Injury Detection and Prediction in Football using Machine Learning

Cluster Innovation Centre

University of Delhi

Dinesh Yadav

December
2023

Semester Long Project submitted for the paper “Flow of information in Living Systems”.

Certificate of Originality

The work embodied in this report entitled **“Injury Detection and Prediction in Football ”** has been carried out by **Dinesh Yadav** for the paper “**Flow of information in Living Systems”**. I declare that the work and language included in this project report are plagiarism-free.

Dinesh
(**Name and signature**)

Abstract

Injury Detection and Prediction in Football using Machine Learning

by

Dinesh Yadav
Cluster Innovation Centre, 2023

This project introduces a foundational approach to evaluating injury risk among elite-level youth and professional soccer players by leveraging machine learning algorithms. The dataset for this investigation was collected from the Pro Football Reference website in CSV format. The methodology employed in this study takes into account a player's match status, distinguishing between those actively playing and those sidelined due to a specific injury. A comprehensive analysis was conducted to explore the correlation between injuries and various factors such as a player's position and participation in practice. Machine learning models were then applied to the dataset to uncover patterns and relationships that contribute to injury risk. The predictive accuracy of the models was assessed by comparing their results with the remaining 20% of the dataset. The outcomes of this project offer valuable insights into the potential of machine learning models in identifying factors associated with player injuries. The application of machine learning in injury risk management strategies could potentially revolutionise the identification of high-risk youth players, paving the way for the development and enhancement of targeted injury prevention measures. This project showcases the machine learning's effectiveness in assessing injury risk and its potential to advance strategies for managing and preventing injuries, especially among youth Football players. The findings provide a foundation for further refining injury prevention measures in Football.

I. INTRODUCTION

In professional Football, injuries significantly impact team performance and incur substantial rehabilitation costs. The effective management of athletic performance is crucial in this context, aiming to maximise player performance while minimising injury risks. Key factors influencing these objectives include player health, emotional conditions (such as stress and anxiety), athletic load, physical demands (e.g., jumping and landing tasks), playing position, and overall performance load. Predicting injuries is paramount for prevention, forming the foundation for targeted preventive measures. Early detection through personalised injury prediction enables the avoidance of injuries by tailoring physical and workload requirements. This project focuses on testing a framework using athletic data from 285 soccer players sourced from the Pro Football Reference website. Integrating these comprehensive datasets into the proposed system enhances the accuracy of injury predictions. The Model developed identifies players at risk, facilitating early interventions to mitigate potential injuries.

in the project, different Machine learning models are used to obtain the maximum accuracy results regarding the players injury so that better predictions can be made.some model names are logistic regression, gradient boosting algorithm, decision tree classifier algorithm etc. in addition to the machine learning, some other analyses are also made through theoretical part which underlying the importance of considering the multifactorial causes behind the sports injuries that indicates that higher number of variables can provides comprehensive detailing of injury causes and some existing research papers suggest the formation of more complex models for identifying injury predictors.

I.1 Background and Context

In the dynamic world of professional soccer, player injuries pose a substantial challenge, influencing team performance and incurring significant rehabilitation costs. injuries can really throw a wrench in things. They not only affect how well the team plays but also cost a lot to get players back on their feet. So, the big question is: how can we predict and prevent these injuries better? Effective injury prevention and management have become paramount in the realm of sports, prompting the integration of advanced technologies, specifically machine learning, to enhance predictive capabilities.In recent years, the sports industry has witnessed a growing interest in leveraging machine learning models to analyze vast datasets and extract meaningful insights.

Importance of Machine Learning:

Machine learning models have emerged as powerful tools in sports analytics, enabling the identification of patterns and trends that may not be immediately apparent through traditional analysis. The application of these models in injury prediction seeks to push the boundaries of our understanding by considering a multitude of variables simultaneously. The aim is to move beyond simple cause-and-effect relationships and embrace the complexity inherent in sports injuries.

Over the past few decades, the use of Machine Learning (ML) models has gained popularity as a tool for sports injury prediction. ML models are algorithms that can learn from data and make predictions based on previous patterns. Logistic regression, decision tree, gradient boosting algorithm and random forest are well-known ML models that have been used for different machine learning based problems. Logistic Regression Predicts binary outcomes by modelling the relationship between input variables and the probability of a specific class. Decision Tree Utilises a tree-like structure to make decisions based on feature splits, creating a flowchart-like model for mapping features to outcomes. Gradient Boosting Algorithm Boosts weak models sequentially to correct errors, creating a strong predictive model by combining multiple decision trees. Random Forest Classifier Constructs an ensemble of decision trees during training and outputs the most common class for classification tasks, providing robust and accurate predictions.

I.2 Scope and Objectives

The primary objectives of this project are two-fold. Firstly, to employ various machine learning models on the dataset gathered from the Pro Football Reference website to enhance the accuracy of injury predictions. Secondly, to complement the machine learning analyses with a theoretical exploration that underscores the importance of considering diverse factors in understanding and preventing sports injuries.

Objectives:

- To evaluate the performance of ML models such as LR, Gradient boosting, Random Forest and KNN in predicting football injuries.
- To compare the results of different ML models and determine the most effective model for injury prediction.
- to analyse the different variables related to the football injuries which can be considered as the important predictor for injury prediction in football.
- To contribute to the existing body of knowledge on sports injury detection and prediction by providing new insights into the behaviour of the sports injuries.

I.4 Literature Survey

1. The first research article considered for literature survey is ROMMERS et al. “A machine learning approach to assess injury risk in elite youth football players”.[1]

   Purpose was To assess injury risk in elite-level youth football (soccer) players based on anthropometric, motor coordination and physical performance measures with a machine learning model.

method : evaluated 734 male youth players under 15 yr age categories, the evaluation process was measured on the basis of anthropometric, motor coordination and physical fitness.used XGBoost gradient boosting application to classify acute and overuse injuries.During the boosting process a set of weak learners is combined to improve prediction accuracy.

The approach employed boosted regression tree models, which focus on maximising precision and accuracy in classification, allowing for the inclusion of a larger number of variables compared to traditional multivariate models. This might explain the contrast with previous studies that didn't find associations between preseason performance tests and injury risk using traditional statistics.

In terms of predicting injuries, the five most important variables were anthropometric measures. Higher predicted age at peak height velocity (PHV), longer legs, greater body height, and lower body fat percentage were associated with an increased injury risk. These findings align with previous research that identified higher body height and weight, as well as higher predicted age at PHV, as risk factors for injuries.

