**Download the dataset from Kaggle:** [**https://www.kaggle.com/datasets/icliu30/nba-player-stats-and-injured-data-from-13-to-23**](https://www.kaggle.com/datasets/icliu30/nba-player-stats-and-injured-data-from-13-to-23)

**File: MS\_Logistic.R**

1. Read the 'nba\_injured\_players.csv' into a data frame
2. Remove any non-numerical variables except for the INJURED\_TYPE variable
3. Convert the INJURED\_TYPE variable into a factor for the model to be able to predict it
4. Create a train and test set, with an 80/20 split into the train and test respectively. This was done with the createDataPartition function in the carat package.
   1. SMOTE was performed using the themis package to balance out the data set
5. In the nnet package, the multinom function was used to be able to perform logistic regression on the multiple injury types.
6. The predict function is then used with the test set, with the INJURED\_TYPE variable made sure to be converted back to a factor
7. The confusion matrix is then made to be able to see the results and accuracy of the model.

**File: MS\_PCA\_LinearRegression.R**

1. Read the 'nba\_injured\_players.csv' into a data frame
2. Remove any non-numerical variables except for the INJURED\_TYPE variable
3. Convert the INJURED\_TYPE variable into a factor, and then an integer for the model to be able to predict it
4. Using the scale function, the data is scaled before being put into the prcomp function to run the PCA.
5. A scree plot and biplot are made with the screeplot and biplot function in ggplot2 to see how many components are needed in the model.
6. These results are then made into a data frame with the as.data.frame function.
7. This data frame is then used to select the first six components and the injury type as a numeric into a subset, which is the data frame used by the model.
8. The lm function is then used to make the final linear model. A summary can be printed to show all results.
9. The predict function is then used comparing the actual results to the results from the model. Tables are made to show the results of the predictions, comparing the actual corrects to the false positives and incorrects.

**File: MS\_XGBoost.R**

1. Read the 'nba\_injured\_players.csv' into a data frame
2. Remove any non-numerical variables except for the INJURED\_TYPE variable
3. Convert the INJURED\_TYPE variable into a factor for the model to be able to predict it
4. Create a train and test set, with an 80/20 split into the train and test respectively. This was done with the createDataPartition function in the carat package.
5. Create the labels for the INJURED\_TYPE variable to be used in the model by converting the INJURED\_TYPE variable as a factor into a numerical.
6. Since the XGB model needs matrices, a train and test matrix needs to be made, this is done using the sparse.model.matrix in the Matrix package.
7. Some tests were done to ensure the validity of the data before the model was run
   1. Use the is.na function to check for any missing values
   2. Use the identical function to ensure the labels and matrix names are the same
   3. Use the is.infinite to check for infinite values
   4. Use the range function to check for any extreme values
8. The XGB model needs the matrices, so the xgb.DMatrix function in the xgboost package was used to make the dmatrices for the train and test matricies used in the model.
9. The model is then built using the dtrain matrix in the xgb.train function in the xgboost package with the number of rounds set to 100 to ensure the best results.
10. The predict function is then used with the test set, with the INJURED\_TYPE variable made sure to be converted back to a factor
11. The confusion matrix is then made to be able to see the results and accuracy of the model.

**File: MS\_XGBoost\_CrossVal.R**

1. Read the 'nba\_injured\_players.csv' into a data frame
2. Remove any non-numerical variables except for the INJURED\_TYPE variable
3. Convert the INJURED\_TYPE variable into a factor for the model to be able to predict it
4. Create a train and test set, with an 80/20 split into the train and test respectively. This was done with the createDataPartition function in the carat package.
   1. SMOTE was performed using the themis package to balance out the data set
5. Create the labels for the INJURED\_TYPE variable to be used in the model by converting the INJURED\_TYPE variable as a factor into a numerical.
6. Since the XGB model needs matrices, a train and test matrix needs to be made, this is done using the sparse.model.matrix in the Matrix package.
7. The XGB model needs the matrices, so the xgb.DMatrix function in the xgboost package was used to make the dmatrices for the train and test matricies used in the model.
8. The model is then built using the dtrain matrix in the xgb.train function in the xgboost package with the number of rounds set to 100 to ensure the best results. Cross-validation was also added to the function call by creating a trainControl with five-fold cross-validation as the method.
9. The predict function is then used with the test set, with the INJURED\_TYPE variable made sure to be converted back to a factor
10. The confusion matrix is then made to be able to see the results and accuracy of the model.

**File: Capstone\_project.ipynb**

1. Read the 'nba\_injured\_players.xlsx' into a data frame
2. Convert date columns to datetime format, handle missing values
3. Remove any non-numerical variables except for the INJURED\_TYPE variable.
4. Convert INJURED\_TYPE to a categorical variable for classification
5. Split the dataset into training (80%) and testing (20%) sets
6. Apply feature scaling for models requiring normalization.
7. Train and build model with and without SMOTE.
8. Generate predictions on the test set.
9. Evaluate accuracy, precision, recall, and F1-score.
10. Improve the model using Hyper parameter tuning with Grid search and Randomized search
11. Generate a confusion matrix for Best model.