Report

Section 1:

This report examines run production functions in One Day International (ODI) cricket matches from 1999 to 2011 using the available dataset. The purpose of the analysis is to discover the best-fit run production functions based on the variables of wickets-in-hand (w) and overs-to-go (u), applying a model given by the equation Z(u, w) = Z0(w) * [1 - exp(-Lu/Z0(w))]. The assignment involves using the sum of squared errors loss function and only looking at the first innings of data.

Data Preprocessing

Before delving into the analysis, the dataset is pre-processed to extract relevant information. The following steps are performed:

1. Load Data: The dataset is loaded from the provided CSV file using the pd.read_csv() function.
2. Clean Data: Rows with an 'Error.In.Data' value of 1 are dropped to remove erroneous data points.
3. Select Columns: Columns 'Over', 'Wickets.In.Hand', and 'Total.Runs' are selected as they are essential for the analysis.
4. Calculate Overs Remaining: A new column 'Overs.Remaining' is calculated by subtracting the 'Over' column from 50 (the total overs in an ODI match).
5. Convert Date Format: The 'Date' column is converted from the format 'dd/mm/yyyy' to 'dd-mm-yyyy'.
6. The pre-processed dataset is then ready for further analysis.

Model Implementation

A model class DLModel is defined to approximate the run production function. This class encapsulates methods to calculate predictions, loss, and save/load model parameters. The parameters 'Z0' (for each wicket count) and 'L' are initialized within the model's constructor.

Model Training and Optimization

With the preprocessed data and the DLModel class in place, the next step is to train the model using the available data. This involves finding the optimal values for the model parameters 'Z0' and 'L' that minimize the mean squared error loss.

The SciPy library is utilized for this purpose, particularly the minimize function. The calculate_loss method of the DLModel class is used as the loss function to be minimized. By providing initial guesses for the parameters, the minimize function iteratively refines the parameters until convergence is achieved.

The trained model parameters are then used to make predictions for different combinations of wickets-in-hand and overs-to-go.

Visualization of Run Production Functions

To visually interpret the performance of the trained model, a plotting function plot is implemented. This function generates a plot showing the run production functions for different scenarios of wickets-in-hand. The overs-to-go is plotted on the x-axis, and the predicted runs are plotted on the y-axis. Each line on the plot represents a different wickets-in-hand scenario, showcasing how the run production changes as the overs-to-go decreases.

Printing Model Parameters

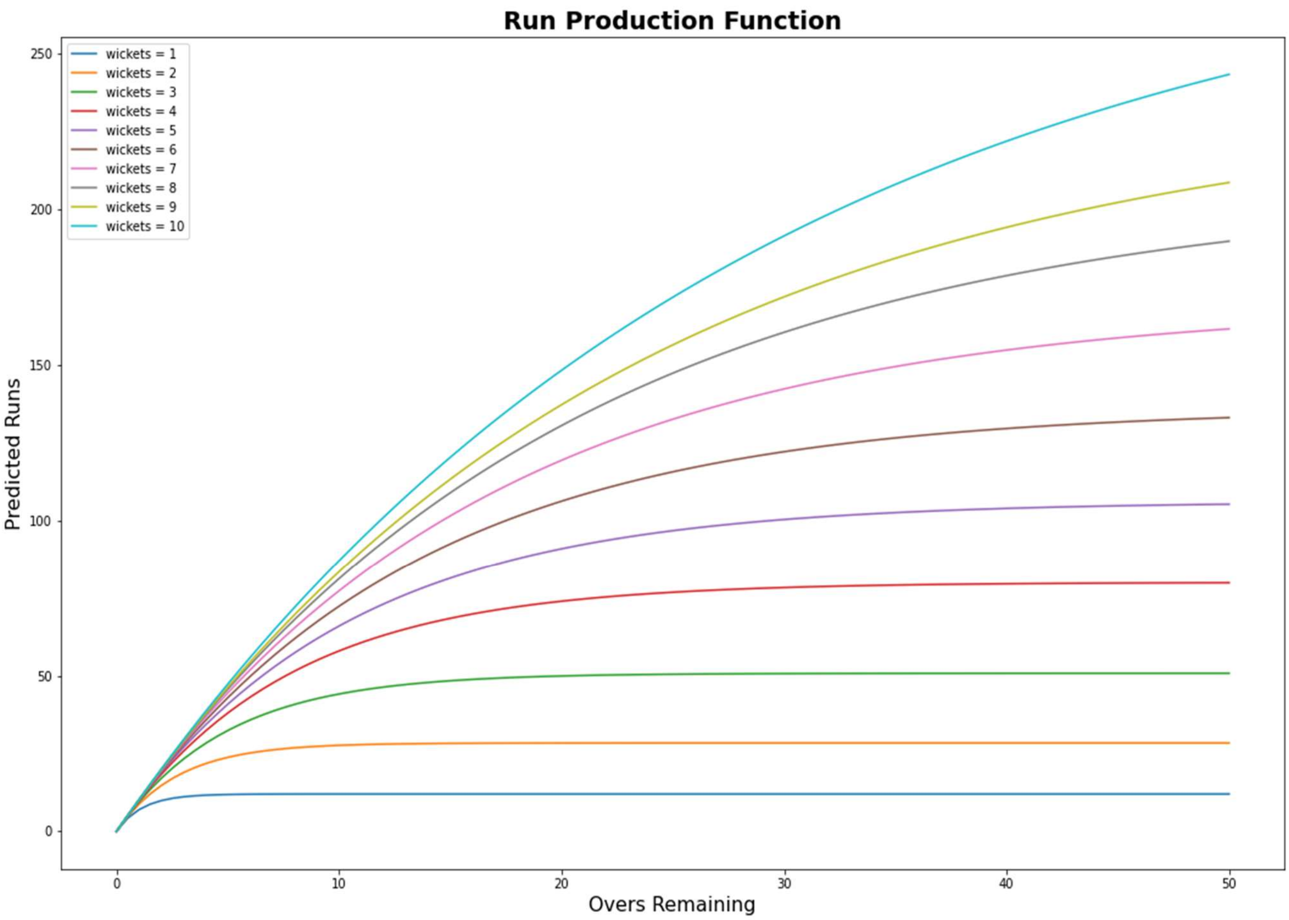
Additionally, a function print_model_params is provided to print the 11 model parameters: Z0 values for 10 wickets-in-hand scenarios and the L parameter. These parameters offer insight into the model's behaviour and its sensitivity to different wickets and overs situations.

Conclusion

Finally, using a complex model that accounts for wickets-in-hand and overs-to-go, this analysis digs into the run production functions in ODI cricket matches. Data preprocessing, model training, parameter optimization, visualization, and parameter printing are among the steps. This Assignment not only tackles a hard topic but also demonstrates the potential of data analysis and visualization approaches in the realm of sports analytics by utilizing Python modules such as Pandas, SciPy, and Matplotlib. By understanding how runs are produced under different conditions, the insights gathered from this study can help with strategic decision-making in cricket matches.

Section 2:

Section 2.1:



Section 2.2:

Average Mean Squared Loss: 1561.30

Section 2.3:

Model parameter Z1: 11.98
Model parameter Z2: 28.37
Model parameter Z3: 50.77
Model parameter Z4: 79.99
Model parameter Z5: 106.08
Model parameter Z6: 136.20
Model parameter Z7: 169.79
Model parameter Z8: 206.99
Model parameter Z9: 234.89
Model parameter Z10: 294.76

Model parameter L: 10.31