SCDT64 – AI and Machine Learning

CW1: Practical and Presentation

Finley Edwards (EDW17006550)

2024

Contents

[Introduction 2](#_Toc161183471)

[Comparison of Selected Tools and Techniques 3](#_Toc161183472)

[CRISP-DM Implementation 4](#_Toc161183473)

[Data Understanding 4](#_Toc161183474)

[Data Preparation 8](#_Toc161183475)

[Data Modelling 8](#_Toc161183476)

[Data Evaluation 8](#_Toc161183477)

[Deployment 9](#_Toc161183478)

[Tools and Techniques Evaluation 10](#_Toc161183479)

[CRISP-DM Evaluation 11](#_Toc161183480)

[References 12](#_Toc161183481)

[Appendices 13](#_Toc161183482)

[Appendix A 13](#_Toc161183483)

[Figure 1 13](#_Toc161183484)

[Figure 2 13](#_Toc161183485)

[Figure 3 13](#_Toc161183486)

[Appendix B 14](#_Toc161183487)

[Figure 1 14](#_Toc161183488)

[Figure 2 14](#_Toc161183489)

# Introduction

# Comparison of Selected Tools and Techniques

One of the most commonly used tools used for machine learning classification is Python. There are many advantages to using Python for machine learning including its wide range of useful and extensive libraries including Scikit-Learn, Pickle, Pandas, and Flask. These libraries each serve a different purpose.

Flask will be the chosen tool to create the application for deploying the model. Flask is a Python library used to help design and create web applications. Flask is a lightweight framework designed with ease of use in mind and is therefore better suited to small, lightweight applications (Deery, 2022). When testing the performance ASP.NET Core, Express.JS, and Flask, Jonsson (2022) found that ASP.NET Core used the least amount of CPU, and memory usage while also performing the fastest. Flask however used the most CPU and memory usage, while also performing the slowest. Despite the slower performance, Flask will be suitable for this type of application due to its smaller scale, meaning it will not need as many resources.

Python is the selected programming language for developing the machine learning algorithm and Flask application in. When comparing performance between Python and ML.NET, another popular tool used for machine learning classification, Torre (2019) states that ML.NET will usually provide better performance when running predictions with less time needing to be spent on optimising the code when deploying the machine learning models. Maryliya (2024) counters this by stating that the performance benefits ML.NET has over Python may not be significant enough to outweigh the benefits of using Python and in most cases will not lead to a significant difference in performance. This will be especially relevant in this instance, since the dataset is relatively small and therefore the time taken to perform the necessary stages of the CRISP-DM model on the data set will not result in a significant decrease in performance when using Python. An advantage Python has that

The chosen tool for saving the created model is Pickle. Pickle is a python library that can export each model as a PKL file which can be used by the flask application to predict new data with. Joblib is another tool for Python that serves a similar purpose and is included in the scikit-learn library. will (Raghuvanshi, 2022)

# CRISP-DM Implementation

## Data Understanding

A screenshot of a computer

Description automatically generated

Figure i

Figure i shows the results of the built in pandas functions ‘head’ and ‘info’ being used on the dataset to provide basic information about its features and structure.

A chart of red and blue squares

Description automatically generated

Figure ii

Figure ii shows a created to show the correlation between the different characteristics of red wine. The code to create this graph can be found in Appendix A, Figure 1.

A screenshot of a chart

Description automatically generated

Figure iii

Figure iii shows a created to show the correlation between the different characteristics of white wine. The code to create this graph can be found in Appendix A, Figure 2.

A screenshot of a graph

Description automatically generated

Figure iv

Figure iv show a pair plot created to show the correlation between different variables for both wine types. The code used to create this can be found in The code to create this graph can be found in Appendix A, Figure 3.

## Data Preparation

As shown in Appendix B, Figure 1, The data set has been cleaned by removing all any duplicated entries and removing any rows that contain null values. This will improve the accuracy and performance of the models by ensuring they are only using reliable and correct data.