2. The second article for literature survey is Ahemed naglah et al., "Athlete-Customized Injury Prediction using Training Load Statistical Records and Machine Learning," [2].

previous reserach:

Measuring and monitoring load during exercise was addressed and received focus for a long time. The excessive load leads usually to exertion or strain which probably increases the likelihood of injury. Some studies described the relation between physical stress and exertion and modeled it as power proportionality [3]. This relation can vary for each individual with the variations in physical and physiological capacities [4].

Method:

In this paper, the framework was tested on athletic data for 21 soccer players that was collected and/or measured from different sources including internal load data (such as heart rate), external load data (such as the duration of workout and number of jumps), as well as questionnaire data. All these data are integrated into the proposed system to increase injury prediction accuracy.to collect data , they used wearable devices,and gathered the measurements and statistics using advanced GPS technology.

To predict injuries, they used KNN(k-nearest neighbour), k-means classifier, and support vector machine classifier(SVM), they only analysed non-contact injuries.The analysis involved selecting important features, creating a balanced dataset of injured and non-injured players, and conducting various classifications. results showed that when looking at individual features, both KNN and k-means classifiers had low accuracy due to significant overlap between positive and negative samples. For the overall prediction, SVM with k-fold cross-validation was used. Normalising data didn't drastically improve accuracy, but it reduced variability across different folds.The final step involved finding correlations between individual features and the overall prediction. Features that aligned with the global prediction were identified, and a personalised prediction map was created for each player. These maps showed consistent attributes across different players, providing insights for managers and trainers to pay attention to specific factors for injury prevention.

3. The third paper considered for literature survey is Rossi et al., “ Effective injury forecasting in soccer with GPS training data and machine learning”[5]

method : GPS measurements and statistical data gathering of 26 Italian soccer players, describing the training workload of players in a professional football club during a season, then creating an injury forecaster.

They divided the dataset into 5 factors such as daily features, Exponential Weighted Moving Average Feature, Acute chronic workload ratio features,MSWR features ,and previous injury features and built a model based on the ADASYN algorithm.

The study demonstrates that a Decision Tree (DT) model for injury prediction in soccer players shows promising results. The DT can detect about 80% of injuries with a relatively low false positive rate, making it more reliable than existing methods. The model's effectiveness improves over the season, with a cumulative F1-score of 0.60, outperforming other approaches. The study suggests an initial period of data collection is crucial for accurate predictions. Notably, the features influencing injury prediction evolve over the season, emphasising the importance of continuous monitoring. The model highlights specific factors, such as a player's return from injury and kinematic workloads, contributing to injury risk.

In practical terms, the findings suggest that clubs should pay special attention to players returning from injuries, particularly in their initial training sessions. Monitoring kinematic workloads during these periods can significantly contribute to injury prevention. The economic impact of injuries on the club is substantial, and the DT model offers a practical and cost-effective solution to reduce both financial and performance-related consequences.

What is Logistic regression :

Logistic regression is a statistical method used for binary classification problems, where the outcome variable is categorical and has two possible classes (usually labeled as 0 and 1). It's an extension of linear regression and is particularly suited for problems where the dependent variable represents the probability of belonging to a particular class.

Here are key features and concepts associated with logistic regression:

1. Sigmoid Function (Logistic Function): The logistic regression model applies the sigmoid (or logistic) function to the linear combination of input features. The sigmoid function converts any real-valued number into the range of [0, 1]. The formula is:

   $P(Y=1) = 1/ 1+Exp(b0+b1X1+b2X2+....+bnXn)$

   Here, P(Y=1) is the probability of the dependent variable (Y) being 1, e is the base of the natural logarithm, $b0$ is the intercept, $b1$,$b2$,...,bn are the coefficients associated with the input features $X1$, $X2$,...,Xn.

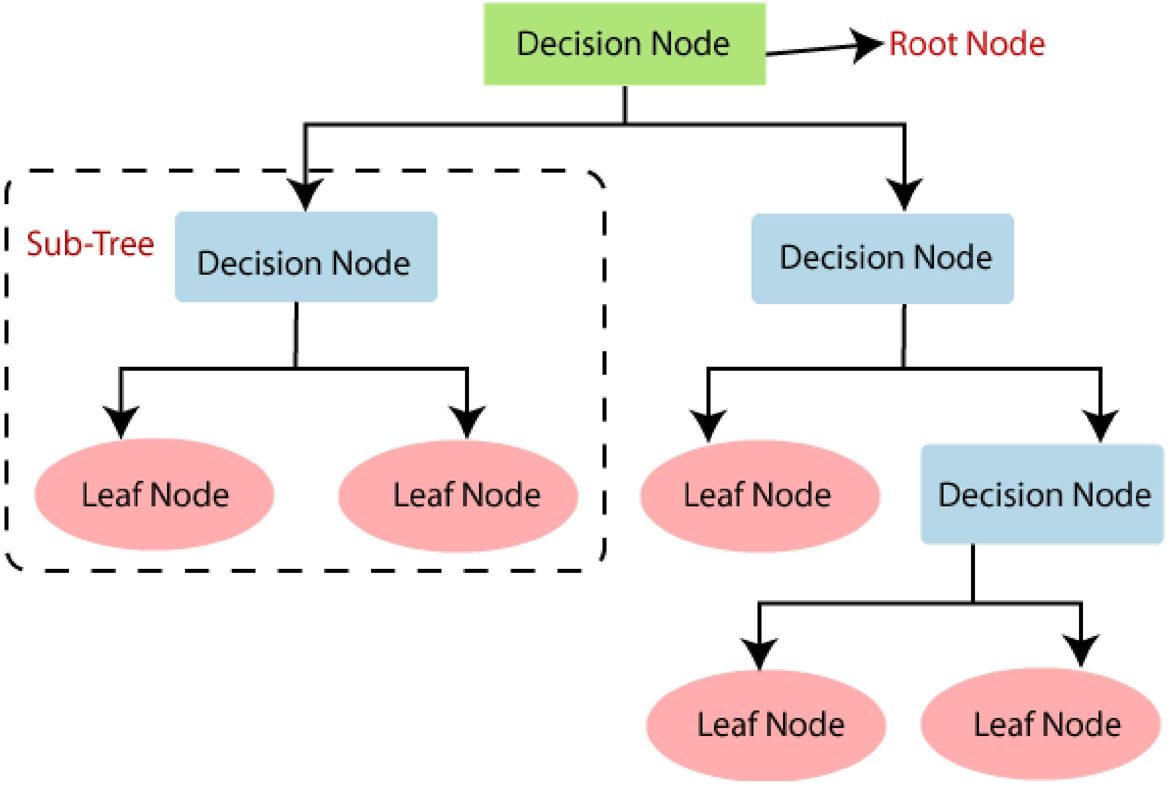
2. Decision Boundary: The output of the logistic regression model is a probability. By default, if the predicted probability is greater than 0.5, the model predicts class 1; otherwise, it predicts class 0. The decision boundary is the threshold at which the model transitions from predicting one class to another.

