Assignment 1

Kaggle Competition

Vishal Raj
Student ID: 14227627

4th September 2023

36120 - Advanced Machine Learning Application
Master of Data Science and Innovation
University of Technology of Sydney

Table of Contents

1. Executive Summary

Introduction and Scope

The world of professional basketball is as competitive as it is captivating. From college courts to the grand arenas of the NBA, the journey of a basketball player is filled with challenges, opportunities, and career-defining moments. One such moment is the NBA Draft—an annual event where college players are selected to join the ranks of the professionals. The NBA Draft changes the lives of young athletes and shapes the future of the teams that draft them. However, the process of selecting which players will make it to the NBA is complex, influenced by numerous factors like skills, performance metrics, potential, and often, a bit of luck.

Objectives and Motivation

Given the intricacies involved in the draft process, the primary objective of this project was to simplify and enhance draft decision-making by employing the power of machine learning. The aim was to create a predictive model that could accurately determine the likelihood of a college basketball player being drafted into the NBA based on their season statistics. The motivation for this project stemmed from the need to provide a data-driven tool that could serve multiple stakeholders. An accurate predictive model for NBA teams could be the difference between drafting a future star player and missing out on one. Sports commentators and analysts could benefit from data-backed insights to enrich their discussions, predictions, and analyses during draft events. Understanding their draft likelihood could offer a focus area for improvement for college players and their coaches, increasing their chances of being drafted. Lastly, for the fans, who are always eager to know about the next big thing in basketball, these predictions can be a subject of engagement and discussion.

Methodology and Technologies

To achieve the objectives here, the project followed a structured methodology involving data collection, preparation, exploratory data analysis, feature engineering, model building, and evaluation. Two machine learning algorithms, Random Forest and XGBoost, were used to build the predictive models. Python's rich ecosystem of data science libraries, including Pandas for data manipulation, Matplotlib and Seaborn for data visualisation, and Scikit-learn for machine learning, was utilised to carry out the project. The entire codebase was maintained on GitHub, ensuring version control and collaborative development.

Results and Impact

The outcomes of the project were highly encouraging. The Random Forest and XGBoost models achieved an ROC-AUC (Receiver Operating Characteristic - Area Under Curve) score above

0.9999. This near-perfect score indicates that the model is highly effective in distinguishing between players who are likely to be drafted and those who are not. From a business perspective, the impact of these results is multifaceted. NBA teams can now make more informed decisions during drafts, potentially securing star players for their future. College players can use these insights to understand where they stand and what they need to improve to increase their draft chances. Sports analysts and commentators can enrich their discussions with data-backed insights, and fans can engage in more informed debates and discussions.

2. Business Understanding

a. Business Use Cases

The Multifaceted Impact of the Draft Process

The NBA Draft is an annual event that captures the attention of basketball enthusiasts, players, coaches, and executives worldwide. Its impact is multifaceted and reaches far beyond the apparent player-team relationships. For NBA teams, making the right draft pick can have long-term implications, affecting the team's performance for years to come. A poor choice could be a financial burden and affect the team's competitiveness, while a wise selection could turn into a franchise player that leads the team to championships.

The Need for Data-Driven Decision Making

The draft process involves a complex interplay of various factors, including player statistics, performance metrics, interviews, and even gut feelings. While some aspects of the draft are unavoidably subjective, a considerable part of the decision-making process can be made more empirical and data-driven. That's where this project comes into play by providing a machine-learning-based model to predict which college basketball players are likely to be drafted into the NBA.

Stakeholders and Scenarios

- **NBA Teams:** Teams can use the model to cross-verify their internal assessments and scouting reports, increasing the chances of making a successful draft pick.
- **Sports Analysts and Commentators:** These professionals can use the model's predictions to offer richer analyses and discussions during draft events.

- **College Players and Coaches:** Players can use the model to assess their draft prospects, helping them focus on improvement areas. Coaches can tailor training programs based on these insights.
- **Fans:** For the everyday NBA fan, the model provides an engaging way to speculate on future stars, enhancing fan discussions and engagement on social media platforms.

b. Key Objectives

Addressing the Information Gap

The project aims to bridge the information gap in the draft decision-making process. While scouts and teams have their methods for identifying talent, there needs to be a standardised, data-driven approach that can predict draft outcomes accurately.