Appendix B, Figure 2 displays code used to Prepare the data for the modelling process. This code separates the features and target variable, then splits them again into training and testing data, finally it normalises the data using a standard scalar.

## Data Modelling

For determining the type of wine using new data, three different types of classification models were developed and evaluated agains each other. The chosen classification models are a k-nearest neighbors (KNN) model with the number of neighbors being set to 5, a logistic regression (LR) model, and a support vector machine (SVM) model.

The code used to create these models can be found in:

* Appendix C, Figure 1 KNN Classification Model
* Appendix C, Figure 2 LR Classification Model
* Appendix C, Figure 3 SVM Classification Model

## Data Evaluation

Evaluating the results of the KNN model as shown in Appendix D, Figure 1 reveals that the model has an accuracy rating of 99.25%. This can be seen in the confusion matrix which reveals that ‘red’ was correctly classified 288 with only 1 false negative, and white being correctly classified 768 times with only 7 false positives. The classification report reveals that the average F1-score for both ‘red’ and ‘white’ classes is 99%.

Evaluating the results of the LR classification model as shown in Appendix D, Figure 2 shows the model received a slightly higher accuracy rating of 99.53%. This model had one more false negative on the confusion matrix with 287 true positives and 2 false positives, however white was correctly classified more then KNN classification with it being correctly identified 772 times with only 3 false positives. The F1 score results for LR classification were 99% for red wine and 100% for white wine.

The SVM classification evaluation provided the overall best results out of all three models used. This model has an accuracy of 99.62%. The confusion matrix results listed only 2 false positive and 2 false negatives. The F1 score results listed were the same as the LR classification results with a 99% score for red wine and a 100% score for white wine. The results can be found in Appendix D, Figure 3.

## Deployment

A screenshot of a computer

Description automatically generated

*Figure i*

Deployment was achieved through using two python libraries; Pickle and Flask. Pickle was used to export the selected model as a PKL file, which can then be used to make predictions with new data. A flask application was then created which will provide a form to the user which can be used to enter in new data. When clicking the submit button, the Flask application will use the exported model file to predict the type of wine using the data the user has entered, as shown in Figure i.

Code can be found in:

* Appendix E, Figure 1
* Appendix E, Figure 2
* Appendix E, Figure 3

# Tools and Techniques Evaluation

Python has a variety of advantages that make it suitable for machine learning. Python features many different libraries that make handling, transforming, and predicting data easy such as Scikit-learn, Pandas, Numpy, and more. Ryabtsev (2024) states that Python is a very flexible option for coding, with it being completely platform independent and allowing for developers to quickly create and run code without the need for it to be recompiled due to it being an interpreted language. Contrasting this, Limón (2022) states that Interpreted languages like Python typically have slower execution speeds when compared to other compiled languages, since they need to translate the source code at run time each time every time the code is executed.

Sci-kit Learn was used to perform the data modelling since there are many different advantages to using it. Sci-kit Learn has a wide array of supported model types for classification, regression, dimension reduction, and more, with extensive support and documentation for each of its supported models (Data Basecamp, 2022).

Data Basecamp, (2022). *What is Scikit-Learn? | Data Basecamp*. [online] Available at: https://databasecamp.de/en/python-coding/scikit-learns.

‌

‌

# CRISP-DM Evaluation

CRISP-DM is a tiered, industry standard, iterative model used for data mining and data analytics (Hotz, 2023). As a result of being iterative, the model will allow for continuous development, to ensure the final product meets all requirements and is ready to be deployed. CRISP-DM has a wide variety of benefits when used for machine learning projects such as its naturally easy to follow and implement design. Saltz (2021) states that due to the simple design of the model, data scientists will naturally tend to follow a process similar to CRISP-DM if not provided with management directions, making it easy to follow and implement into projects. Mithun (2021) contrasts this by stating that due to the model’s simplicity, it is missing key features that should be included with larger projects, such as including hypothesis farming between the modelling and data preparation phases in order to evaluate any ideas put forward by the client. Comparing CRISP-DM to Agile

Hotz, N. (2023). *What is CRISP DM?* [online] Data Science Project Management. Available at: https://www.datascience-pm.com/crisp-dm-2/.

