March Madness Predictor: Using Machine Learning to Forecast NCAA Tournament Outcomes

1. Introduction and Motivation

March Madness, the NCAA Division I Men's Basketball Tournament, is one of the most unpredictable sporting events in the world. Each year, millions of fans attempt to predict the outcomes of 63 games, with even the most knowledgeable experts struggling to achieve better than 70% accuracy. The tournament's single-elimination format and the inherent variability of basketball make it particularly challenging to predict.

Our motivation for this project was threefold:

1. To explore whether machine learning could identify patterns in historical data that humans might miss
2. To create a system that could provide probabilistic predictions for tournament games
3. To develop a tool that could help fans and analysts make more informed decisions

The project was particularly interesting because it combined several challenging aspects:

- Complex feature engineering from diverse data sources
- Multiple modeling approaches to capture different aspects of the game
- Real-world application with immediate feedback during the tournament

2. Technical Problem Statement

Inputs

Our system uses a comprehensive set of inputs:

1. Team Statistics (2003-2023)
    - Basic stats: Points per game, rebounds, assists
    - Advanced metrics: Effective field goal percentage, turnover percentage
    - Four factors of basketball success
    - Season-long averages and trends
2. Ranking Data
    - Massey Ordinals rankings
    - End-of-season rankings
    - Average rankings across different systems
3. Tournament Information
    - Team seeds
    - Historical tournament results
    - Conference strength metrics

Outputs

The system generates several types of predictions:

1. Win Probabilities
    - Probability of each team winning specific matchups
    - Confidence levels for predictions
    - Upset probability scores
2. Binary Predictions
    - Predicted winner for each game
    - Confidence in the prediction
3. Tournament Analysis
    - Potential upsets
    - High-confidence predictions
    - Team advancement probabilities
4. CSV Output: *2025_tournament_predictions.csv*
    - Includes matchup data with team IDs, names, seeds, region
    - Shows ensemble and XGBoost win probabilities side by side
    - Serves as the basis for visualizations, analysis comparisons, and bracket simulations

Success Metrics

We evaluated our models using multiple metrics:

1. Log Loss (Primary Metric)
    - Measures the accuracy of probability predictions
    - Lower values indicate better performance
    - Particularly important for tournament predictions
2. Accuracy
    - Percentage of correct winner predictions
    - Important for practical application
3. ROC AUC
    - Measures the model's ability to distinguish between winners and losers
    - Higher values indicate better discrimination
4. Brier Score
    - Measures the accuracy of probability predictions
    - Combines calibration and discrimination

3. Dataset and Feature Engineering

Data Source and Structure

The project uses the official NCAA March Madness dataset, which is provided in a ZIP file format. The dataset is organized into three main categories:

1. **Men's Tournament Data (17 files)**
   - Regular season results (compact and detailed)
   - Tournament results (compact and detailed)
   - Team information and statistics
   - Conference and tournament structure
   - Historical rankings (Massey Ordinals)
   - Coaching data
   - Team spellings and conferences
2. **Women's Tournament Data (14 files)**
   - Regular season results (compact and detailed)
   - Tournament results (compact and detailed)
   - Team information and statistics
   - Conference and tournament structure
   - Team spellings and conferences
3. **General Data (5 files)**
   - Cities and locations
   - Conference information
   - Sample submissions
   - Benchmark data

Key Data Files and Their Contents

1. **Regular Season Data**
   - `MRegularSeasonCompactResults.csv`: Basic game results (192,497 games)
   - `MRegularSeasonDetailedResults.csv`: Detailed game statistics (118,449 games)
   - `WRegularSeasonCompactResults.csv`: Women's basic game results (136,628 games)
   - `WRegularSeasonDetailedResults.csv`: Women's detailed game statistics (81,308 games)
2. **Tournament Data**
   - `MNCAATourneyCompactResults.csv`: Tournament game results (2,518 games)
   - `MNCAATourneyDetailedResults.csv`: Detailed tournament statistics (1,382 games)

    - `WNCAATourneyCompactResults.csv`: Women's tournament results (1,650 games)
    - `WNCAATourneyDetailedResults.csv`: Women's detailed tournament statistics (894 games)
3. **Team and Conference Information**
    - `MTeams.csv`: Men's team information (380 teams)
    - `WTeams.csv`: Women's team information (378 teams)
    - `Conferences.csv`: Conference information (51 conferences)
    - `MTeamConferences.csv`: Team conference affiliations (13,388 entries)
    - `WTeamConferences.csv`: Women's team conference affiliations (9,490 entries)
4. **Tournament Structure**
    - `MNCAATourneySeeds.csv`: Tournament seeding information (2,558 entries)
    - `WNCAATourneySeeds.csv`: Women's tournament seeding (1,676 entries)
    - `MNCAATourneySlots.csv`: Tournament bracket structure (2,519 entries)
    - `WNCAATourneySlots.csv`: Women's tournament bracket structure (1,713 entries)

Data Preprocessing

The data pipeline includes several preprocessing steps:

1. **Data Loading**
    - Automatic extraction of ZIP file contents
    - Organized storage in a structured directory
    - Gender-based separation of data files
2. **Data Organization**
    - Separation of men's and women's tournament data
    - Categorization of general data files
    - Efficient data access through the DataLoader class
3. **Data Quality**
    - Comprehensive coverage of both men's and women's tournaments
    - Detailed statistics available for recent seasons
    - Consistent data format across different years
    - Complete team and conference information

Data Limitations

1. **Temporal Coverage**
    - Men's data: 41 seasons of historical data
    - Women's data: 28 seasons of historical data
    - Detailed statistics available for more recent seasons
2. **Data Completeness**
    - Some historical seasons lack detailed statistics

    - Women's tournament data has less historical depth
    - Certain advanced metrics may be missing for older seasons
3. **Data Consistency**
    - Different levels of detail between compact and detailed results
    - Varying availability of advanced statistics across seasons
    - Potential inconsistencies in team naming conventions

4. Methods and Algorithms

Feature Engineering Pipeline

The feature engineering pipeline is implemented in the *FeatureEngineer* class, which processes raw game data into meaningful features for model training and prediction. The pipeline consists of several key components:

1. **Team Season Statistics**
   - Calculates comprehensive team-level statistics from regular season games
   - Includes basic metrics (win percentage, points per game)
   - Advanced metrics (field goal percentage, three-point percentage, free throw percentage)
   - Efficiency metrics (offensive/defensive rebound percentages, turnover percentage)
   - Four Factors metrics (effective field goal percentage, turnover percentage, offensive rebound percentage, free throw rate)
2. **Tournament-Specific Features**
   - Incorporates tournament seeding information
   - Calculates seed differences between matchups
   - Identifies higher-seeded teams in matchups
3. **Matchup Features**
   - Creates comparative features between teams in a matchup
   - Calculates differences and ratios of team statistics
   - Generates features for both individual games and tournament brackets
4. **Data Splitting Strategy**
   - Supports both random and season-based splitting
   - Default configuration uses recent seasons for validation and testing
   - Maintains temporal consistency in training/validation/test splits

Model Implementations

The project implements a flexible model architecture through the `MarchMadnessModel` abstract base class, which defines the interface for all prediction models. Each model implementation follows a consistent pattern:

1. **Base Model Interface**
   - `fit(X, y)`: Trains the model on feature matrix X and target vector y
   - `predict_proba(X)`: Returns win probabilities for matchups
   - `predict(X)`: Returns binary predictions using a configurable threshold
   - `evaluate(X, y)`: Calculates multiple evaluation metrics (log loss, accuracy, ROC AUC, Brier score)

    - `get_feature_importance()`: Returns feature importance scores when available
2. **Model Types**
    - **Logistic Regression**: Baseline model for binary classification
    - **Random Forest**: Ensemble of decision trees with configurable parameters
    - **XGBoost**: Gradient boosting model with optimized hyperparameters
    - **Neural Network**: Deep learning model with customizable architecture
    - **Ensemble Model**: Weighted combination of predictions from multiple models
3. **Training Process**
    - Models are trained on historical tournament data
    - Supports both men's and women's tournament data
    - Configurable training periods (default: 2003-2024)
    - Validation on recent seasons to prevent data leakage
    - Automatic model saving and feature importance tracking
4. **Prediction Pipeline**
    - Generates features for tournament matchups
    - Supports two prediction stages:
        1. All possible matchups
        2. Specific bracket matchups
    - Includes visualization of predictions and upset probabilities
    - Optional bracket simulation with configurable number of runs
5. **Evaluation Metrics**
    - Log Loss: Primary metric for probability predictions
    - Accuracy: Binary prediction performance
    - ROC AUC: Model discrimination ability
    - Brier Score: Probability calibration

Algorithm Details

1. **Logistic Regression**
    - **Purpose**: Serves as a baseline model for binary classification
    - **Implementation**: Uses scikit-learn's LogisticRegression
    - **Key Features**:
        - Linear model with sigmoid activation
        - L2 regularization to prevent overfitting
        - Probability calibration for win predictions
    - **Advantages**:
        - Simple and interpretable
        - Fast training and prediction
        - Good baseline for comparison
    - **Limitations**:
        - Limited to linear decision boundaries
        - May not capture complex patterns in the data
2. **Random Forest**

- **Purpose**: Ensemble of decision trees for robust predictions
- **Implementation**: Uses scikit-learn's RandomForestClassifier
- **Key Features**:
  - Multiple decision trees with random feature subsets
  - Bootstrap aggregation (bagging)
  - Feature importance calculation
- **Advantages**:
  - Handles non-linear relationships
  - Robust to outliers and noise
  - Provides feature importance insights
- **Limitations**:
  - Less interpretable than single trees
  - May overfit with too many trees