Identifying Stakeholder Requirements

- **For NBA Teams:** The requirement is a reliable model that can predict draft likelihood based on quantifiable metrics.
- **For Analysts:** The need is for a tool that can add empirical weight to their analyses.
- **For Players and Coaches:** A model that can identify areas for improvement based on what metrics are most indicative of a draft pick.
- **For Fans:** A tool that can fuel informed discussions and speculations.

Meeting Stakeholder Needs through Machine Learning

By developing a predictive model using machine learning algorithms, this project aims to offer a tool that serves these varied stakeholder requirements. The machine learning model will utilise player statistics from their college basketball careers to predict their likelihood of being drafted into the NBA. These predictions will provide actionable insights for all stakeholders involved in or affected by the NBA Draft.

3. Data Understanding

Dataset Overview: The Foundation of Predictive Modeling

The dataset used for this project is a comprehensive collection of statistics related to college basketball players, capturing various facets of their performance. The data offers rich features, including points scored, assists, rebounds, and other advanced metrics like Effective Field Goal

Percentage(eFG). The dataset is the cornerstone for building a predictive model, fulfilling the project's objective of providing data-driven insights into NBA draft selections.

Data Sources: Where Quality Meets Quantity

The data was sourced from multiple datasets meticulously curated to include relevant metrics. These datasets include train.csv, test.csv, sample_submission.csv, and metadata.csv. Each dataset serves a specific purpose:

- **Train.csv:** Contains the historical data, including player statistics and whether they were drafted.
- **Test.csv:** Similar to the training set but lacks the draft outcome used for model validation.
- **Sample_submission.csv:** Provides a template for submitting predictions on Kaggle.
- **Metadata.csv:** Includes additional information that might be useful for exploratory data analysis.

Data Inspection: A First Look

Initial data inspection uses the 'display_head' function, providing a snapshot of the data's structure. This quick overview serves as a preliminary check to identify glaring issues such as missing values, irrelevant columns, or inconsistent data types requiring immediate attention.

Features and Their Significance

- **Team and Conference:** The dataset contains information about the player's team (team) and conference (conf). This is crucial as the level of competition varies across different conferences.
- **Games Played (GP):** The total number of games a player has participated in during the season. It provides context to the other statistics, showing durability and experience.
- **Offensive and Defensive Metrics:** Features such as ORtg (Offensive Rating) and DRtg (Defensive Rating) are pivotal in understanding a player's efficiency on both ends of the court. They estimate how many points a player produces or allows per 100 possessions.
- **Shooting Metrics:** Shooting efficiency is gauged using metrics like eFG% (Effective Field Goal Percentage) and TS% (True Shooting Percentage). These stats factor in 2-pointers, 3-pointers, and free throws, giving a nuanced view of a player's shooting prowess.
- **Rebounding:** ORB% and DRB% indicate a player's prowess in grabbing offensive and defensive rebounds, respectively, relative to the opportunities available during their playing time.
- **Ball Handling:** AST% gives an estimate of the percentage of teammate field goals a player assisted on, while TO_per gauges a player's tendency to turn over the ball.

- **Defensive Playmaking:** stl_per and blk_per offer insights into players' ability to disrupt the opponent's offence through steals and blocks.
- **Advanced Metrics:** Features like bpm, obpm, and dbpm estimate a player's contribution above the league average per 100 possessions, split by overall, offensive, and defensive contributions. These metrics are invaluable for gauging a player's overall impact on the game.
- **Physical Attributes:** ht signifies a player's height, a vital attribute, especially for positions like centres and power forwards.
- **Seasonal Context:** The year and type columns provide context about the specific season and the type of games played, respectively.
- **Other Metrics:** There are various other metrics, such as assist-to-turnover ratio (ast_tov), shots made near the rim (rimmade), and dunks (dunksmade), which provide more granular insights into a player's game.

Data Limitations: Challenges and Workarounds

Every dataset has its limitations, and this one is no exception:

- **Missing Values:** Columns like 'pick' have a high percentage of missing values and are thus dropped from the dataset.
- **Imbalanced Data:** The target variable 'drafted' is imbalanced, with fewer instances of undrafted players.
- **Feature Scaling:** Some algorithms require features to be on the same scale, necessitating preprocessing steps.

We address these limitations through various data preprocessing techniques like imputation, SMOTE for balancing classes, and feature scaling.

