March Madness Predictive Modeling

Pedro Castro, Ryan Walentowicz, Nick Zappia, Pranati Mitta, Bugz Valente

Addressed Problem Details and Motivation

The project aims to utilize machine learning techniques to predict the outcome of the NCAA March Madness basketball games. With data spanning the last 15 years of March Madness statistics, the goal is to develop a model capable of accurately predicting game winners. The motivation behind this project was each of our interests and the immense popularity surrounding March Madness. Predicting game outcomes could have significant implications for basketball fans, bettors, and sports analysts.

Formal Problem Definition

- Task (T): Predict the winning percentage of March Madness teams
- Experience (E): Data from March Madness seasons 2002-2019
    - https://www.kaggle.com/datasets/nishaanamin/march-madness-data
- Performance (P): Accuracy in predicting a team's winning percentage

Machine Learning Type

Supervised Learning

Regression

Used Dataset Details

Disk Size: 2.5 MB

Source: Kaggle – NCAA March Madness Dataset (2008-2023/2024)

Number of Samples: Approximately 23,000 games

Number of Features: 541 Total Features in all files, 135 used

Column Details: 135 columns, listed by variable name below

YEAR, CONF, CONF ID, QUAD NO, QUAD ID, TEAM NO_x, TEAM ID_x, TEAM, SEED_x, ROUND_x, K TEMPO, K TEMPO RANK, KADJ T, KADJ T RANK, K OFF, KO RANK, KADJ O, KADJ O RANK, K DEF, KD RANK, KADJ D, KADJ D RANK, KADJ EM, KADJ EM RANK, BADJ EM, BADJ O, BADJ D, BARTHAG, GAMES_x, W_x, L_x, WIN%_x, EFG%, EFG%D, FTR, FTRD, TOV%, TOV%D, OREB%, DREB%, OP OREB%, OP DREB%, RAW T, 2PT%, 2PT%D, 3PT%, 3PT%D, BLK%, BLKED%, AST%, OP AST%, 2PTR, 3PTR, 2PTRD, 3PTRD, BADJ T, AVG HGT, EFF HGT, EXP, TALENT, FT%, OP FT%, PPPO, PPPD, ELITE SOS, WAB, BADJ EM RANK, BADJ O RANK, BADJ D RANK, BARTHAG RANK, EFG% RANK, EFGD% RANK, FTR RANK, FTRD RANK, TOV% RANK, TOV%D RANK, OREB% RANK, DREB% RANK, OP OREB% RANK, OP DREB% RANK, RAW T RANK, 2PT% RANK, 2PT%D RANK, 3PT% RANK, 3PT%D RANK, BLK% RANK, BLKED% RANK, AST% RANK, OP AST% RANK, 2PTR RANK, 3PTR RANK, 2PTRD RANK, 3PTRD RANK, BADJT RANK, AVG HGT RANK, EFF HGT RANK, EXP RANK, TALENT RANK, FT% RANK, OP FT% RANK, PPPO RANK, PPPD RANK, ELITE SOS RANK, TEAM NO_y, SEED_y, ROUND_y, AP VOTES, AP RANK, RANK?, TEAM ID_y, PAKE, PAKE RANK, PASE, PASE RANK, GAMES_y, W_y, L_y,

WIN%_y, R64, R32, S16, E8, F4, F2, CHAMP, TOP2, F4%, CHAMP%, BY YEAR NO, BY ROUND NO, TEAM NO, SEED, ROUND, CURRENT ROUND, SCORE

Metrics

Evaluation Metrics:

- Mean Squared Error (MSE)
- R Squared (R^2)
- Mean Absolute Error (MAE)

Models

- Linear Regression (With Stepwise Regression)
    - Train RMSE: 0.059534579738275605
    - Test RMSE: 0.0839783171013451
    - Train R^2: 0.9354877866832415
    - Test R^2: 0.8719361629312715
- SVM for Regression (With Stepwise Regression)
    - Train MSE: 0.005071586635460595
    - Test MSE: 0.007923909728489692
    - Train R^2: 0.8759123857741552
    - Test R^2: 0.8033526510106256
    - Train MAE: 0.0623489868371097
    - Test MAE: 0.06996411135115455
- Random Forest for Regression (With Stepwise Regression)
    - Train MSE: 1.0108503613181397e-05
    - Test MSE: 5.605191853239837e-05
    - Train R^2: 0.999815141844911
    - Test R^2: 0.9989718253667119
    - Train MAE: 0.0012113874138681877

    - Test MAE: 0.0028673426765913903
- Neural Network
    - Train MSE: 0.004650577991182201
    - Test MSE: 0.009215902974294038
    - Train R^2: 0.912418176436343
    - Test R^2: 0.8604091569472837
    - Train MAE: 0.04788041231369084
    - Test MAE: 0.06552074742953605

Chosen ML Models

- Support Vector Machine (SVM) for regression
- Random Forest for regression
- Linear Regression
- Neural Network