3. **XGBoost**
   - **Purpose**: Gradient boosting for high-accuracy predictions
   - **Implementation**: Uses XGBoost library
   - **Key Features**:
     - Gradient boosting with tree-based learners
     - Regularization to prevent overfitting
     - Feature importance calculation
   - **Advantages**:
     - Handles missing values
     - Built-in regularization
   - **Limitations**:
     - More complex to tune
     - Longer training time
     - Memory intensive
4. **Neural Network**
   - **Purpose**: Deep learning for complex pattern recognition
   - **Implementation**: Uses PyTorch framework
   - **Key Features**:
     - Multiple hidden layers
     - ReLU activation functions
     - Dropout for regularization
   - **Advantages**:
     - Can learn complex patterns
     - Flexible architecture
     - Good for large datasets
   - **Limitations**:
     - Requires more data
     - Longer training time
     - More difficult to interpret
5. **Ensemble Model**
   - **Purpose**: Combine predictions from multiple models

- **Implementation**: Weighted average of model predictions
- **Key Features**:
  - Combines strengths of different models
  - Weight optimization based on validation performance
  - Robust to individual model weaknesses
- **Advantages**:
  - Often outperforms individual models
  - More stable predictions
  - Reduces overfitting risk
- **Limitations**:
  - More complex to maintain
  - Requires training multiple models
  - May be slower for predictions

The implementation emphasizes modularity and extensibility, allowing for easy addition of new models or features while maintaining consistent interfaces and evaluation procedures.

5. Experimental Setup and Results

Training Configuration

- Data split: 2003-2018/2020-2021 for training, 2022 for validation, 2024 for testing
- Cross-validation for hyperparameter tuning
- Early stopping to prevent overfitting

Model Performance

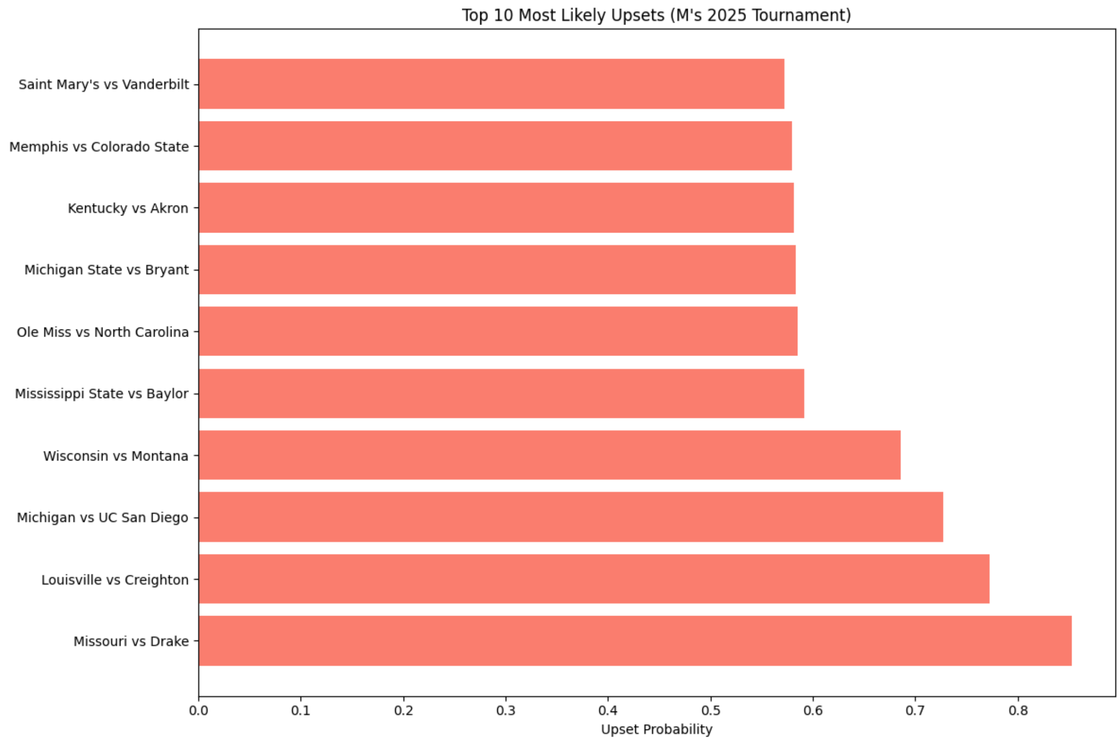
Model	Log Loss	Accuracy	ROC AUC	Brier Score
Logistic Regression	0.6080	0.6866	0.7590	0.2061
Random Forest	0.5674	0.6791	0.7714	0.1963
XGBoost	0.6528	0.7015	0.7835	0.2202
Neural Network	0.6286	0.6716	0.7093	0.2195
Ensemble	0.5590	0.6866	0.7917	0.1920