Random Forest

Random Forest is an ensemble learning technique that builds multiple decision trees during training and outputs the mean prediction for regression tasks or the mode prediction for classification tasks. Each tree is trained on a random subset of the data, and the final prediction is a combination of predictions from all trees. This approach enhances robustness and reduces overfitting, making Random Forest a versatile and powerful algorithm for various machine learning tasks.

Decision tree Classifier

A Decision Tree is a tree-like model used for both classification and regression tasks in machine learning. It recursively splits the dataset into subsets based on the most significant features, creating a tree structure where each leaf node represents a class label (in classification) or a numerical value (in regression). The decision-making process involves answering sequential questions about the input features, leading to a final prediction at the leaf nodes. Decision Trees are



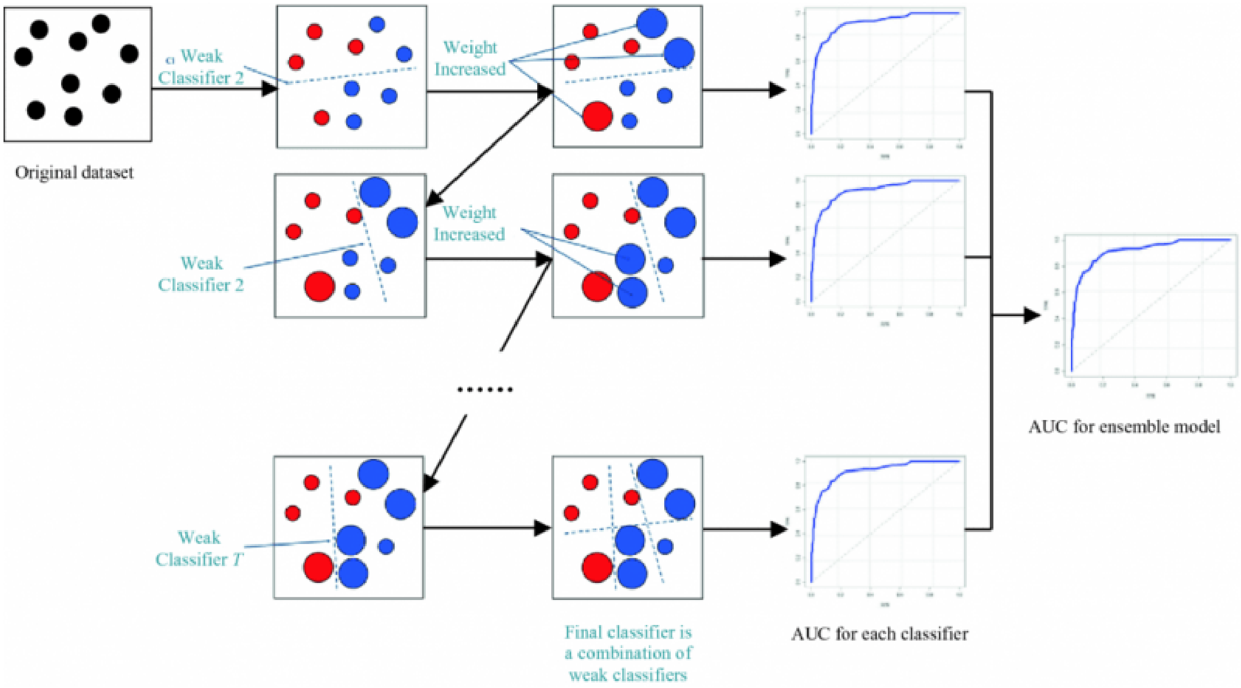
interpretable, easy to understand, and can capture complex relationships in data. However, they are prone to overfitting, which can be mitigated by techniques like pruning.

KNN:

KNN, or k-Nearest Neighbors, is a simple and effective machine learning algorithm used for both classification and regression tasks. In KNN, the prediction for a new data point is determined by the majority class (for classification) or the average of nearby points (for regression) among its k-nearest neighbors in the training dataset.

Gradient Boosting Model

Gradient Boosting is an ensemble learning technique that combines the predictions of multiple weak learners to create a strong predictive model. It belongs to the class of boosting algorithms, and one of the most popular implementations of gradient boosting is the Gradient Boosting Machine (GBM). XGBoost and LightGBM are examples of gradient boosting libraries that have gained popularity due to their efficiency and performance.



Key Concepts of gradient boosting

Weak Learners (Decision Trees): In the context of gradient boosting, a weak learner is typically a decision tree with a shallow depth (sometimes called a "stump"). These trees are often referred to as "weak" because they have limited predictive power on their own.

Gradient Boosting is an ensemble method that combines weak learners, often shallow decision trees, to improve predictive accuracy. It uses gradient descent to minimize a loss function measuring the difference between predicted and actual values. XGBoost and LightGBM are efficient gradient boosting libraries, with XGBoost offering features like regularization and handling missing data, while LightGBM is designed for distributed and efficient training, ideal for large datasets.

I.4 Achievements

• **A decent accuracy rate of Predictions:** Ability to make decent accurate injury predictions using different machine learning models.

• **Identification of Key Factors:** Identification of key factors contributing to players injuries during matches, providing valuable information for risk management.

• **Comparison of ML Models:** Comparison of various machine learning models to determine the most effective model for injury prediction.

• **Increased Understanding** of football injuries Dynamics: Increased understanding of the dynamics of the football players injuries and the factors affecting the performance of players.

• **Improved Risk Management:** Improved risk management through the ability to forecast players injuries type and minimise the impact of performance and health downturns.
• **Contribute in Ongoing Research:** Contribution to ongoing research in the field of injuries in the sports industry analysis and machine learning by adding new knowledge and insights.

II. FORMULATION OF THE PROBLEM

II.1 Problem Statement

Using Various Machine Learning Model for Predicting Injury Risk in Youth Football Players.
Sports players face a hidden challenge behind all their successes – injuries. Every player dreams of a season without getting hurt, but injuries are just part of the game. Luckily, technology, like machine learning, is making a big impact in many areas, and sports is no exception. By using machine learning, sports teams and scientists can predict when a player might get injured and take steps to prevent it. This could change the game, helping players stay healthier and perform better. It's like giving athletes a better shot at staying at the top of their game!

II.2 Methodology

1. Data collection and processing:

- Collection of past data on the selected sports injuries

- Calculating the indicators:Calculating said indicators for the whole dataset.
- Cleaning and pre-processing the data:Cleaning and preprocessing to remove missing or irrelevant values, and ensure that the data is in a suitable format for analysis.
- Storing or uploading the data: Uploading the dataset to [3] for efficient access and manipulation.

