**COURSEWORK SUBMISSION FORM**

|  |  |  |  |
| --- | --- | --- | --- |
| **STUDENT USE** | | **STAFF USE** | |
| Module Name | Distributed Systems and Cloud Computing | First Marker’s  (acts as signature) |  |
| Module Code | 6BUIS014C-n | Second Marker’s  (acts as signature) |  |
| Lecturer Name | Isomiddin Abdunabiev | Agreed Mark |  |
| UoW Student IDs |  | **For Registrar’s office use only (hard copy submission)** | |
| WIUT Student IDs | 00010023 |
| Deadline Date | 07.11.2023 |
| Assignment Type | Individual |
| Word Count | 1287 |

**SUBMISSION INSTRUCTIONS**

**COURSEWORKS *must* be submitted in *both* HARD COPY (to the Registrar’s Office) *and* ELECTRONIC unless instructed otherwise.**

For hardcopy submission instructions refer to: <http://intranet.wiut.uz/Shared%20Documents/Forms/AllItems.aspx> - Coursework hard copy submission instructions.doc

For online submission instructions refer to: <http://intranet.wiut.uz/Shared%20Documents/Forms/AllItems.aspx> - Coursework online submission instructions.doc

|  |
| --- |
| **MARKERS FEEDBACK (Continued on the next page)** |
|  |

Table of Contents

[Introduction 2](#_Toc150107005)

[MVC Frontend 3](#_Toc150107006)

[Structured Models in MVC 3](#_Toc150107007)

[MVC with CRUD Requests to API 4](#_Toc150107008)

[Implementation and Comparison of Machine Learning Algorithms 5](#_Toc150107009)

[Confusion Matrix 5](#_Toc150107010)

[Classification Metrics 6](#_Toc150107011)

[Conclusion 7](#_Toc150107012)

[References 8](#_Toc150107013)

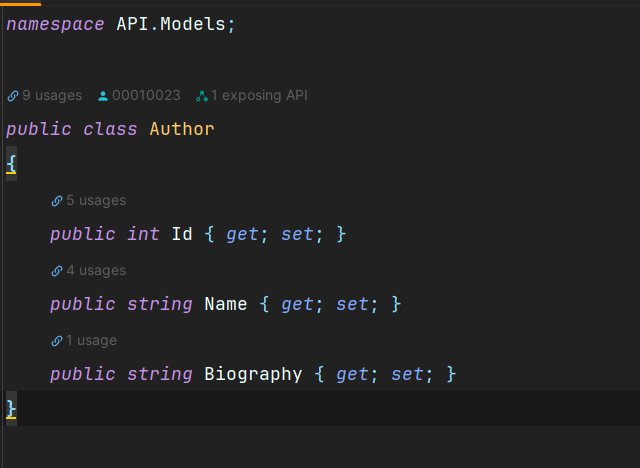
# Introduction

This coursework aims to present skills learned on seminars and workshops that introduced students how to use AWS services, writing server and frontend microservices and how to deploy, manage and maintain pipelines. This report will be following sections stated on requirements saving the order to meet the requirements for report.

# MVC Frontend

### Structured Models in MVC

Application has only 2 models which are Post and Author. Author model has fields as Unique Id, Name and Biography.



Posts model is relational to Author and has fields like Unique Id, Title, Content, AuthorId which is relational field to Author model and binds to Author of Post and Author which is not stored on database but initialized on instance initialization.

A screen shot of a computer program

Description automatically generated

### MVC with CRUD Requests to API

Following the Models stated above, corresponding controllers has been created on server side microservice. 2 controllers were created which are AuthorController and PostController. Unlik ordinary CRUD operations, PostController and AuthorController has been modified to suit a few logics.

Starting with AuthorController, following endpoints were created:

* Get: Get all

After the data cleaning process, the next step involved dividing the columns into dependent and independent variables. Given the investigation's aim to determine the mental health status of the individuals, the Mental Illness column was selected as the dependent variable, with the remaining columns serving as independent variables. The train\_test\_split method was imported, facilitating the division of the dataset into training and testing sets. Lastly, the dataset was standardized using the StandardScaler() method.