Exploratory Data Analysis (EDA)

To better understand the data, we perform extensive EDA. This includes displaying first five rows of the datasets with the help of ‘display_head’ function, analysing the distribution of the target variable 'drafted' using the ‘target_distribution’ function and identifying correlations between features through random forest inbuilt feature. We also plotted some selected feature to get an overview of the data. The insights from EDA help in data cleaning and feature engineering, thus shaping the modelling strategy.

4. Data Preparation

Initial Data Loading: The Starting Point

The first step in data preparation involves loading multiple datasets using the load_data function from the make_dataset.py module. This function ingests data from files such as train.csv, test.csv, sample_submission.csv, and metadata.csv, each serving specific roles in the project. While this seems straightforward, it's a crucial phase that sets the stage for all subsequent steps.

Missing Value Analysis: The Art of Data Cleaning

One of the fundamental challenges in any data science project is handling missing values. We used the 'missing_values_analysis' function to identify and quantify columns with missing values. This function generates a data frame that displays the percentage of missing values for each feature. Based on this analysis, we dropped columns like 'pick' with a high percentage of missing values, while others are attributed using statistical methods.

Data Imputation: Filling in the Gaps

For the remaining columns with missing values, we employ the DataPreprocessor class, which is in data_preprocessor.py. This class fills in missing values using their median for numeric columns and mode for non-numeric columns. This strategy ensures that every data point is retained due to missing information, maximising the utility of the available data.

Feature Engineering: Crafting the Data

Feature engineering is the backbone of any successful machine learning model. In this project, we use a multitude of techniques:

- **Temporal Transformation:** We transform the year using sine and cosine functions to capture its cyclical nature.
- **Polynomial and Interaction Features:** New features like GP_Min_per and Ortg_eFG are introduced to capture complex relationships between existing features.
- **Handling Outliers:** Outliers are identified using the Z-score method and removed to ensure model robustness.

Class Imbalance: Leveling the Playing Field

Imbalanced classes can introduce a bias in machine learning models. In this dataset, we use the 'apply_smote' function to generate synthetic samples for the minority class using the Synthetic Minority Over-sampling Technique (SMOTE), thus balancing the classes.

Data Encoding and Scaling: Preparing for the Algorithm

We performed Data encoding by converting categorical columns into a format suitable for machine learning algorithms. Functions like 'encode', 'scale' and 'encode_and_scale' are employed to one-hot encode categorical variables and scale the features using the 'StandardScaler'.

Data Splitting: Training and Validation Sets

The final step in data preparation involves splitting the data into training and validation sets. The 'train_test_split' function is utilised, ensuring that the model has a fresh set of data for validation, separate from the training data.

Test Data Preparation: Consistency is Key

The test data is also subjected to a similar treatment just as the training data is prepared. The 'encode_and_scale_test' & 'encode_and_scale_test_features' functions ensure that the test data has the same features as the training data and is scaled appropriately.

5. Modelling

Algorithm Selection: Random Forest and XGBoost

The project employs two types of classifiers: Random Forest and XGBoost. We selected these algorithms for specific reasons:

- **Random Forest:** Known for its robustness and ability to handle large, high-dimensional datasets, Random Forest is an ensemble method that offers high interpretability and feature importance metrics.
- **XGBoost:** Renowned for its speed and performance, XGBoost is an advanced implementation of gradient-boosted decision trees. It has an in-built ability to handle missing values and offers a variety of tuning parameters.

Hyperparameter Tuning: Fine-Tuning for Performance

Hyperparameter tuning is an essential step in model building. The 'grid_search' method in the Model class employs 'GridSearchCV' to fine-tune the hyperparameters for both classifiers. We tuned parameters like 'n_estimators', 'max_depth', 'min_samples_split', and 'bootstrap' for Random Forest. We also tuned XGBoost parameters such as 'learning_rate', 'n_estimators', and

'max_depth'. The best models are selected based on their ROC-AUC scores, ensuring they are tuned to optimise for the specific evaluation metric that matters the most for this project.

Model Training: Learning from the Data

After hyperparameter tuning, the 'train' method is invoked to train the model on the entire training set, and then we train two sets of models:

- **Without Feature Selection:** The first training phase includes all available features to explore the raw predictive power of the data.
- **With Feature Selection:** The second training phase employs a more data-driven approach, using only those features deemed most important by the feature importance metrics.

a. Approach 1: Random Forest without Feature Selection

Algorithm and Key Hyperparameters

- **Algorithm:** Random Forest
- **Key Hyperparameters:** For this initial approach, we trained the Random Forest model using optimised values for 'n_estimators', 'max_depth', 'min_samples_split', and other relevant parameters.