2. Model Selection:

- Research about the Models: Review and select appropriate machine learning models for sports injury prediction.

- Evaluating the models: Evaluate the strengths and weaknesses of each model, and consider the suitability of each model for the data and research question.
- Final selection: Decide on the final models to be used for the analysis. The final models implemented are LR, Random Forest, KNN, Gradient boosting.

3. Model Training:

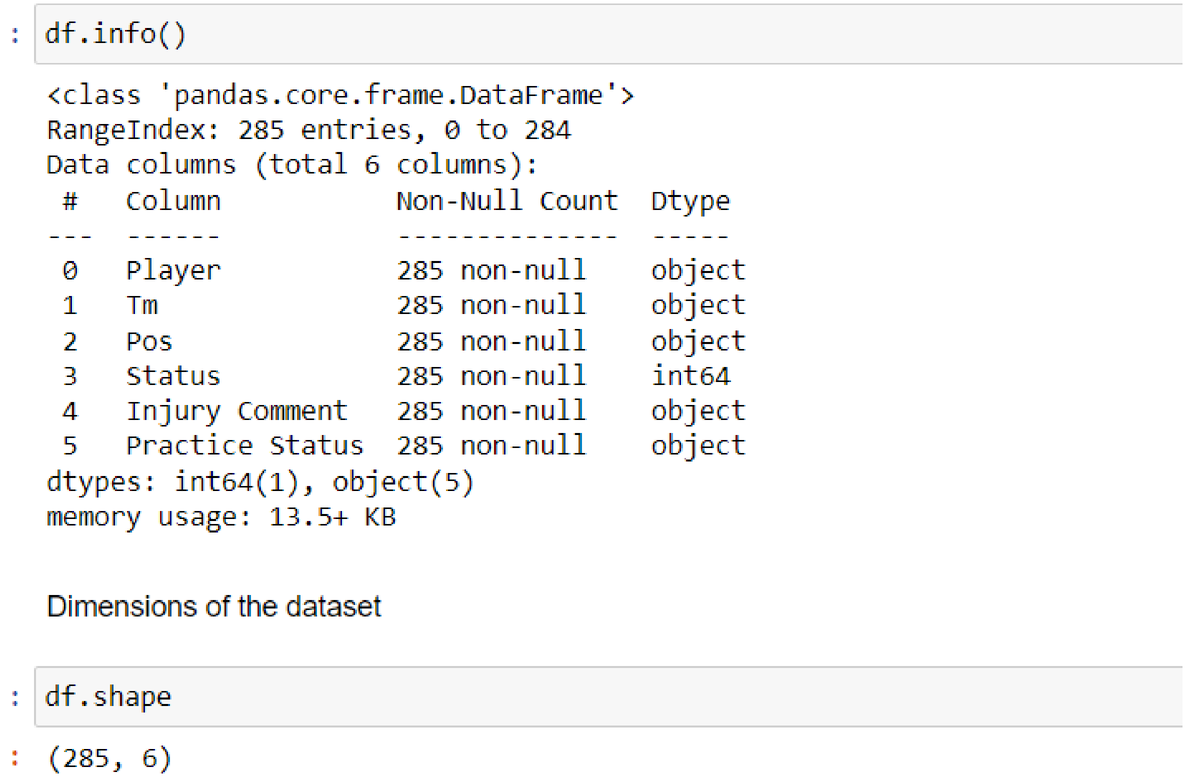
- Splitting the data: Split the data into training and testing sets, and use the training set to train the machine learning model.
- Improving the hyperparameters: Evaluate the model performance on the training set, and make any necessary adjustments to improve the model's accuracy.

4. Model Testing:

- Testing the models: Use the testing set to evaluate the performance of the trained model on unseen data.
- Evaluating the Model: Calculate metrics such as accuracy, root mean squared error, and mean absolute percentage error to measure the performance of the model.

III.3 Results

Data Processed:



```
: df.info()

  <class 'pandas.core.frame.DataFrame'>
  RangeIndex: 285 entries, 0 to 284
  Data columns (total 6 columns):
   #   Column           Non-Null Count  Dtype 
  ---  ------           --------------  ----- 
   0   Player           285 non-null    object
   1   Tm               285 non-null    object
   2   Pos              285 non-null    object
   3   Status           285 non-null    int64 
   4   Injury Comment   285 non-null    object
   5   Practice Status  285 non-null    object
  dtypes: int64(1), object(5)
  memory usage: 13.5+ KB
```

Dimensions of the dataset

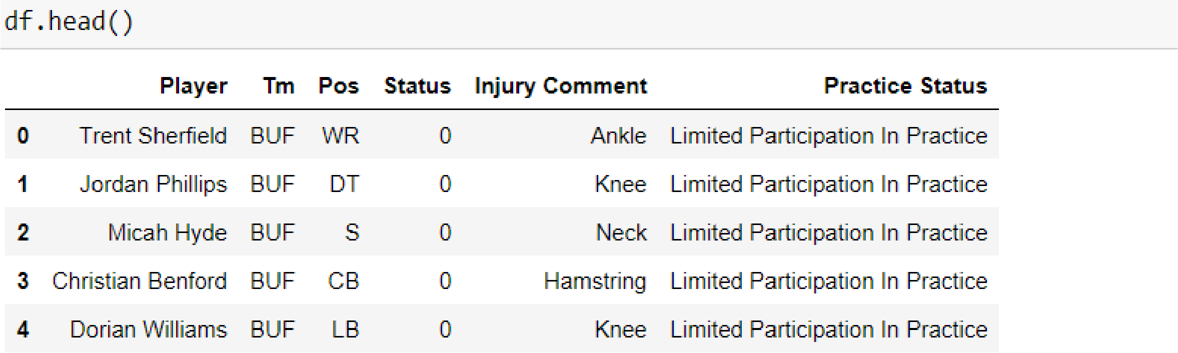
```
: df.shape

: (285, 6)
```

Data visualisation:

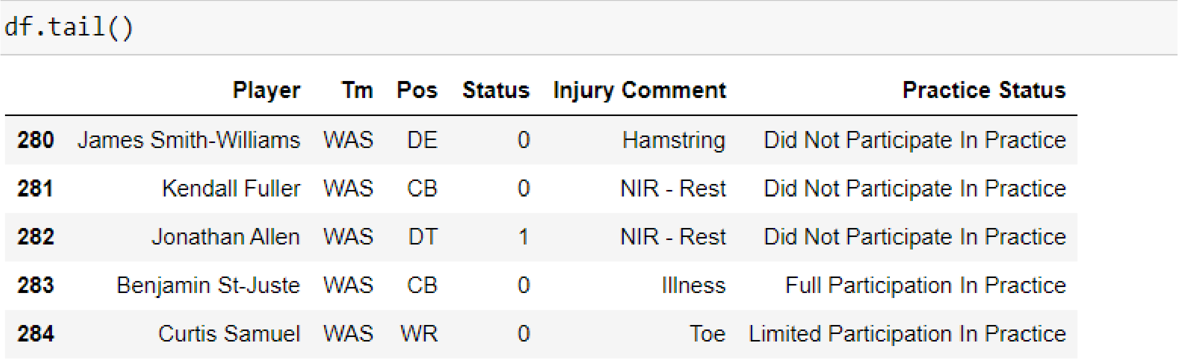
in this process, we use python libraries such as pandas, seaborn and matplotlib etc to visualise our csv data, so that we get an idea that how our data look like.

and after forming the countplots and corelation plots, we can make the observations regarding the data on which we apply the machine learning models aftermath.



```
df.head()
```

	Player	Tm	Pos	Status	Injury Comment	Practice Status
0	Trent Sherfield	BUF	WR	0	Ankle	Limited Participation In Practice
1	Jordan Phillips	BUF	DT	0	Knee	Limited Participation In Practice
2	Micah Hyde	BUF	S	0	Neck	Limited Participation In Practice
3	Christian Benford	BUF	CB	0	Hamstring	Limited Participation In Practice
4	Dorian Williams	BUF	LB	0	Knee	Limited Participation In Practice



```
df.tail()
```

	Player	Tm	Pos	Status	Injury Comment	Practice Status
280	James Smith-Williams	WAS	DE	0	Hamstring	Did Not Participate In Practice
281	Kendall Fuller	WAS	CB	0	NIR - Rest	Did Not Participate In Practice
282	Jonathan Allen	WAS	DT	1	NIR - Rest	Did Not Participate In Practice
283	Benjamin St-Juste	WAS	CB	0	Illness	Full Participation In Practice
284	Curtis Samuel	WAS	WR	0	Toe	Limited Participation In Practice

Information about player’s position:(“Pos” in dataset)

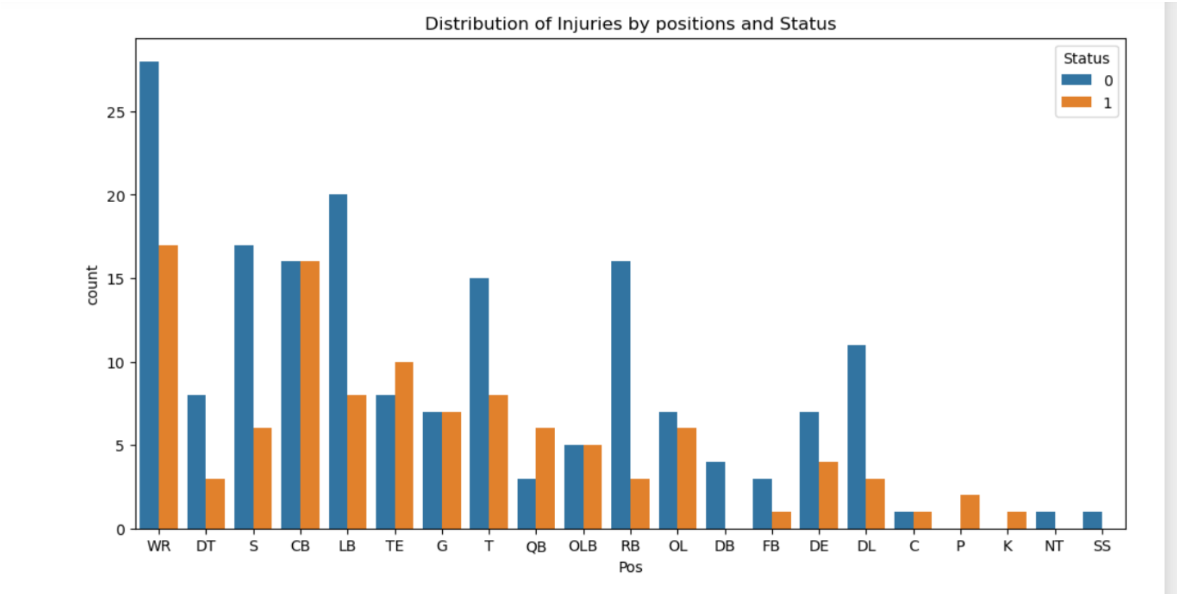
1. WR - Wide Receiver
2. DT - Defensive Tackle
3. S - Safety
4. CB - Cornerback
5. LB - Linebacker
6. TE - Tight End
7. G - Guard
8. T - Tackle
9. QB - Quarterback
10. OLB - Outside Linebacker
11. RB - Running Back
12. OL - Offensive Line
13. DB - Defensive Back
14. FB - Fullback
15. DE - Defensive End

16. DL - Defensive Lineman

17. C - Center
18. P - Punter
19. K - Kicker
20. NT - Nose Tackle
21. SS - Strong Safety

These abbreviations are commonly used in football to denote specific player positions on the field.

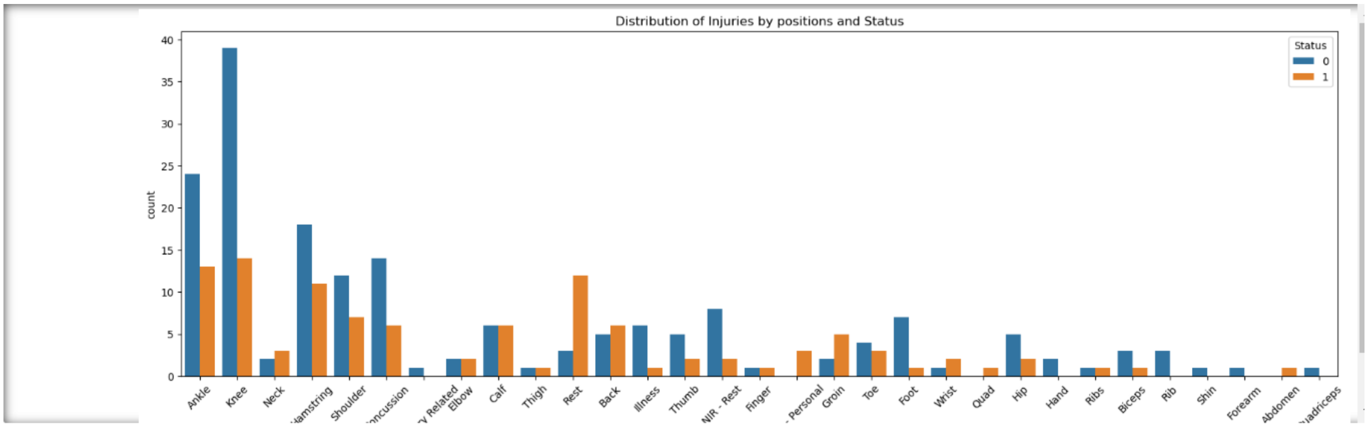
Chart 1.