# Implementation and Comparison of Machine Learning Algorithms

The dataset was exclusively subjected to supervised machine learning (ML) algorithms. Specifically, four ML algorithms were selected to analyze the data: Logistic Regression, Decision Tree, K-Nearest Neighbors, and Naïve Bayes. Interestingly, these four ML algorithms displayed both similarities and differences in their predictions when measured using two metrics: the Confusion Matrix and Classification Metrics.

For comparing ML algorithms using the Confusion Matrix, four components were considered: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). As per West (2020), these four elements represent test outcomes, where TP and FN indicate a perfect match between the actual data and prediction, while TN and FP denote a discrepancy between the actual data and prediction.

Concerning Classification Metrics, three criteria will be used to compare the ML algorithms: Precision, Recall, and Accuracy. To clarify how these metrics operate, Vujocovij (2021) provides formulas for calculating the results. For instance, Precision can be calculated by dividing the number of correct positive predictions (TP) by the total number of positive predictions (FP + TP). He indicates that the optimal score for each criterion is 1.0, with the least desirable score being 0.0.

Comparisons are illustrated in the subsequent two tables, where predictions are based on 4132 (33%) randomly selected data samples to determine whether individuals were mentally ill or not.

## Confusion Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | TP | TN | FP | FN |
| Logistic Regression | 0 | 878 | 2 | 3243 |
| Decision Tree | 217 | 661 | 667 | 2578 |
| K-Nearest Neighbors | 108 | 770 | 217 | 3028 |
| Naïve Bayes | 54 | 824 | 160 | 3085 |

## Classification Metrics

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithms** | Precision | Recall | Accuracy |
| Logistic Regression | False – 0.79  True – 0.00 | False – 0.1  True – 0.0 | 0.79 |
| Decision Tree | False – 0.80  True – 0.25 | False – 0.79  True – 0.25 | 0.68 |
| K-Nearest Neighbors | False – 0.80  True – 0.33 | False – 0.93  True – 0.12 | 0.76 |
| 0.Naïve Bayes | False – 0.79  True – 0.25 | False – 0.95  True – 0.06 | 0.76 |

# Conclusion

Interpreting the above tables allows us to draw several conclusions. In the context of the confusion matrix, among 4132 randomly selected individuals, the Logistic Regression ML algorithm predicted that there were no mentally ill individuals (TP) and that 3243 individuals were not mentally ill (FN). However, there was a discrepancy with 878 individuals who were indeed mentally ill but were predicted by the ML algorithm to be healthy (TN). Nevertheless, the algorithm performed well in predicting individuals who were not mentally ill, with only 2 healthy individuals misclassified (FP).

On the other hand, the Decision Tree ML algorithm's predictions were considerably less accurate. This was demonstrated by over 1200 individuals, both mentally ill and healthy, for whom the predictions did not align with the actual results. As for the K-Nearest Neighbors and Naïve Bayes ML algorithms, these displayed somewhat similar prediction performances but were not flawless. Both algorithms inaccurately predicted over 900 mentally ill and healthy individuals.

The Classification Metrics table serves as a translation of results for all four ML algorithms. The numbers within each criterion provide the accurately predicted statistics for individuals who were mentally ill (True) and those who were not (False). Overall, the Logistic Regression algorithm appears to be the most effective, with a predicted accuracy rate of 79% from the 33% randomly selected individuals. Conversely, the least effective ML algorithm appears to be the Decision Tree, with only a 68% accuracy rate. Interestingly, both the K-Nearest Neighbors and Naïve Bayes algorithms produced identical prediction results – 76%, which is a favorable outcome compared to Logistic Regression.

# References

Vujokovic, Z. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*. Available from <https://thesai.org/Downloads/Volume12No6/Paper_70-Classification_Model_Evaluation_Metrics.pdf> [Accessed 30 May, 2023].

West, R. (2020). Understanding the Accuracy of Diagnostic and Serology Tests: Sensitivity and Specificity. *Johns Hopkins Bloomberg School of Public Health*. Available from <https://www.centerforhealthsecurity.org/resources/COVID-19/COVID-19-fact-sheets/201207-sensitivity-specificty-factsheet.pdf> [Accessed 30 May, 2023].