Preprocessing and Feature Engineering Specifics

We also performed the initial preprocessing steps, such as handling missing values and scaling. However, this approach did not employ specific feature selection based on feature importance metrics.

Training Process

We trained the model on a resampled dataset using techniques like 'SMOTE' to handle the class imbalance, ensuring a balanced learning process. This method ensures that the model gets a balanced data view and does not favour the majority class.

b. Approach 2: Random Forest with Feature Selection

Algorithm and Key Hyperparameters

- **Algorithm:** Random Forest
- **Key Hyperparameters:** We again trained the model using the Random Forest algorithm but with hyperparameters further optimised to account for the refined feature set.

Preprocessing and Feature Engineering Specifics

- This approach employed a more data-driven strategy. Features were selected based on their importance as determined by the Random Forest algorithm.
- Additionally, features that strongly correlated with others and did not add unique information were removed to prevent multicollinearity.

Training Process

The training process remained consistent, using the resampled dataset to address class imbalance. The significant difference here was the reduced and more focused feature set, allowing the model to hone in on the most predictive aspects of the data.

c. Approach 3: XGBoost with and without Feature Selection

Algorithm and Key Hyperparameters

- **Algorithm:** XGBoost
- **Key Hyperparameters:** Parameters such as 'learning_rate', 'n_estimators', 'max_depth', and others were fine-tuned to ensure optimal performance for both the full and the reduced feature sets.

Preprocessing and Feature Engineering Specifics

- We trained the model on the full and the reduced feature sets (based on feature importance metrics).
- We introduced interaction features like 'GP_Min_per' and 'Ortg_eFG', along with applying temporal transformations to capture the cyclical nature of certain features.
- These additions aimed to extract even more information and predictive power from the data.

Training Process

As with the Random Forest models, the XGBoost model was trained on a resampled dataset to handle class imbalance. This ensured the model got a comprehensive data view without bias towards the majority class.

Model Persistence: Future-Proofing the Work

After the training and validation phases, the best Random Forest and XGBoost models are saved using Joblib. This enables easy retrieval and deployment for future predictive tasks, offering a plug-and-play solution for stakeholders.

6. Evaluation

a. Evaluation Metrics: ROC-AUC as the Gold Standard

In this project, the Receiver Operating Characteristic Area Under the Curve (ROC-AUC) score serves as the primary metric for model evaluation. The choice of ROC-AUC is deliberate for several reasons:

- **Discriminative Power:** The ROC-AUC score measures the model's ability to distinguish between players likely to be drafted and those not, which is critical for our business objective.
- **Imbalanced Data:** Given the imbalanced nature of the 'drafted' target variable, ROC-AUC proves to be more informative than accuracy, as it considers both sensitivity and specificity.
- **Stakeholder Requirements:** ROC-AUC aligns well with the needs of various stakeholders, including NBA teams and sports analysts, who are interested in the model's discriminative capability rather than mere accuracy.

b. Results and Analysis: Exceptional Performance

Both Random Forest and XGBoost models demonstrated extraordinary performance, as per the stakeholder and other requirements, and to keep it simple, only ROC-AUC was used as the evaluation metric:

- **Without Feature Selection:**

Random Forest achieved a ROC-AUC score of 0.999986132207113.

XGBoost scored 0.9999719323000272 on the ROC-AUC metric.

- **With Feature Selection:**

Random Forest's performance improved to a ROC-AUC of 0.9999896602113071.

XGBoost also benefited, scoring 0.9999779766233615.

Performance Comparison

When comparing both models, we find that:

- **Random Forest:** Shows a minor improvement in ROC-AUC after feature selection.
- **XGBoost:** Also displays a similar trend, further validating the model's efficacy.

Key Insights

While the ROC-AUC scores are remarkably high for both models, it's crucial to exercise caution. High scores may indicate overfitting, which could compromise the model's performance on new, unseen data.

c. Business Impact and Benefits: Revolutionising Draft Decisions

The Game-Changing Model

Here, the ability to spot talent early can be a game-changer. With our predictive models, NBA teams have the power to leverage data-driven insights like never before:

- **Strategic Drafting:** Teams can look beyond intuition and personal judgment, incorporating data-backed predictions into their draft decisions. This might be the difference between spotting a future MVP or overlooking one.
- **Resource Allocation:** NBA teams invest significant resources in scouting. Our model can streamline these efforts, allowing teams to allocate scouts more efficiently, focusing on players more likely to be drafted.
- **Contract Negotiations:** Understanding a player's potential could also be instrumental during contract negotiations, providing teams with an edge in securing favourable terms.
- **Fan Engagement:** In the age of fantasy sports, fans are more engaged than ever. A model that predicts NBA drafts could also be marketed as a tool for fans, enhancing their engagement with the sport.