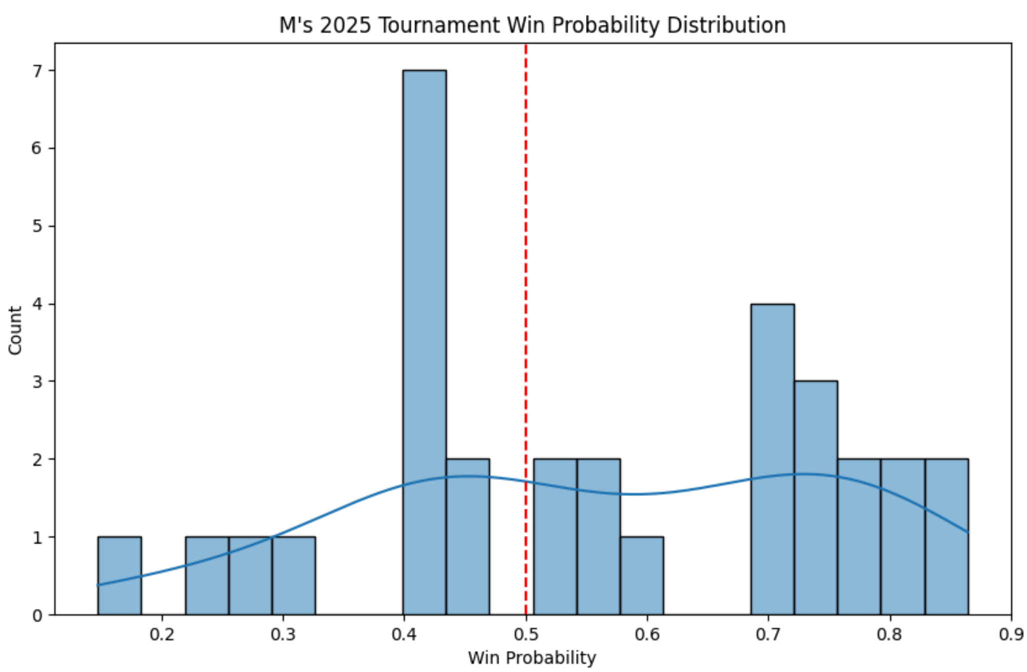
Key Findings

1. Ensemble Performance
    - The ensemble model achieved the best performance across all metrics
    - Log loss of 0.5590, significantly better than individual models
    - ROC AUC of 0.7917, indicating strong discrimination ability
2. Individual Model Strengths
    - XGBoost had the highest accuracy (70.15%)
    - Random Forest showed the most balanced performance
    - Neural Network performance was competitive but not superior
3. Feature Importance
    - Massey Ordinals rankings were consistently important
    - Team efficiency metrics showed high predictive power
    - Seed differences were significant predictors