on the basis of above countplot which is a kind of plot , following observation are made -

Following an injury, wide receiver (WR) position players face the highest likelihood of being sidelined during a match, with linebackers (LB), running backs (RB), safeties (S), tackles (T), defensive linemen (DL), and others in various positions following suit. Notably, cornerbacks (CB), guards (G), and outside linebackers (OLB) share an equal probability of either continuing to play or exiting the match after sustaining injuries. On the other hand, tight ends (TE), quarterbacks (QB), and punters (P) exhibit higher chances of remaining in the game even in the aftermath of an injury.

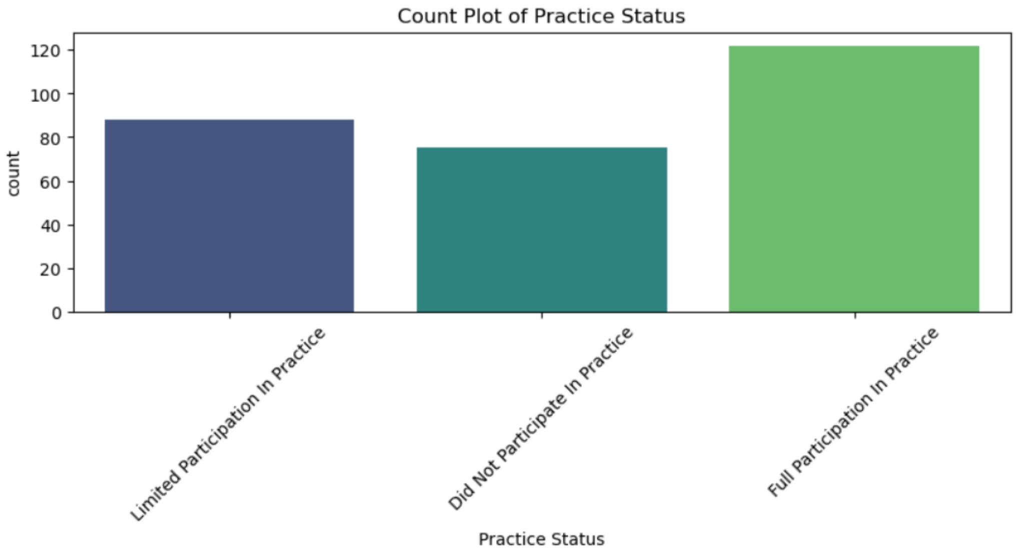
Chart 2.

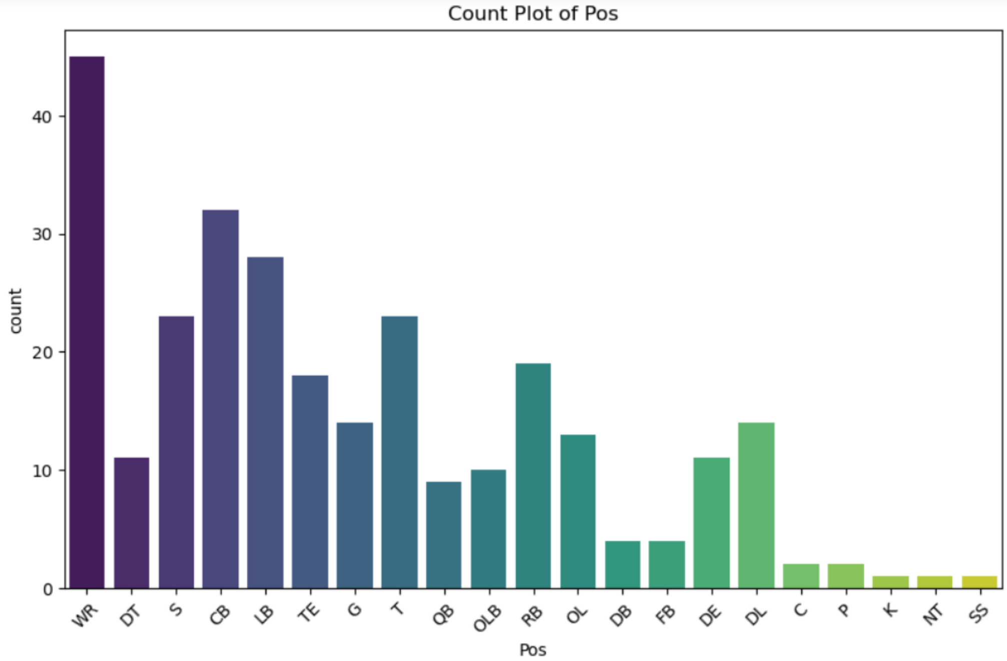


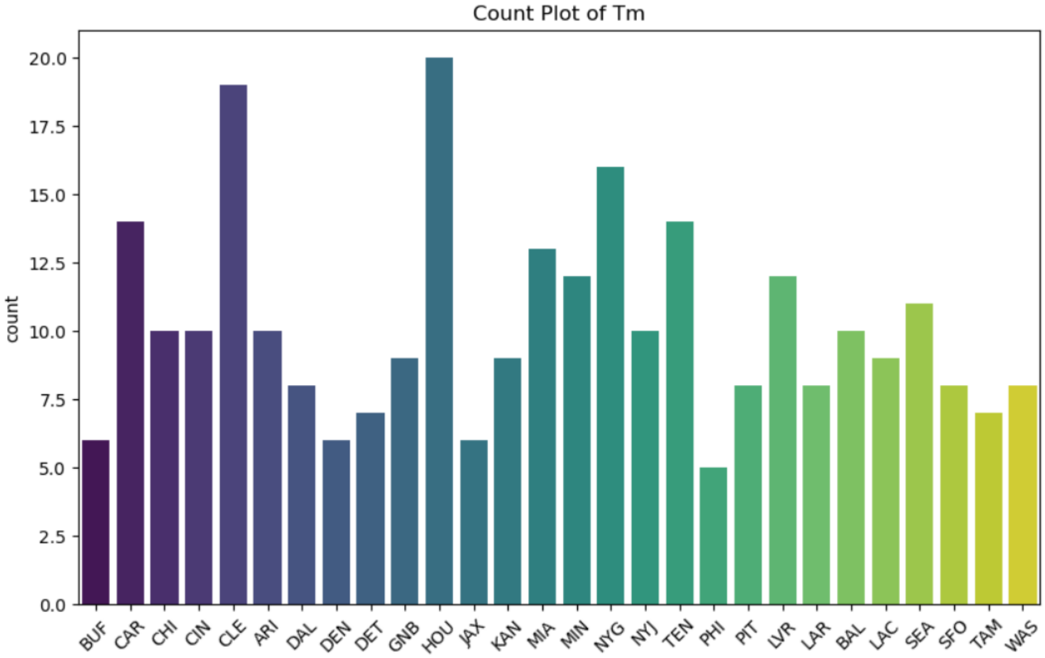
on the basis of above countplot which is a kind of plot , following observation are made -