Training and Test Details

- Data is generated synthetically with 1000 samples and 10 features
- Data is split into training and testing sets with a size of 20%

Analysis of Obtained Results

1. Linear Regression:
    a. Training Performance:
        i. RMSE: 0.0595
        ii. R^2: 0.9355
    b. Test Performance:
        i. RMSE: 0.0840
        ii. R^2: 0.8719

c. Analysis: Linear Regression provides a straightforward approach, predicting outcomes through a linear combination of features. The training R^2 score indicates that it explains around 93% of the variance in the training data, while the test R^2 shows that it captures about 87% of the variance in the test data. The slight drop in test R^2 suggests some overfitting, though the model still generalizes reasonably well.

2. Support Vector Machine (SVM) for Regression:
    a. Training Performance:
        i. MSE: 0.0051
        ii. R^2: 0.8759
        iii. MAE: 0.0623
    b. Test Performance:
        i. MSE: 0.0079
        ii. R^2: 0.8034
        iii. MAE: 0.0700
    c. Analysis: SVM demonstrates a solid performance, achieving an R^2 score of about 87% on training data and 80% on test data. The drop in test performance indicates overfitting. Since the SVM model uses a hyperplane to capture complex relationships between features, it makes it a suitable choice for datasets with non-linear patterns.
3. Random Forest for Regression:
    a. Training Performance:
        i. MSE: 1.0108e-05
        ii. R^2: 0.09998
        iii. MAE: 0.0012
    b. Test Performance:
        i. MSE: 5.6052e-05
        ii. R^2: 0.9990
        iii. MAE: 0.0029

c. Analysis: Random Forest model shows a good performance with a near-perfect R^2 score on both training and test sets. The low MAE values further highlight its precise predictions. The model's ability to handle both linear and non-linear relationships makes it highly versatile for our task.

4. Neural Network:
    a. Training Performance:
        i. MSE: 0.0047
        ii. R^2: 0.9124
        iii. MAE: 0.0479
    b. Test Performance:
        i. MSE: 0.0092
        ii. R^2: 0.8604
        iii. MAE: 0.0655
    c. Analysis: Neural Network model appears well performing with an R^2 value of around 91.2% for training and 86.0% for testing. This for testing indicates overfitting likely due to the high number of layers in the model and neurons in each layer relative to the low complexity of the Team Results Dataset (used to create this model). This model is ideal to gain a visual understanding of the various teams by using the second to last layer of the model to output an embedding for each team to be represented in a 3-dimensional space. The embeddings of teams closer together in this space represent more similar performing teams.

Comparative Analysis

1. Accuracy:
    a. The Random Forest model leads in accuracy, with an R^2 score of 0.999 on both training and test sets, indicating it can effectively capture most of the data's variance.
    b. Linear Regression and SVM for Regression perform comparably, with R^2 scores between 0.80 and 0.87, demonstrating a reasonable balance between capturing variance and generalization.

    c. Neural Network's accuracy performs comparably, with R^2 scores between .91 and .86, demonstrating a relatively good capture of the data's variance.
2. Overfitting:
    a. Random Forest shows minimal overfitting, maintaining a high R^2 score on both training and test sets. This suggests its ensemble approach is effective at capturing both training and unseen data patterns.
    b. Linear Regression and SVM for Regression exhibit moderate overfitting, with tests R^2 scores lower than training ones, though they still generalize reasonably well.
    c. The Neural Network model demonstrates moderate overfitting, with the test R^2 score being around 0.05 lower than that of the train, likely due to the large number of neurons in each layer, relative to the complexity of the Team Results Dataset.
3. Use Cases:
    a. Linear Regression: Provides a simple, interpretable model, useful for understanding relationships between variables and making straightforward predictions.
    b. SVM for Regression: Suitable for datasets with non-linear patterns, utilizing a hyperplane to capture complex relationships.
    c. Random Forest: Versatile and accurate, making it ideal for this task due to its ensemble nature, which combines multiple decision trees.
    d. Neural Networks: Ability to extract embeddings (one for each team) from the second to last layer of the neural network. For future steps, these embeddings can be projected on a three-dimensional space using PCA or other visualization techniques like UMAP (and possible clustering analysis can be done to understand the relations of teams). This is ideal for gaining a better understanding of the various March Madness teams to get a visualization that demonstrates the teams most similar in performance as closest together.

Link to Presentation:

https://www.canva.com/design/DAGD8fZlsNk/0lAOyQXlFI2tqnSraTN8hw/edit?utm_content=DAGD8fZlsNk&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

TensorFlow Visualization:

https://projector.tensorflow.org/?config=https://gist.githubusercontent.com/ryanwal28/5af70e2696b6f9bf1fa2b708a1c306bc/raw/cab2b210ba6f49cf1a922b28bbb01aee6f10acbe/projector_config.json