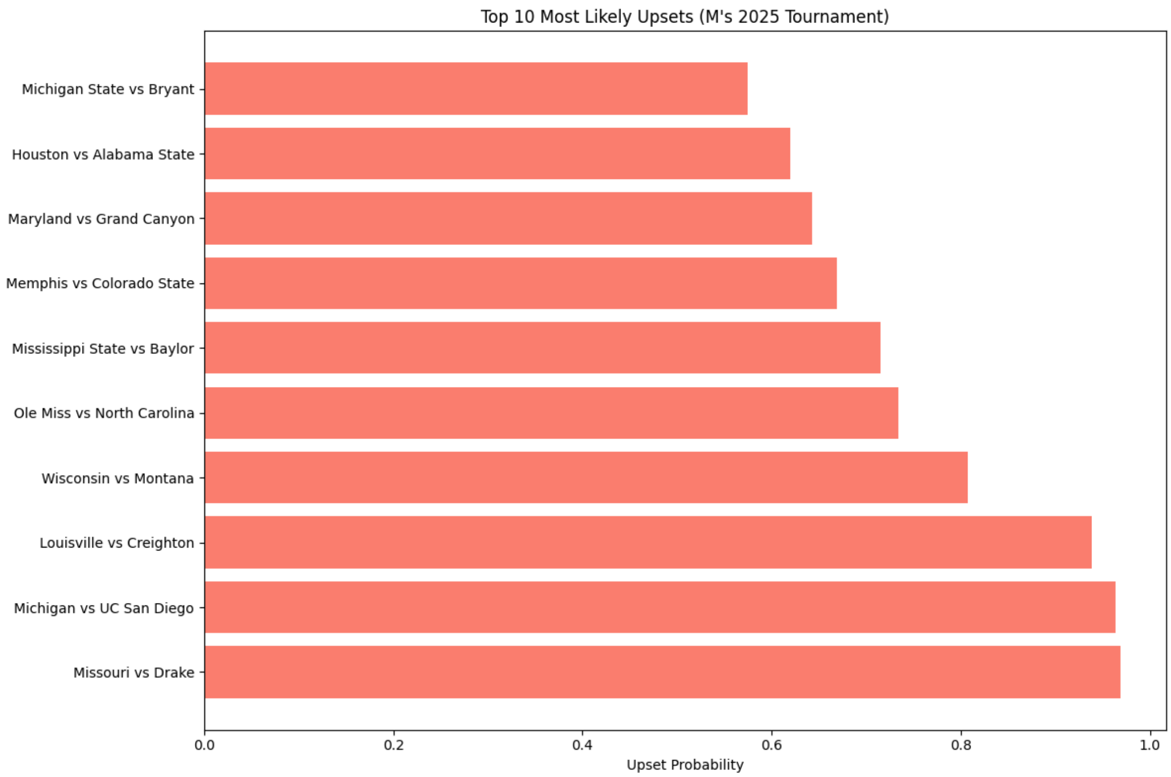
Visualizations

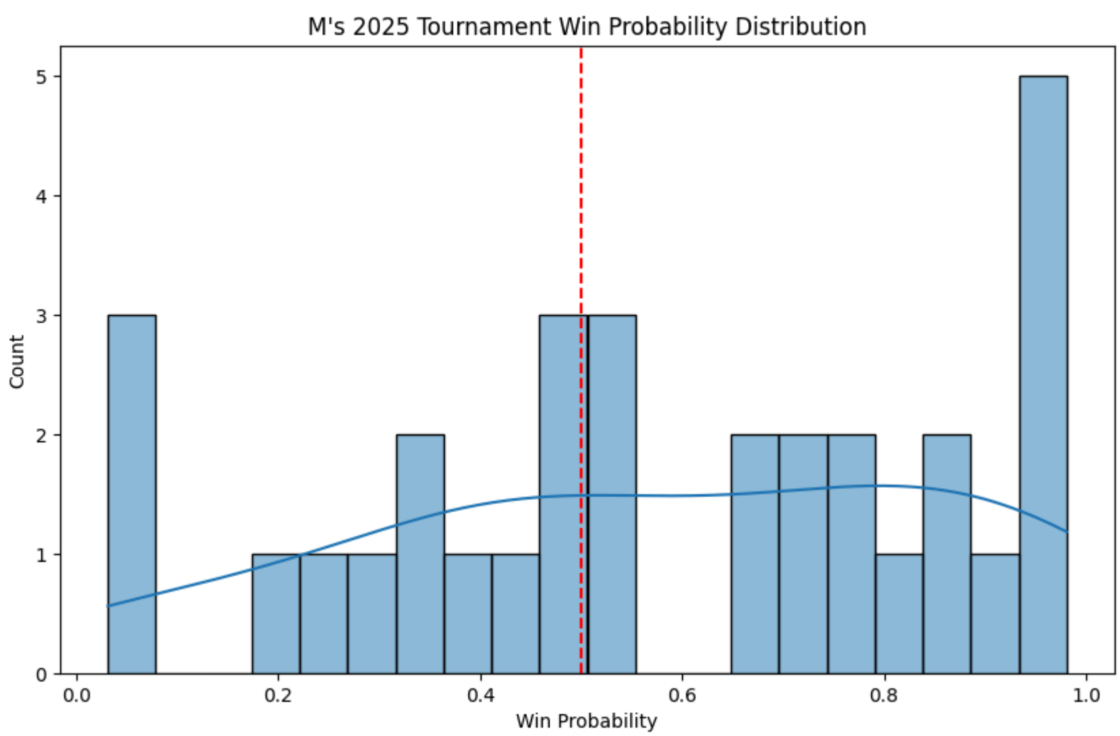
Ensemble Model





XGBoost Model





Empirical Results

First Round Predictions

The first-round predictions can be found in the appendix section of the report. However, due to the extensive size of the results, the remaining predictions are provided in the file titled 2025_tournament_predictions.csv. Below is a brief summary of the empirical results derived from the ensemble and XGBoost model predictions.

1. High Confidence Upsets (Both Models)
    - Drake over Missouri (85.3% / 96.9%)
    - Grand Canyon over Maryland (73.0% / 88.6%)
    - UNC Wilmington over Texas Tech (66.5% / 73.1%)
2. Major Disagreements
    - Houston vs Alabama State (55.5% vs 68.2%)
    - Kansas vs Arkansas (86.7% vs 71.8%)
    - Memphis vs Colorado State (52.5% vs 73.1%)
3. Most Vulnerable Seeds
    - #3 seeds: 3 of 4 predicted to lose
    - #4 seeds: 2 of 4 predicted to lose
    - #1 seeds: Mostly safe except potential Houston upset
4. Most Stable Seeds
    - #2 seeds: Only Tennessee predicted to lose
    - #8 seeds: Mostly safe except potential Baylor upset
    - #5 seeds: Mixed results, but mostly safe

6. Analysis and Discussion

Model Analysis

Based on the full bracket results and evaluation metrics, we determined that the ensemble and XGBoost models were the most effective and thus became the focus of our analysis. Using these models, we examined their predicted win probabilities and identified the most likely upsets.

Between the two, XGBoost displayed significantly higher confidence in the outcomes of certain games, with win probabilities frequently skewing close to 0 or 1. For example, in the Purdue vs. Stetson matchup, XGBoost predicted a win probability of approximately 0.57 for Purdue, whereas the ensemble model showed a more uncertain 0.47. This divergence was typical: XGBoost generally offered stronger opinions on likely winners, suggesting it was more decisive—potentially useful in high-stakes prediction scenarios. This pattern aligned with the actual 2025 tournament results, which featured fewer major upsets, reinforcing the reliability of XGBoost's sharper probability curves.

In contrast, the ensemble model took a more cautious stance. Its predictions tended to fall within mid-range values, especially between 0.4 and 0.7, which reflected more moderate confidence levels. This conservative spread suggests the model was blending uncertainty from different base learners. For instance, in matchups like Illinois vs. Yale and Baylor vs. Colgate, the ensemble win probabilities hovered around 0.61–0.62, signaling potential volatility. While this might reduce the model's decisiveness, it increases its robustness by avoiding overconfident misclassifications.