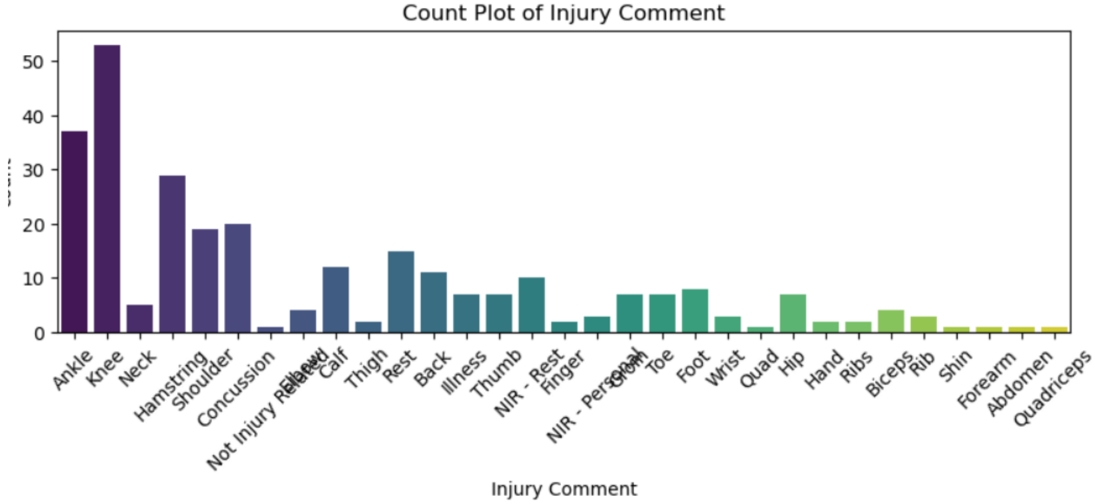
Players with knee injuries face the highest likelihood of being sidelined during a game, with ankle, hamstring, shoulder, and concussion injuries following closely in terms of their impact on player participation. Interestingly, players with calf injuries exhibit an equal chance of either remaining in the game or being sidelined.

Creating the count plot of each column:


Count Plot of Practice Status
count
0
20
40
60
80
100
120
Limited Participation In Practice
Did Not Participate In Practice
Full Participation In Practice
Practice Status


Count Plot of Pos
count
0
10
20
30
40
WR
DT
S
CB
LB
TE
G
T
QB
OLB
RB
OL
DB
FB
DE
DL
C
P
K
NT
SS
Pos



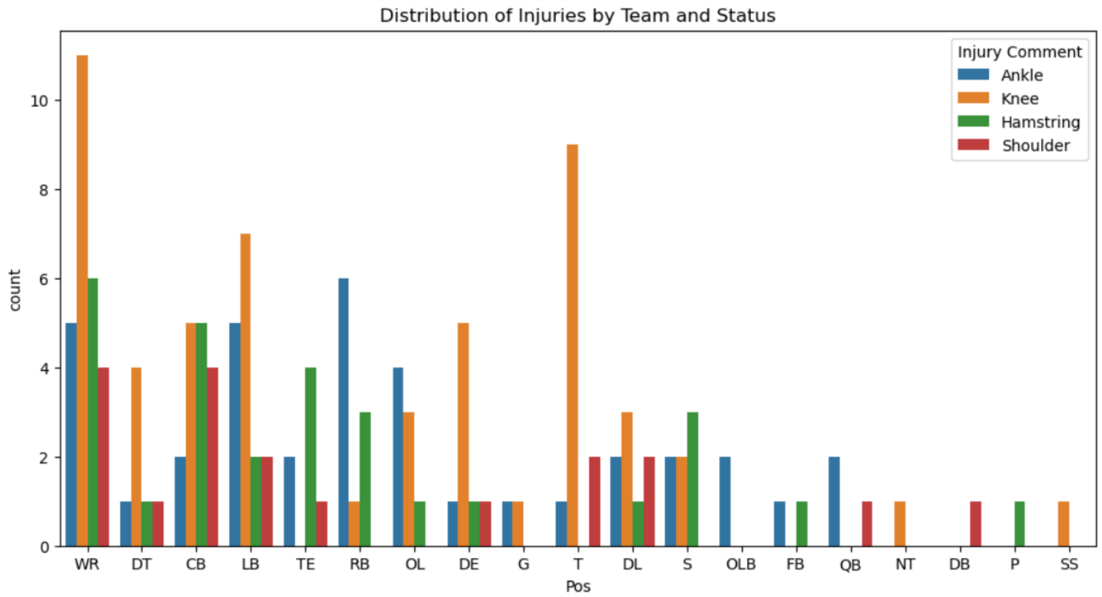


Following observations are based on countplots

The WR postion player has the highest possibility of getting injured. Whereas the other higher probability of injury chances positions are CB, LB, T, S, RB and TE.

The WR postion player has the highest possibility of getting injured. Whereas the other higher probability of injury chances positions are CB, LB, T, S, RB and TE.

