Cricket Team Winning Probability Prediction

Objective:

The objective of this assignment is to develop a machine learning model that predicts the probability of a cricket team winning a match based on various factors such as team performance, match conditions, and historical data. The model will be evaluated on metrics like accuracy, precision, and recall, and the analysis will provide insights into the key factors that influence match outcomes, helping cricket teams make data-driven strategic decisions.

---

1. Dataset Overview:

The provided dataset contains information about cricket matches, including team performances, match conditions, and outcomes. Each row represents data from a single match, with the following key columns:

- **MatchID:** A unique identifier for each match.
- **Team1 & Team2:** Names of the two teams participating in the match.
- **Venue:** The location where the match was held (e.g., stadium name or city).
- **TossWinner:** The team that won the toss (Team1/Team2).
- **BatFirst:** The team that chose to bat first (Team1/Team2).
- **Team1Score & Team2Score:** The total runs scored by Team1 and Team2, respectively.
- **Team1WicketsLost & Team2WicketsLost:** The number of wickets lost by Team1 and Team2.
- **Team1RunRate & Team2RunRate:** The run rate (runs per over) for Team1 and Team2.
- **WeatherCondition:** The weather during the match (e.g., Clear, Rainy, Overcast).
- **PitchCondition:** The condition of the pitch (e.g., Dry, Wet, Green).
- **WinningTeam:** The target variable indicating which team won the match (Team1/Team2).

---

2. Preprocessing and Feature Engineering:

- **Loading and Inspecting Data:** Load the dataset, check for missing values, and inspect data types.
- **Handling Missing Values:** Fill missing numerical values with the mean/median and categorical values with the mode or a separate category.
- **Encoding Categorical Variables:** Convert categories like Team1, WeatherCondition, and PitchCondition using **Label Encoding** or **One-Hot Encoding**.
- **Target Variable:** Convert WinningTeam into binary format (1 for Team1, 0 for Team2).
- **Feature Scaling: Scale numerical features (e.g., scores, run rates) using standardization or normalization.**

- **New Features:**
    - **Net Run Rate: Difference between Team1 and Team2 run rates.**
    - **Average Score Per Over: Total score divided by overs played for both teams.**

---

3. Model Training:

Model Training

1. **Train-Test Split:** Split the dataset into training and testing sets (e.g., 80-20 split) to evaluate model performance.
2. **Model Selection:** Train multiple classification models, such as:
    - **Logistic Regression**
    - **Decision Trees**
    - **Random Forest**
    - **Gradient Boosting**
3. **Hyperparameter Tuning:** Use **Grid Search** or **Random Search** to optimize model hyperparameters for better performance.

Model Evaluation

1. **Evaluation Metrics:** Assess model performance using:
    - **Accuracy**
    - **Precision**
    - **Recall**
    - **F1-Score**
    - **AUC-ROC**
2. **Model Comparison:** Compare the performance of different models and select the best one based on the above metrics.
3. **Feature Importance:** Analyze feature importance to identify key factors that influence match outcomes.

---

4. Feature Importance:

1. **Tree-Based Models:** Random Forest and Gradient Boosting rank features based on their contribution to decision splits. Likely important features:
    1. **Run Rates**
    2. **TossWinner** and **BatFirst**
    3. **Weather** and **Pitch Conditions**
2. **Logistic Regression:** Larger coefficients indicate higher importance.
3. **Permutation Importance:** Measures the drop in accuracy when features are shuffled, identifying critical features.

Key Insights:

- **Run Rates** and **Wickets Lost** are major predictors.
- **Toss** and **BatFirst** decisions strongly affect outcomes.
- **Weather** and **Pitch** conditions influence performance.

---

5. Recommendations for Optimizing Car Resale Pricing:

Based on the feature importance and model analysis, here are actionable recommendations for cricket teams to improve their chances of winning:

1. **Focus on Run Rate Management:**
    - Teams should aim for a consistently high run rate, as it's a critical factor in match outcomes. Strategic batting partnerships and pacing the innings are essential.
2. **Optimize Toss Decisions:**
    - If winning the toss, teams should consider historical data on how pitch and weather conditions affect performance to make informed decisions on whether to bat or bowl first.
3. **Adapt to Weather and Pitch Conditions:**
    - Tailor team strategies based on weather (e.g., rainy or overcast conditions may favor bowlers) and pitch conditions (e.g., dry or green pitches). Select players who excel in specific conditions.
4. **Minimize Wickets Lost Early:**
    - Protecting wickets early in the match helps in setting a competitive total or chasing down scores, influencing the outcome significantly.
5. **Monitor Opponent's Strengths:**
    - Analyze opponents' weaknesses and adjust strategies, such as targeting specific bowlers or adapting field placements based on prior data..

---

6. Conclusion:

In this assignment, we developed a machine learning model to predict the winning probability of a cricket team based on various match-related factors. By preprocessing the data, engineering relevant features, and evaluating multiple models, we identified that **run rates**, **toss decisions**, and **match conditions** (weather and pitch) play significant roles in determining match outcomes.

The best-performing model provided actionable insights, allowing teams to focus on optimizing run rates, making informed toss decisions, and adjusting strategies based on weather and pitch conditions. These findings can help cricket teams improve their decision-making and increase their chances of success in future matches.