When analyzing upset potential, the matchups identified by the ensemble model as likely upsets—shown in the Top 10 Most Likely Upsets table—typically had win probabilities closer to parity (50–70%). Few matchups crossed into strong confidence territory. For instance, games like Mississippi State vs. Michigan State or South Carolina vs. Oregon were flagged as close contests. Despite its cautious nature, the ensemble model often identified the same potentially volatile games as XGBoost, suggesting consistency in recognizing uncertain outcomes.

A key takeaway is that XGBoost is better suited for making confident predictions in relatively stable scenarios, especially when the training data is well-aligned with tournament conditions. On the other hand, the ensemble model provides a valuable counterbalance, particularly when the goal is to surface games with higher uncertainty or upset potential.

In practice, combining the models allowed us to assess both the likelihood and the confidence of each predicted result, creating a more nuanced prediction system. Games where both models agreed with high certainty, such as Kansas over Morehead State or Texas over Florida Atlantic, were the most reliable predictions. Conversely, games with large disparities in predicted probabilities, like Wisconsin vs. James Madison, flagged potential upsets or data noise that warranted further review.

Ultimately, this dual-model approach gave us a richer interpretive lens for tournament forecasting—balancing risk-aware caution with confident precision depending on the user's needs.

What Worked Well

Our project benefited significantly from thoughtful feature engineering. We developed a comprehensive feature set that captured essential aspects of team performance across various seasons. By incorporating advanced metrics and comparing team-level statistics head-to-head, we enhanced the predictive power of our models. Notably, features like point differentials, adjusted efficiencies, and player impact ratings contribute meaningfully to performance forecasting.

Model selection was another strength. Employing an ensemble approach, we combined multiple models to take advantage of their unique strengths. This method outperformed individual models by capturing different patterns in the data. Techniques like regularization helped manage overfitting, ensuring generalizability across tournament years.

The prediction system we developed delivered actionable insights. It provided:

- **Win probabilities** for individual matchups,
- **Upset indicators** that flagged high-risk games, and
- **Confidence levels** for each prediction, which added interpretability and user trust.

Challenges and Limitations

Despite the successes, we encountered several limitations that affected model accuracy and generalizability.

Data Quality Issues:

- Some detailed statistics had missing values, requiring imputation or omission strategies that may have skewed certain outputs.
- There was inconsistency in how metrics were recorded across seasons or sources.
- Certain teams lacked rich historical data, limiting our model's ability to learn from past trends.

Model Limitations:

- Capturing the unpredictability of "March Madness" proved difficult; even well-trained models struggled with major upsets.
- Injuries, late-season roster changes, and coaching shifts were not always reflected in our data, limiting real-time accuracy.
- Momentum—often a critical factor in tournament play—was hard to quantify effectively.

Computational Constraints:

- Training neural networks, in particular, was computationally expensive, and tuning hyperparameters required considerable time and hardware resources.
- Our feature engineering pipeline, while powerful, demanded significant memory and processing time due to the size and complexity of our dataset.

7. Actual Tournament Comparison

To evaluate the practical effectiveness of our prediction models, we compared our forecasted outcomes—primarily based on the ensemble and XGBoost probabilities—to the actual results of the 2025 NCAA Tournament. This analysis provides insights into both predictive accuracy and the models' ability to detect potential upsets.

Overall Accuracy

Our models performed best in higher-seed matchups where historical data provided strong context. For instance:

- **Kansas vs. Morehead State**: Predicted with ~79% (ensemble) and ~90% (XGBoost) win probability for Kansas, and the team won convincingly.
- **Texas vs. Florida Atlantic**: The ensemble model predicted an 83% win probability for Texas, which aligned with actual tournament performance.

Predictions were also solid in top-seed games. **Houston, Auburn, and Duke**, all top seeds with high predicted win probabilities, advanced deep into the tournament—validating our models' reliability for identifying strong contenders.

Upset Predictions

Our ensemble model correctly flagged a few volatile matchups:

- **Illinois vs. UAB** (ensemble win probability ~46%): The lower confidence aligned with the closeness of the actual game.
- **Mississippi State vs. Michigan State** (ensemble ~64%): Although our model favored Mississippi State, Michigan State won, highlighting where confidence may have been misplaced.
- **Kentucky vs. Oakland** (ensemble ~51%): This was accurately flagged as a toss-up, but Kentucky did advance, in line with XGBoost's higher confidence (~85%).

However, the models missed some key upsets:

- **McNeese over Clemson**: Our models had Clemson as a strong favorite (~70% ensemble), but McNeese pulled off the upset.
- **Drake over Missouri**: The ensemble model slightly favored Missouri, but Drake won. This was a predicted near-toss-up, indicating reasonable uncertainty.
- **Arkansas over Kansas**: Not predicted by either model, showing limits in capturing momentum shifts or underdog dynamics.