Implications for Other Stakeholders

While NBA teams stand to benefit immensely, they aren't the only stakeholders:

- **Sports Analysts:** With the model's insights, analysts can bolster their discussions with data-backed predictions, enhancing the quality of draft event analyses and commentaries.
- **Players and Coaches:** The model serves as a reality check for college players and their coaches. Knowing their draft likelihood can guide their training regimes, focusing on areas that could enhance their draft chances.
- **Merchandisers and Advertisers:** Anticipating a player's entry into the NBA could also be valuable for merchandisers and advertisers, allowing them to align their strategies to capitalise on a player's growing popularity.

d. Data Privacy and Ethical Concerns: Treading Carefully

Navigating the Data Minefield

In an era where data privacy is paramount, it's crucial to approach any data-driven project with sensitivity:

- **Consent:** All data used in this project is publicly available and pertains to players' on-court performance. However, the ethical gathering of data requires that players, especially those in college, know their performance metrics are being used for such analyses.
- **Misplaced Predictions:** While our models have demonstrated high accuracy, every model is infallible. Erroneous predictions could unfairly influence perceptions about a player's potential, possibly affecting their career trajectory.

Balancing Utility with Responsibility

As data scientists, while we harness the power of data, we also shoulder the responsibility of ethical application:

- **Transparency:** Being transparent about the model's capabilities and limitations is crucial. Overselling its accuracy could lead to misplaced trust and consequent repercussions.
- **Feedback Mechanism:** Establishing a feedback mechanism where predictions are regularly validated against actual draft outcomes can help refine the model and ensure its accuracy.
- **Bias and Fairness:** We must also be vigilant about any biases in the data. A model trained on biased data could perpetuate existing stereotypes, leading to unfair predictions.
- **Ethical Deployment:** Even if the model is accurate, its deployment should be ethical. For instance, if used by NBA teams, it should complement, not replace, traditional scouting to ensure a holistic player assessment.

7. Deployment

Deployment Process: Model Serialization and GitHub Integration

The journey from experimentation to deployment for our Random Forest and XGBoost models involves several crucial steps:

- **Model Serialisation:** After the intensive training process, both Random Forest and XGBoost models were serialised using Python's 'joblib' library. This method facilitates the reloading of the models effortlessly for future predictions without the need to retrain.
- **Modularisation of the Code:** We structured the project in a modular fashion, developing separate modules for data cleaning, data preprocessing, training, prediction, and visualisation. Using classes and functions in these modules ensures ease of maintenance and better readability.
- **GitHub Integration:** The entire project was pushed to GitHub, making it publicly available. This approach allows for collaborative development, version control, and easy sharing. Any interested party, be it other data scientists or NBA teams, can fork the repository, study the models, and even make improvements or adjustments as they see fit.

Real-world Considerations: Ensuring Robustness and Reusability

- **Data Versioning:** New data will become available over time as new college basketball seasons unfold. Using platforms like DVC (Data Version Control) alongside GitHub could be recommended for future endeavours to track changes in data and ensure the reproducibility of results.
- **Scalability:** The modular approach to the code ensures that, as more data becomes available or as adjustments to the model are needed, changes can be made efficiently without disrupting the entire system.
- **Security:** Since the repository is public on GitHub, it's essential to ensure that no sensitive information or credentials are committed. Using '.gitignore' files and ensuring a review process before commits can help maintain security.
- **Documentation:** Extensive documentation on the GitHub repository, including README files and inline comments, ensures that anyone forking or reviewing the repository can understand and utilise the models and codes effectively.

Recommendations for Future Deployment

- **Continuous Integration and Testing:** Integrating CI/CD pipelines like GitHub Actions or Travis CI can ensure that future codebase changes do not introduce bugs. Automated testing can help maintain the integrity of the models and the preprocessing steps.

- **Collaboration:** Encouraging collaboration through the GitHub platform can lead to improvements in the models or codebase. Open-source collaboration can provide fresh perspectives and introduce novel techniques to enhance the project's performance and efficiency.
- **Feedback Loop:** Establishing a feedback mechanism where users of the models can report any inconsistencies or provide suggestions can be beneficial. This feedback can be used to refine the models further and make them more robust.

