EXPERIMENT REPORT

Student Name	Kritika Dhawale
Project Name	AT1 - Kaggle Competition Week 3
Date	30th August 2023
Deliverables	dhawale_kritika-24587661-week3_B_rand_forest.ipynb *`nba_draft_prediction.joblib`* *`scaler_w3b.joblib`* + *interim data files* + *preprocessed data files*
Github Link	[adv_mla_at1]
Kaggle Link	[kritz_23]

EXPERIMENT BACKGROUND	
Provide information about the problem/project such as the scope, the overall objective, expectations. Lay down the goal of this experiment and what are the insights, answers you want to gain or level of performance you are expecting to reach.	
1.a. Business Objective	- The goal is to provide NBA teams, sports commentators, and fans with a reliable predictive model that evaluates the likelihood of college basketball players being drafted into the NBA based on their current season's performance statistics. - The model aims to provide valuable insights into player potential by leveraging machine learning, assisting teams in making informed draught decisions and fostering a data-driven approach to talent selection.
1.b. Hypothesis	- Does the current season performance metrics of college basketball players have a discernible impact on their likelihood of being drafted into the NBA? - It is expected that by analysing these statistics, a predictive model that effectively distinguishes between players who are drafted and those who are not will be developed. - The hypothesis assumes that the model's predictive power will outperform random chance, as demonstrated by an AUROC score greater than 0.5. - Furthermore, the hypothesis predicts that certain key metrics will emerge as important factors in the draft selection process. - Can synthesizing the raw data (for players being drafted) help in making better predictions?

1.c. Experiment Objective

- A selective list of factors from given 63 features in the dataset that have a constructive relationship with the target “**drafted”**.
- Balance the dataset with upsampling techniques.
- Create and test a classification model that fits the selected data accurately with a meaningful level of predictive accuracy, as measured by the AUROC score, so that we can predict the players that can be drafted in the NBA on certain given factors.
- Possible scenarios:
    1. The model achieves a high AUROC score (>0.90) and successfully differentiates between drafted and undrafted players. Key player performance metrics that influence draft selection are identified, providing teams and analysts with actionable insights.
    2. The model's AUROC score remains near 0.5, indicating difficulties in accurately predicting expected outcomes. If the model is not accurate enough, it cannot be relied on for decision-making for business aspects.
    3. There might be several other factors that affect the probability of being drafted, so we can’t state a fact based on this given dataset.
    4. Several **ethical concerns** can be raised such as bias and discrimination that can have an impact on certain groups of individuals or society as a whole. The model may perpetuate biases if certain groups of players are systematically favored or disadvantaged.

EXPERIMENT DETAILS

Elaborate on the approach taken for this experiment. List the different steps/techniques used and explain the rationale for choosing them.

2.a. Data Preparation

***Includes feature engineering**

Part A:

Data Cleaning:

- Removed 32162 duplicates from the data (repeated player IDs)
- Removed non-important features (which are mostly categorical).
- Filled Nan values with 0.
- Repeated the same steps for test set.

Feature Engineering:

- `ht` column was in wrong format (date). Ex: 05-Jun meant 6’5. Month represents foot and date inches. Converted date format to numeric handling all exceptional cases through a function `convert_height_numeric`. Calculated height in inches and saved in a new column `ht_inch`.
- Dropped the original height column `ht`

Part B:

Read the interim data saved in Part A and performed Synthetic Minority Oversampling Technique (SMOTE)

- Applied oversampling using SMOTE on the minority target variable (1.0) as there were only 176 samples of players being drafted out of 23929.

Data Preprocessing:

- Separated numerical columns to perform standard scaling on numerical

features.
- Repeated the same steps for test set and dumped the scaler model.

Data Split:
- Split the train set into train and val with a ratio of 30% (used stratify to create unbiased split and shuffled before split)
- Shape check and saved the preprocessed sets in a separate folder.

2.b. Modelling

On the balanced dataset, three models - Logistic Regression, Random Forest and Adaboost were trained to see if upsampling has really improved the results. Reason to use these models were to see if the data exhibits linear or non-linear relationships and does ensembling help in improving the performance.