‌ Saltz, J.S. (2021). CRISP-DM for Data Science: Strengths, Weaknesses and Potential Next Steps. *2021 IEEE International Conference on Big Data (Big Data)*. doi:https://doi.org/10.1109/bigdata52589.2021.9671634.

Mithun, D.J. (2021). *Why CRISP DM is dead?* [online] Available at: https://race.reva.edu.in/race-lab/why-crisp-dm-is-dead/.

‌ <https://krisolis.ie/crisp-dm-vs-agile-the-face-off/>

<https://agilethought.com/blogs/scaling-data-science-use-crisp-dm-agile/>

‌

# References

Torre, C.D. la (2019). What is ML.NET 1.0 - Machine Learning for .NET. [online] Cesar de la Torre. Available at: https://devblogs.microsoft.com/cesardelatorre/what-is-ml-net-1-0-machine-learning-for-net/ [Accessed 11 Mar. 2024].

Maryliya, M.J. (2024). .NET Vs. Python For Machine Learning Development . [online] Zenkins. Available at: https://zenkins.com/updates/net-vs-python-for-machine-learning-development/ [Accessed 11 Mar. 2024].

‌Deery, M. (2022). The Flask Web Framework: A Beginner’s Guide. [online] careerfoundry.com. Available at: https://careerfoundry.com/en/blog/web-development/what-is-flask/.

‌Jonsson, M., Qvarnström, E., Lindell, R. and Gustafsson, J. (2022). A PERFORMANCE COMPARISON ON REST- APIS IN EXPRESS.JS, FLASK AND ASP.NET CORE. [online] Available at: https://www.diva-portal.org/smash/get/diva2:1669487/FULLTEXT01.pdf.

‌Ryabstev, A. (2024). 8 Reasons Why Python is Good for Artificial Intelligence and Machine Learning. [online] Available at: <https://djangostars.com/blog/why-python-is-good-for-artificial-intelligence-and-machine-learning/>.

Limón, B. (2022). Compiled vs interpreted language: Basics for beginning devs. [online] Available at: https://www.educative.io/blog/compiled-vs-interpreted-language.

Raghuvanshi, A. (2023). Pickle or Joblib for saving ML Models? [online] Available at: https://technorep.in/2023/04/06/pickle-or-joblib-for-saving-ml-models/ [Accessed 12 Mar. 2024].

Patel, U. Why Python is the best for Artificial Intelligence and Machine Learning. [online] www.tristatetechnology.com. Available at: <https://www.tristatetechnology.com/blog/why-is-python-the-best-for-artificial-intelligence-and-machine-learning>.

Joblib. (n.d.). Joblib: running Python functions as pipeline jobs — joblib 1.3.2 documentation. [online] Available at: https://joblib.readthedocs.io/en/stable/.

‌

‌

# Appendices

## Appendix A

### Figure 1

A screen shot of a computer program

Description automatically generated

### Figure 2

A screen shot of a computer code

Description automatically generated

### Figure 3

A black screen with colorful text

Description automatically generated

## Appendix B

### Figure 1

A screen shot of a computer code

Description automatically generated

### Figure 2

A computer screen shot of a program code

Description automatically generated

## Appendix C

### Figure 1

A screenshot of a computer program

Description automatically generated

### Figure 2

A screenshot of a computer program

Description automatically generated

### Figure 3

A screenshot of a computer program

Description automatically generated

## Appendix D

### Figure 1 KNN Evaluation

A screenshot of a computer program

Description automatically generated

### Figure 2 LR Evaluation

A screenshot of a computer program

Description automatically generated

### Figure 3 SVM Evaluation

A screenshot of a computer program

Description automatically generated

## Appendix E

### Figure 1 Flask app.py

A screen shot of a computer program

Description automatically generated

### Figure 2 Flask index.html

A screen shot of a computer program

Description automatically generated