**F1 Project**

**Documentation for Machine Learning and Deep Learning Models**

**1. Introduction**

* **Objective**: Predict the position of a Formula 1 driver based on various features such as qualifying times (Q1, Q2, Q3), constructor, and driver information.
* **Approach**: We employed multiple machine learning (ML) and deep learning (DL) models, evaluated their performance, and compared their results.

**2. Preprocessing and Feature Engineering**

* **Data Preparation**:
  + Handling missing values using SimpleImputer.
  + Label encoding categorical variables.
  + Scaling numerical features using StandardScaler.
  + Applying feature weights to qualifying times (Q1, Q2, Q3).
* **Feature Weights**:
  + Emphasized qualifying times (Q1, Q2, Q3) as they are crucial in determining the final position of the drivers in races.

**3. Models Used**

**3.1 Logistic Regression (Logistic Regression Model)**

* **Description:**Logistic Regression is a classical machine learning model typically used for classification tasks but was adapted here for predicting driver positions, treated as continuous values.
* **Implementation:**
  + The model was initialized with a maximum iteration limit of 1000.
  + Fitted to the scaled training data and predictions were made on the test dataset.
* **Results:**
  + **A**ccuracy: 10%

**3.2 XGBoost (eXtreme Gradient Boosting)**

* **Description:** XGBoost is an advanced gradient boosting model known for its robustness and efficiency in handling structured/tabular data.
* **Implementation:**
  + Utilized the XGBRegressor with hyperparameters:  
    n\_estimators=100, max\_depth=6, and learning\_rate=0.1.
  + Trained the model on scaled data and evaluated the predictions.
* **Results:**
  + Mean Absolute Error (MAE): 1.77
  + Root Mean Squared Error (RMSE): 2.33
  + R² Score: 85.93%
* **Conclusion:**The XGBoost model demonstrated substantial predictive power, with relatively low error rates. However, some larger prediction errors were observed, indicating room for improvement.
* **Feature Importance:**
  + The following figure shows the most influential features used by the XGBoost model to make predictions.

**A graph with blue bars

Description automatically generated with medium confidence**

Figure Feature importance

**3.3 Random Forest (Random Forest Classifier & Regressor)**

* **Description**: A robust ensemble model using multiple decision trees for classification/regression tasks.
* **Implementation**:
  + Initially used as a classifier (RandomForestClassifier) and later as a regressor (RandomForestRegressor).
  + Tuned with n\_estimators=100 and a fixed random seed (random\_state=42).
* **Results**:
  + R² Score: 86.12%
  + Accuracy: 19.63%

**Conclusion:**

The model showed strong performance with low bias and high variance.

**3.4 Neural Networks (Simple Neural Network & Optimized Neural Network)**

* **Description**: A feedforward neural network for regression tasks.
* **Implementation**:
  + Simple Neural Network: 128 neurons with ReLU activations, dropout regularization to reduce overfitting.
  + Optimized Neural Network: Used Leaky ReLU, BatchNormalization, and Dropout for enhanced learning and stability.
* **Results of NN 1**:
  + Mean Absolute Error (MAE): 2.48923659324646
  + Root Mean Squared Error (RMSE): 3.2515812475057206
  + R² Score: 72.71%
  + **Conclusion:**  
    The simple neural network showed reasonable performance but could benefit from further optimization.

A graph with a line and a line

Description automatically generated

Figure NN model

* + Conclusion: Improved model stability and convergence with optimization techniques.
* **Results of NN 2 improved**:
  + Mean Absolute Error (MAE): 2.371142625808716
  + Root Mean Squared Error (RMSE): 3.133604639225112
  + R² Score: 74.65% %
  + **Conclusion:**  
    The optimized neural network showed improved performance and greater stability during training, with a higher R² score than the simple model.

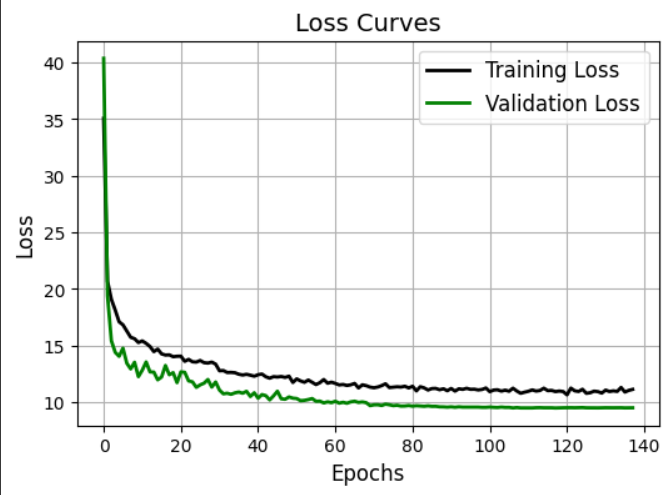


Figure NN improved

**4. Model Comparison**

* **Performance Metrics:** Models were compared based on R² scores (for regression tasks) and accuracy (for classification tasks). Evaluation metrics were presented in bar charts for easier comparison.
* **Results Summary:**
  + Logistic Regression: Accuracy: 10%
  + XGBoost: R² Score: 85.93%
  + Random Forest: R² Score: 86.12%
  + Random Forest Classifier: Accuracy: 19.63%
  + Neural Network (Simple): R² Score: 72.71%
  + Neural Network (Optimized): R² Score: 74.65%

**Conclusion**

* **Best Performing Model:**The XGBoost and Random Forest models performed the best, both achieving high R² scores (85-86%) with relatively low error rates. These models are highly effective for this type of prediction task.
* **Future Improvements:**
  + Hyperparameter Tuning: Further optimization of hyperparameters could improve model accuracy.
  + Ensemble Methods: Experimenting with advanced ensemble techniques or stacking models could lead to even better results.
  + Feature Expansion: Adding additional features such as race conditions or weather data may improve prediction accuracy.

**6. Appendices**

A graph with different colored bars

Description automatically generated

Figure Model Comparison