Part A: Logistic Regression (Basic algorithm)

Hyperparameters: As the data was large, had to maximize the number of iterations (default was 100). Set it to 4000.
- AUROC on validation and test set (through kaggle)

Part B: Random Forest (automatically discovers and models complex patterns, control over-fitting)
- AUROC on validation and test set (through kaggle)

Part C: Adaboost (fits whole data and copies of subsets with weight adjustments for incorrectly classified instances to learn better in future. Represented by learning rate)
- **Hyperparameters:** changed the learning rate to 0.05 (so that the model slowly moves towards the optimal weight and do not miss it by taking a large jump, default = 1)
- AUROC on validation

Models that I didn't use:

1. Gaussian Process Classifier
   - Training time - 372.658 seconds was high

Future models that can be used: Analyze feature importance through random forest feature importance method to understand which statistics contribute most significantly to predicting draft selection. Furthermore, we might consider employing more sophisticated models like Multinomial or Polynomial Logistic Regression as there are a lot of features involved.

EXPERIMENT RESULTS

Analyse in detail the results achieved from this experiment from a technical and business perspective. Not only report performance metrics results but also any interpretation on model features, incorrect results, risks identified.

3.a. Technical Performance

Part A: Logistic regression model
- Val set - **0.9964**
- Test set - **0.9952**

Part B: Random Forest model
- Val set - **0.99997**
- Test set - **0.99967**

	Part C: Adaboost model - Val set - **0.9977** There is a high improvement in the AUCROC scores from last week's experiment. One of the main reason is performing SMOTE and balancing the data. The **random forest** model performs exceptionally well on both the validation and test sets, hence is picked as the **best model**. It achieves extremely high accuracy, which suggests that it's capturing complex relationships in the data. Nevertheless, it is important to check if the model's performance is consistent across different unseen datasets or if it's mainly memorizing the training data. Not to mention, we achieved the the first possible scenario.
3.b. Business Impact	This could be presented to our stakeholders as we have achieved astonishing results on both the sets. The kaggle leaderboard also shows that this is the **best result** amongst all other teams. This final model can prove to be quite useful to fulfill the business objectives mentioned above.
3.c. Encountered Issues	Only issue faced this time was calling function from another file in another directory. I was using the relative calling (..) method in the jupyter notebook. Solved: used **sys.append** method to tackle it.

FUTURE EXPERIMENT	
Reflect on the experiment and highlight the key information/insights you gained from it that are valuable for the overall project objectives from a technical and business perspective.	
4.a. Key Learning	- Data Processing: We learned how to handle missing values and outliers. Scaling numerical values and imputing missing values are essential for data quality and model performance. Feature engineering improved the model performance. - Feature Engineering: Balancing the data with respect to the target variable using SMOTE certainly helped. - Model Selection: Logistic regression was our baseline model. It performed well, but ensemble models like random forest and adaboost gave better results than the baseline model. - Model Evaluation: The AUROC metric clearly measured the model's predictive power. - There are still chances for more research and experimentation, though. For instance, feature selection: We can definitely look into feature importances using random forest technique. We could also look into additional variables for ex: teamwork, personality, and coachability that influence decisions about a player being drafted. Overall, our current strategy has delivered insightful information, but there is still room for investigation and experimentation.

4.b. Suggestions / Recommendations	1. Feature engineering can be investigated to produce more pertinent features that will assist the model to perform better by better capturing the variation in the data. Deployment Scope: Random forest model is **ready** for deployment and can be proceeded by keeping a few things in mind: 1. Ensuring data quality: This can be achieved by doing data cleansing when required and running routine data quality checks. 2. Optimising the model: Refining and optimizing the model created originally.. 3. Deploying the pipeline: A user-friendly interface, continuous monitoring, and retraining are essential for team members. With the model complementing their expertise, analysts, scouts, and decision-makers must collaborate. 4. After the pipeline has been installed, it is essential to constantly check on its operation and take care of any problems as soon as they appear. To match dynamic player performance and team strategies, the model should be deployed iteratively with user feedback and continuous improvement.

Appendix

1. McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).

2. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

3. Waskom, M., Botvinnik, Olga, O&#x27;Kane, Drew, Hobson, Paul, Lukauskas, Saulius, Gemperline, David C, … Qalieh, Adel. (2017). mwaskom/seaborn: v0.8.1 (September 2017). Zenodo. https://doi.org/10.5281/zenodo.883859

4. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science &amp; Engineering, 9(3), 90–95.

5. Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

6. Van Rossum, G. (2020). The Python Library Reference, release 3.8.2. Python Software Foundation.

7. Joblib Development Team (2023) Joblib/joblib: Computing with python functions., GitHub. Available at: https://github.com/joblib/joblib.