**Random Forest:**

The Random Forest Classifier was used to initially train with the default parameters to serve as a baseline. This was crucial for establishing a performance benchmark before trying to tune it up. The Random Forest model was chosen because of its robustness and capability of handling large complex data with little preprocessing steps. Multiple decision trees are created to reduce overfitting and improve the overall prediction accuracy. The results of the model were quite good from the beginning.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| 0 | 0.99 | 1.00 | 0.99 | 47759 |
| 1 | 0.99 | 0.67 | 0.80 | 1505 |

The ‘GridSearchCV’ was used for hyperparameter tuning with a focus on the ‘n\_estimators’ and ‘max\_depth’, to find the optimal model configuration. The process systematically varied key parameters of the Random Forest model and evaluated their impact on the model performance. Varying the number of the trees was helpful for obtaining better model performance but was detrimental due to the computational power required to run the operation. It also ran the risk of overfitting with higher values. Max depth was another variable that was used to try and capture more complex patterns but also ran into a risk of overfitting. In the end the ‘GridSearchCV’ resulted in very similar but worse results. The computation time was also another factor in removing it from being processed again. Accuracy dropped to .66 from .67 and was overall not worth the time invested.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Best Score: 0.9411435671846787 | | | | |
|  | Precision | Recall | F1-Score | Support |
| 0 | .99 | 1.00 | .99 | 47759 |
| 1 | .99 | 0.66 | 0.79 | 1505 |

The tuning was based on the ROC AUC score which was chosen as it provided a comprehensive measure of the model’s ability to decipher whether an injury occurred or not across all possible classifications. Though the ‘GridSearchCV’ was not useful after first use, it did give some interesting looks into the trees that were being produced by the Random Forest. Thus, allowing us to understand how complex the question and predictions were. One of the main findings from this was that it required a robust model that was capable of learning from a very imbalanced and diverse set of data.

|  |  |  |
| --- | --- | --- |
| Random Forest Importance Factors | | |
|  | Feature | Importance |
| 0 | injury\_frequency | 0.213277 |
| 1 | position\_DB | 0.068319 |
| 2 | pass | 0.062914 |
| 3 | play\_type\_pass | 0.062619 |
| 4 | position\_DL | 0.062147 |
| 5 | first\_down | 0.059360 |
| 6 | pass\_attempt | 0.055217 |
| 7 | season | 0.044257 |
| 8 | position\_OL | 0.041927 |
| 9 | position\_LB | 0.034099 |

The selected hyperparameters fully demonstrated the trade-off between computational cost and performance. The Random Forest was chosen for this reason, and it was apparent in the final product. The Random Forest scored better than the Logistic Regression as well as the Naïve Bias models. It is important to note that the Precision was exceptionally high, showing the model’s effectiveness in correctly identifying true injury events. Unfortunately, the trade-off is a lower Recall, which indicated the proportion of actual positives that were identified correctly. This was to be expected due to the highly imbalanced nature of the dataset. The model tended to be conservative in predicting injuries, though very accurate when it did. However, the model in turn missed a larger number of true injuries.

Another major indicator to look at is the ROC AUC Score. The model performed exceptionally well with a 0.94 overall. The accuracy may be a bit misleading because of the imbalance in the dataset because of dominance in the majority class. The score remained high which indicates that it is capable of correctly discriminating between injury and non-injury across various thresholds. Its high AUC score also indicates that the model that it is more likely to correctly rank a positive instance than a randomly chosen negative one.

It is clear the Recall value was a focal point for the project. The model had a very good Recall rate compared to all other models that we tried. With a 0.67 representing that an injury was correctly predicted 2 out of 3 times. Though specificity was extremely high, it does not show us much because it is only representing the plays where injuries did not occur.

Finally, we looked at feature importance in order to reveal which factors were the most critical in determining these scores. The ‘injury\_frequency’ category was engineered within the initial clustering of our data points. It was created by using the player position and play type. These two were mapped together in order to turn the categorical columns into one numerical factor. This seemed to be the deciding factor in enabling the model to work. Creating the variable ‘injury\_frequency’ was a powerful tool in enriching and enhancing the dataset. This was crucial in improving the data performance as well as giving us insights into the model.