# **EXPERIMENT REPORT**

|  |  |
| --- | --- |
| **Student Name** | Tahmidul Islam |
| **Project Name** | Adv MLA AT1 |
| **Date** | 7/10/2023 |
| **Deliverables** | Notebook:  Islam\_Tahmidul-24587139-Week2\_SVM.ipynb  Model:  Support Vector Machine  Git Repo: https://github.com/tahmislam21/adv\_mla\_at1 |

|  |  |
| --- | --- |
| 1. **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 of this project for the business is to create a predictive model that evaluates the probability of a college basketball player being drafted into the NBA based on their current season's statistics.  The results will be invaluable in several ways. NBA teams can make more informed draft choices, potentially improving team performance. Scouts and college coaches can identify promising talents more effectively.  Accurate predictions enhance resource allocation, save time, and optimize player development programs.  Conversely, incorrect results could lead to missed opportunities for both players and NBA teams, affecting talent scouting and team competitiveness, highlighting the critical importance of model accuracy in this context. |
| **1.b. Hypothesis** | The Hypothesis of this experiment is that Support Vector Machine Model will give us a high ROC Accuracy Score, and also generate a AUROC Graph that is close to the inverse-L shaped graph.  It is worth considering because SVMs provide flexibility in controlling regularization strength through the choice of the regularization parameter (C), allowing for the trade-off between model bias and variance. |
| **1.c. Experiment Objective** | The expected outcome is that the model will have a roc score close to 1. The goal is to select the different hyper parameters, to understand which one works better.  The possible outcomes are that Model 2 will perform better in both Training and Validation set than Model 3. The opposite could also be true for Model 2, but both Training and Validation scores will perform worse if that is the case. |

|  |  |
| --- | --- |
| 1. **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** | To start off with, columns 'Rec\_Rank', 'pick', 'dunks\_ratio', were removed because of having too many missing values and columns 'team', 'conf', 'ht', 'yr', 'type','num' and 'player\_id' were removed because they had nominal values.  I decided to not perform any encoding on those nominal columns because they had a high number of unique values. Thus, they would give rise to an excessive amount of extra columns.  Following this, imputations were performed on the remaining null values with mean() values of the columns.  The data were then Scaled/Standardized.  Feature Scaling may be important for future experiments, otherwise future models will not generalise well to the dataset. |
| **2.b. Feature Engineering** | No new features were generated. Rather, PCA was applied to the standardized data to reduce the number of features. There were a total of 60+ rows, and having them all under the running model would lead to overfitting.  The features that were removed and the rationale are mentioned in 2.a. |
| **2.c. Modelling** | Model 1:  SVM (Default Parameters)  This model was chosen to capture potential nonlinear relationships between the input features and the binary outcome (drafted or not drafted). The default parameters are used for an initial run to gauge model performance.  Model 2:  SVM (class\_weight = ‘balanced’)  By selecting this parameter, the algorithm assigns higher weights to the minority class and lower weights to the majority class during training. This means that errors on the minority class have a more significant impact on the model's loss function, encouraging the model to pay more attention to the minority class and make better predictions for it. |

|  |  |
| --- | --- |
| 1. **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** | Model 1:  Train ROC-AUC Score: 0.939  Validation ROC-AUC Score: 0.641  Test: *(Target variable was not available in test dataset)*  Model 2:  Train ROC-AUC Score: 0.995  Validation ROC-AUC Score: 0.847  Test: *(Target variable was not available in test dataset)*  For Model 1 (Default ), the SVM model achieved a higher Score on  the training set (0.939 but low score on the validation set (0.641) . This indicates that the model is over fitting on the training data and failing to generalize on the validation set.  For Model 2 (class\_weight = balanced ), the SVM model achieved a similar high Score on the training set (0.995) and relatively higher than model 1 on the validation set (0.847). This indicates that model 2 doing a better job because of balancing the imbalanced data set. |
| **3.b. Business Impact** | Based on the technical results, it can be found that SVM can be a good predictor of drafting the basketball players into the team. But ofcourse, this can happen once there is balance in class in the dataset.  Accurate predictions enhance resource allocation, save time, and optimize player development programs.  Conversely, incorrect results could lead to missed opportunities for both players and NBA teams, affecting talent scouting and team competitiveness, highlighting the critical importance of model accuracy in this context. |
| **3.c. Encountered Issues** | Imbalanced Dataset: There is evident imbalance present in the dataset for which the model was overfitting. This problem was approached by assigning higher weight to the less abundant class and lower weight to the more abundant.  Missing values:  This was handled in the previous experiment and the prepared data was imported for this experiment.  Too many features: The dataset had too many features (60+) which can make models overfit. Thus feature selection or Dimensionality Reduction may be applied to resolve this problem.  Future experiments: The results from this experiment provide great motivation to tune some more parameters with SVM after class balancing. Thus future experiments are designed to tune regularization parameters to see if improvements can be made. |

|  |  |
| --- | --- |
| 1. **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** | The SVM model showed good promise on predicting the probabilities to draft the players, after balancing class weights.  However, this model can be be improved by tuning regularisation parameters to further avoid overfitting. Further experiments can be designed to work with models that are responsive to this proposal. |
| **4.b. Suggestions / Recommendations** | Given the results achieved and the overall objective of the project, list the potential next steps and experiments. For each of them assess the expected uplift or gains and rank them accordingly. If the experiment achieved the required outcome for the business, recommend the steps to deploy this solution into production.  Rank 1:  Training dataset with SVM, having Regularisation Parameter C = 0.5: We can understand if the model is really overfitting with the Training dataset and increase the scores in the Validation.  Uplift: The uplift is that accuracy scores will be higher and more accurate probabilities will come out.  Rank 2:  Training dataset with SVM, having Regularisation Parameter C = 0.8:  Based on what results we see when C = 0.5, we can really understand if setting C = 0.8 gives us a better result.  Uplift: The uplift is we will know how to optimize our algorithm to understand which value works better.  Model Deployment:  If model is performing well, the steps to deployment are as follows:  (Train, evaluate, and serialize the model) -> Containerize -> Deploy to Production -> Implement Load Balancing -> Monitor performance -> Provide Documentation. |