8. Conclusion

Summarising Key Findings and Insights

After several weeks of intensive data cleaning, feature engineering, modelling, and evaluation, our project aimed at predicting whether a college basketball player will be drafted into the NBA has yielded exceptional results. The Random Forest and XGBoost models achieved ROC-AUC scores close to 1, indicating near-perfect classification. Feature engineering played a crucial role, adding depth and complexity to our predictive algorithms. The models were validated to ensure they were robust and resistant to overfitting, giving us confidence in their predictive power.

Reflecting on the Project's Success

The project successfully met its objectives and exceeded expectations in several key areas. It achieved its primary goal of developing a highly accurate predictive model. The various stakeholders, from NBA teams and sports analysts to college players and their coaches, can significantly benefit from these predictive insights. The model offers a data-driven approach to draft selections for NBA teams, potentially uncovering hidden talents. Sports analysts can now back their commentary with quantitative data. Players and coaches can focus their training regimes on improving draft prospects, effectively fulfilling stakeholder requirements.

Meeting Stakeholders' Requirements

Not only did the project yield a technically proficient model, but it also provided actionable insights for different stakeholders. The business impact of these models could revolutionise the way draft decisions are made, providing a data-backed method of assessing player potential. This aligns well with the stakeholder requirements for an accurate, reliable, and actionable tool for draft prediction.

Future Work, Recommendations, and Next Steps

Despite the model's exceptional performance, there is always room for improvement and additional features to consider:

- **Temporal Features:** Future work could focus on incorporating more temporal data to capture a player's performance trajectory over multiple seasons.
- **Model Diversification:** Exploring other algorithms like Neural Networks or Gradient Boosting Machines could offer alternative perspectives on the data.
- **Real-world Deployment:** The next logical step is to deploy the model in a real-world setting, as discussed in the "Deployment" section, allowing for more extensive validation of the model's performance.
- **Continuous Learning:** As each NBA draft season provides a wealth of new data, the models should undergo regular updates to maintain their accuracy.
- **Ethical and Regulatory Compliance:** As the model moves towards real-world application, ethical considerations such as data privacy and fairness must be rigorously examined.

9. References

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

O'Connor, K. (n.d.). The Ringer's 2023 NBA draft guide. The Ringer's 2023 NBA Draft Guide. https://nbadraft.theringer.com/

Albon, C. (2018). Machine learning with python cookbook. O'Reilly Online Learning. https://www.oreilly.com/library/view/machine-learning-with/9781491989371/ch04.html

Kuhn, M., & Johnson, K. (2019). Feature Engineering and Selection: A Practical Approach for Predictive Models (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781315108230

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

Brownlee, J. (2020, August 17). Ordinal and one-hot encodings for Categorical Data. MachineLearningMastery.com. https://machinelearningmastery.com/one-hot-encoding-for-categorical-

data/

György Kovács, Smote-variants: A python implementation of 85 minority oversampling techniques, Neurocomputing, Volume 366, 2019, Pages 352-354, ISSN 0925-2312. https://doi.org/10.1016/j.neucom.2019.06.100

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874. https://doi.org/10.1016/j.patrec.2005.10.010

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The Elements of Statistical Learning. Springer Series in Statistics. https://doi.org/10.1007/978-0-387-21606-5

Han, J., Pei, J., & Kamber, M. (2012). Data Mining: Concepts and Techniques. Morgan Kaufmann. https://doi.org/10.1016/C2009-0-61819-5

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer Texts in Statistics. https://doi.org/10.1007/978-1-4614-7138-7

Provost, F., & Fawcett, T. (2013). Data Science for Business. O'Reilly Media, Inc. https://www.researchgate.net/publication/256438799_Data_Science_for_Business

Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann. https://doi.org/10.1016/C2009-0-19715-5

Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). Fundamentals of Machine Learning for Predictive Data Analytics. MIT Press. https://mitpress.mit.edu/books/fundamentals-machine-learning-predictive-data-analytics

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The Elements of Statistical Learning. Springer Series in Statistics. https://doi.org/10.1007/978-0-387-21606-5

Schapire, R. E., & Freund, Y. (2012). Boosting: Foundations and Algorithms. MIT Press. https://doi.org/10.7551/mitpress/8291.001.0001