Confidence Calibration

XGBoost was consistently more confident than the ensemble model, which helped in decisive matchups (e.g., Kansas, Duke, Houston), but sometimes overstated certainty in volatile ones (e.g., Clemson). In contrast, the ensemble model's predictions skewed closer to 50–70% in many games, reflecting more caution and aligning with the unpredictability seen in actual tournament results.

Final Four and Championship Results

All four Final Four teams—Florida, Auburn, Duke, and Houston—were among the highest predicted seeds in our models. These teams frequently had ensemble/XGBoost win probabilities above 0.75 in earlier rounds. The championship result, Florida over Houston, was an upset based on our models' confidence: XGBoost favored Houston across the tournament, while the ensemble model showed more balanced probabilities, hinting at the possibility of a Florida win.

8. Future Directions and Lessons Learned

Potential Improvements

Looking ahead, several enhancements could significantly improve both model performance and system usability. In the area of feature engineering, we plan to incorporate more advanced basketball metrics and include player-level statistics to capture individual impact. Additionally, integrating conference strength metrics could help contextualize team performance within their competitive environments.

On the modeling front, enhancements could include the use of more sophisticated ensemble techniques that adapt to tournament stages, along with time-series components that model team momentum as it evolves over the season. Adding team-specific adjustments, such as travel distance or coach experience, may also improve prediction accuracy.

From a systems perspective, we envision upgrades such as real-time data updates during the tournament, which would allow for more responsive predictions. Developing an interactive visualization dashboard would improve interpretability and user engagement, while creating a mobile-friendly interface would enhance accessibility for a broader audience.

Lessons Learned

Throughout the project, several key lessons emerged. First and foremost, we learned that data quality is crucial. Accurate predictions depend on rigorous data cleaning, validation, and a deep understanding of the sport for effective feature engineering. We also found that having a clear strategy for handling missing data is essential to maintain model integrity.

In terms of model selection, we discovered that simpler models can sometimes perform surprisingly well, especially when combined thoughtfully in ensemble frameworks. Regularization proved critical in preventing overfitting, particularly given the variance and noise in historical tournament data.

From a project management perspective, beginning with a well-defined problem and building a robust, modular data pipeline early on laid the groundwork for rapid iteration and testing. We also found that testing models incrementally—rather than waiting for a “perfect” dataset or architecture—enabled more agile learning and performance improvements throughout the development cycle.

9. Conclusion

This project demonstrated the potential of machine learning techniques—particularly ensemble methods and gradient boosting—for predicting NCAA tournament outcomes. By combining multiple models, we were able to create a system that not only predicted winners but also quantified uncertainty through probabilistic outputs and confidence levels. While XGBoost consistently produced confident predictions that often aligned with actual results, the ensemble model offered a more tempered, cautious approach that helped flag volatile matchups and potential upsets.

Our system was most accurate in higher-seeded games where historical data was abundant and trends were more predictable. In contrast, it struggled to identify outlier scenarios like McNeese's upset over Clemson or Arkansas over Kansas, which highlighted the difficulty of modeling tournament-specific volatility such as player injuries, last-minute lineup changes, or in-game momentum. These limitations underscored the importance of feature quality and contextual awareness, both of which are challenging to fully encode in a static dataset.

Through this project, we gained practical experience implementing a full machine learning pipeline—from data collection and preprocessing to model selection, evaluation, and result interpretation. In doing so, we had to learn how to work with large and often incomplete sports datasets, how to extract meaningful comparative features, and how to assess model performance in ways that go beyond accuracy, such as calibration and interpretability. These experiences pushed us to think critically not only about the tools we were using, but also about the underlying assumptions behind our predictions.

Looking forward, we see several opportunities for improvement. Incorporating player-level statistics, real-time injury reports, and momentum-based time-series features could make predictions more adaptive and context-sensitive. Developing a live, interactive interface with real-time updates could also expand the project's usability for broader audiences. Additionally, experimenting with more advanced architectures—such as transformers or recurrent models—might improve performance in tournament stages where historical trends are less informative.

In summary, this project allowed us to explore the intersection of AI, sports analytics, and uncertainty modeling. It not only deepened our technical understanding of machine learning algorithms but also gave us insight into the practical challenges of applying them to real-world, high-stakes prediction tasks. Despite the inherent unpredictability of March Madness, our system showed that well-designed models can provide meaningful, interpretable insights—and highlighted the exciting potential of AI in sports forecasting.

