d.f[,"V2"]), col="red", lty=1) # lowess
line (x,y)

mod <- lm(d.f[,"V2"]~d.f[,"V1predprob"])
ex.cs1 <- expression(Regression, Lowess) # 2 ways
#utils::str(legend(.01, .95, ex.cs1, lty = 1:1, plot = FALSE)) # adj y !
legend(.01, .95, ex.cs1, lty = c(1,1), col=c("blue","red"))

summary(mod)
```

References

Chapman, P., Clinton, J. M., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. R. H.

for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies, The
MIT Press.

funding project success with deep learning, 2018 IEEE 15th International Conference
on e-Business Engineering (ICEBE), pp. 1–8.