UNIVERSITY OF THE FREE STATE

BLOEMFONTEIN CAMPUS

CSIS3764

DEPARTMENT: COMPUTER SCIENCE AND INFORMATICS

CONTACT NUMBER: 051 401 2929

SEMESTER TEST 1
02 September 2022

ASSESSOR: 1. Mnr. W.S.J. Marais
MODERATOR: 1. Mev. S.E.S. Campher

TIME: 120 min **MARKS:** 80

INSTRUCTIONS:

- Log into Blackboard and go to CSIS3764 -> Assessments -> Semester Test 1.

Question 1 [20 Marks - 20 Minutes]

1.1 Give a detailed definition of Data Science. (6)

1.2 Explain the difference between oversampling and undersampling as it pertains to the data exploratory analysis and pre-processing phase. (2)

1.3 Draw a diagram of the CRISP-DM model and discuss each stage briefly as a model for conducting Data Science projects. (10)

1.4 Briefly explain what data leakage is and how it can be prevented. (2)

[20]

Question 2 [60 Marks - 100 Minutes]

- Please log into Blackboard and go to CSIS3764 -> Assessments -> Semester Test 1.
- Download the *chess_games.csv* data file.
- Create a Jupyter Notebook that contains the Python code to import, explore, analyse, clean and pre-process the data for numerical data analysis. Call the notebook *CSIS3764_YourStudentNumber_ST1_2022*.

- The Jupyter Notebook should do the following (NB!!! - Show the changes in the data after each operation):

2.1 Import the data file into a data frame, called *chess_games*, with the following column headings: (3)

- game_id
- game_is_rated
- start_time
- end_time
- number_of_moves
- game_outcome
- winner
- time_increment
- white_piece_player_id
- white_piece_player_rating
- black_piece_player_id
- black_piece_player_rating
- all_moves_in_standard_chess_notation
- standardised_code_for_any_given_opening_moves
- opening_moves_name
- number_of_moves_in_the_opening_phase

2.2 Inspect the data by: (3)

- Displaying the first 20 records of the dataset.
- Generating a statistical summary of <u>all</u> the features.
- Displaying the data type for each column.

2.3 Determine and display the row and column count of the dataset. (1)

2.4 Determine and display the total number of elements that exist in the dataset. (1)

2.5 Determine and display the overall rating of the: (3)

- lowest rated player.
- highest rated player.

2.6 Determine and display the count (number of games) for each of the different game outcomes. (2)

2.7 Determine and display the percentage of all the games that ended in "mate". Round the percentage to 2 decimal values. (3)

2.8 Plot a pie chart to illustrate the percentages of the different game outcomes. Round the percentages to zero decimal values. (4)

2.9 What was the rating of the player that won the most games? (3)

2.10 Determine and display out of all the white piece players' losses the percentage of games that they "resigned". Round the percentage to 2 decimal values. (4)

2.11 Determine and display the average number of (full/completed) moves for games that ended in a "draw". (2)

2.12 Determine and display by means of a pie chart whether the data is balanced with regard to the number of games that were won by either black or white or neither (when there was a draw). (3)

2.13 Print the answer to this question in your Jupyter notebook: Discuss your recommendation on whether or not the chess games should be balanced so that the number of games that were won by black or white or drawn are equally represented in the data. (2)

2.14 Print the answer to this question in your Jupyter notebook: If the games were to be balanced with regard to the number of games that were won by black or white or drawn, would you recommend oversampling or undersampling? Give a reason for your answer. (2)

2.15 Check for missing values. (1)

2.16 Handle the missing values by calculating the missing values from information contained in other columns. (3)

2.17 Discard the *game_id*, *time_increment*, *all_moves_in_standard_chess_notation* and *opening_moves_name* columns. (1)

2.18 Convert the text values in the remaining dataframe to numeric values (excluding *standardised_code_for_any_given_opening_moves*): (5)

- Convert the columns that contain only two unique text values to binary values [0, 1].
- Convert the columns (excluding *standardised_code_for_any_given_opening_moves*) that contain more than two unique text values to numeric values [0, 1, 2, …], with the highest numeric value depending on the number of unique text values.

2.19 Calculate a correlation matrix and plot a heatmap to identify the features that influence each other the most. (4)

2.20 Print the answer to this question in your Jupyter notebook: Discuss the output of the correlation matrix and heatmap with regard to the features that influence each other the most. (2)

2.21 Convert the column *standardised_code_for_any_given_opening_moves* to a vector space. (3)

2.22 Ensure that all the values in the dataframe are on the same scale. (3)

2.23 The Jupyter Notebook should be able to receive the data from the *chess_games.csv* file and output the final cleaned and pre-processed data to a file called *clean_chess_games.csv*. (2)

- After you have completed Question 2, please submit your completed Jupyter Notebook on Blackboard.

[60]

End of Paper