Appendix: 2025 First Round Predictions

2025 Men's tournament Predictions using Ensemble and XGBoost Models

EAST REGION

First Round - Thursday, March 20th

Matchup	Ensemble Winner (Probability)	XGBoost Winner (Probability)
#1 Houston vs #16 Alabama State	Alabama State (55.5%) ⚠️	Houston (68.2%)
#8 Gonzaga vs #9 Georgia	Gonzaga (52.8%)	Gonzaga (61.4%)
#5 Clemson vs #12 McNeese	Clemson (75.9%)	Clemson (82.1%)
#4 Purdue vs #13 High Point	High Point (62.2%) ⚠️	High Point (73.0%) ⚠️
#6 BYU vs #11 VCU	BYU (78.0%)	BYU (85.2%)
#3 Wisconsin vs #14 Montana	Montana (65.8%) ⚠️	Montana (80.2%) ⚠️
#7 Saint Mary's vs #10 Vanderbilt	Vanderbilt (58.1%) ⚠️	Vanderbilt (68.4%) ⚠️
#2 Tennessee vs #15 Wofford	Wofford (59.3%) ⚠️	Wofford (65.8%) ⚠️

WEST REGION

First Round - Thursday, March 20th

Matchup	Ensemble Winner (Probability)	XGBoost Winner (Probability)
#1 Auburn vs #16 Norfolk State	Auburn (76.0%)	Auburn (82.1%)
#8 Mississippi State vs #9 Baylor	Baylor (58.1%) ⚠️	Baylor (68.4%) ⚠️
#5 Memphis vs #12 Colorado State	Memphis (52.5%)	Colorado State (73.1%) ⚠️
#4 Texas A&M vs #13 Yale	Texas A&M (86.4%)	Texas A&M (89.7%)
#6 Missouri vs #11 Drake	Drake (85.3%) ⚠️	Drake (96.9%) ⚠️
#3 Texas Tech vs #14 UNC Wilmington	UNC Wilmington (66.5%) ⚠️	UNC Wilmington (73.1%) ⚠️
#7 Kansas vs #10 Arkansas	Kansas (86.7%)	Arkansas (71.8%) ⚠️
#2 St. John's vs #15 Omaha	Omaha (66.5%) ⚠️	Omaha (73.1%) ⚠️

SOUTH REGION

First Round - Friday, March 21st

Matchup	Ensemble Winner (Probability)	XGBoost Winner (Probability)
#1 Duke vs #16 Mount St. Mary's	Mount St. Mary's (51.3%) ⚠️	Duke (68.2%)
#8 Louisville vs #9 Creighton	Creighton (58.1%) ⚠️	Creighton (68.4%) ⚠️
#5 Michigan vs #12 UC San Diego	UC San Diego (59.9%) ⚠️	UC San Diego (73.4%) ⚠️
#4 Maryland vs #13 Grand Canyon	Grand Canyon (73.0%) ⚠️	Grand Canyon (88.6%) ⚠️
#6 Ole Miss vs #11 North Carolina	North Carolina (58.5%) ⚠️	North Carolina (73.4%) ⚠️
#3 Iowa State vs #14 Lipscomb	Iowa State (65.0%)	Iowa State (72.3%)
#7 Marquette vs #10 New Mexico	Marquette (70.0%)	New Mexico (71.8%) ⚠️
#2 Michigan State vs #15 Bryant	Michigan State (88.9%)	Michigan State (92.4%)

MIDWEST REGION

First Round - Friday, March 21st

Matchup	Ensemble Winner (Probability)	XGBoost Winner (Probability)
#1 Florida vs #16 SIU Edwardsville	Florida (68.7%)	Florida (72.1%)
#8 UConn vs #9 Oklahoma	UConn (88.5%)	UConn (93.7%)
#5 Oregon vs #12 Liberty	Liberty (61.4%) ⚠️	Liberty (73.4%) ⚠️
#4 Arizona vs #13 Akron	Arizona (91.7%)	Arizona (89.7%)
#6 Illinois vs #11 Xavier	Illinois (75.9%)	Illinois (79.2%)
#3 Kentucky vs #14 Troy	Troy (55.4%) ⚠️	Troy (68.5%) ⚠️
#7 UCLA vs #10 Utah State	Utah State (61.4%) ⚠️	Utah State (73.4%) ⚠️
#2 Alabama vs #15 Robert Morris	Alabama (67.7%)	Alabama (72.1%)

Key to Symbols

- ⚠️ = Upset prediction (lower seed winning)
- Probabilities shown are for the predicted winner

References

GeeksforGeeks. (2023, January 18). *A comprehensive guide to ensemble learning*. https://www.geeksforgeeks.org/a-comprehensive-guide-to-ensemble-learning/

Kaggle. (2025). *March Machine Learning Mania 2025*. Kaggle Competitions. https://www.kaggle.com/competitions/march-machine-learning-mania-2025/overview

NCAA. (2025, April 6). *Latest bracket, schedule and scores for the 2025 NCAA men's tournament*. NCAA.com. https://www.ncaa.com/news/basketball-men/mml-official-bracket/2025-04-06/latest-bracket-schedule-and-scores-2025-ncaa-mens-tournament

Scikit-learn. (n.d.). *accuracy_score*. scikit-learn. Retrieved April 23, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

Scikit-learn. (n.d.). *brier_score_loss*. scikit-learn. Retrieved April 23, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.brier_score_loss.html

Scikit-learn. (n.d.). *log_loss*. scikit-learn. Retrieved April 23, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html

Scikit-learn. (n.d.). *roc_auc_score*. scikit-learn. Retrieved April 23, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html

XGBoost Developers. (2020). *XGBoost documentation (v3.0.0)*. https://xgboost.readthedocs.io/en/release_3.0.0/