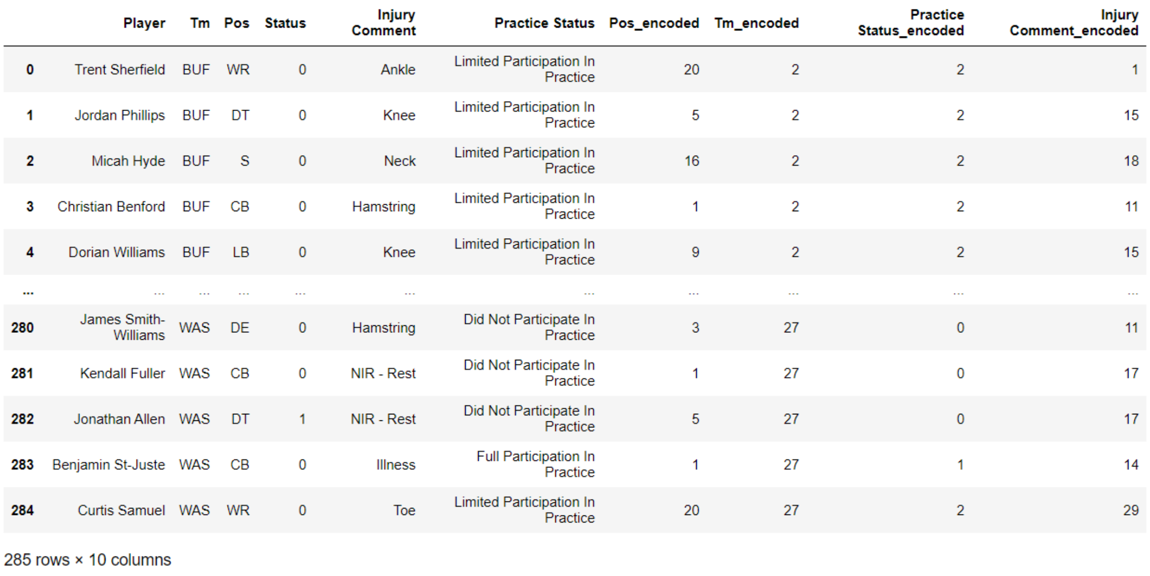
teams HOU, CLE, NYG, TEN, CAE,and MIA are having higher number of injured players.



here i have taken 4 injury types which are more common and then observed with the players position and the observations are -

WR, T, DE and SS position players have knee injuries in most of the cases.so they are more prone to sustaining knee injury in comparision to the other injuries.

hamstring and knee injuries are most common in all players position.



	Player	Tm	Pos	Status	Injury Comment	Practice Status	Pos_encoded	Tm_encoded	Practice Status_encoded	Injury Comment_encoded
0	Trent Sherfield	BUF	WR	0	Ankle	Limited Participation In Practice	20	2	2	1
1	Jordan Phillips	BUF	DT	0	Knee	Limited Participation In Practice	5	2	2	15
2	Micah Hyde	BUF	S	0	Neck	Limited Participation In Practice	16	2	2	18
3	Christian Benford	BUF	CB	0	Hamstring	Limited Participation In Practice	1	2	2	11
4	Dorian Williams	BUF	LB	0	Knee	Limited Participation In Practice	9	2	2	15
...	...	...	...	...	...	...	...	...	...	...
280	James Smith-Williams	WAS	DE	0	Hamstring	Did Not Participate In Practice	3	27	0	11
281	Kendall Fuller	WAS	CB	0	NIR - Rest	Did Not Participate In Practice	1	27	0	17
282	Jonathan Allen	WAS	DT	1	NIR - Rest	Did Not Participate In Practice	5	27	0	17
283	Benjamin St-Juste	WAS	CB	0	Illness	Full Participation In Practice	1	27	1	14
284	Curtis Samuel	WAS	WR	0	Toe	Limited Participation In Practice	20	27	2	29

285 rows × 10 columns

this is the the dataset after encoding the feature variable columns.

Models Accuracy:

Decision tree classifier = 68%

Random Forest classifier = 68%

Gradient Boosting classifier = 68%

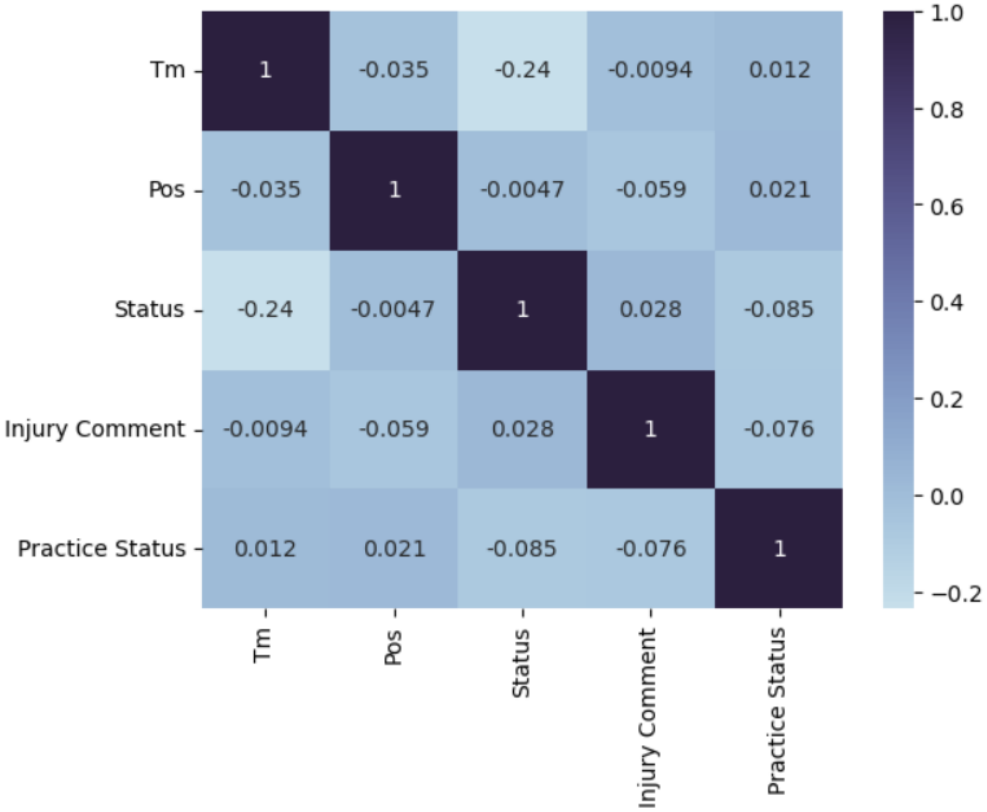
logistic regression = 67%

KNN = 84%

XGBclassifier = 65%

KNN is the best performing model in comparison to the other ML models.

Heat-Map for correlation :-



According to the heatmap, it can be observed that the practice status is positively related to the team name, meaning each team has their own practice schedule, which impacts the players performance accordingly.

and the player's position is also positively correlated to practice status , meaning each position requires a certain amount of practice.and which affects the overall team performance.

Injury comment and status are positively correlated as they confer a strong relation between the two, as injury like knee, shoulder, hamstring, ankle etc are more likely to keep the pliers outside the matches.

What i Have learned:

- Increase in understanding of the subject: The project helped me to get a better understanding of the subject and also gave me the opportunity to have hands-on practical experience.

CONCLUSION

In conclusion, the injury prediction and detection project aimed to predict the injuries and forecast the outcome regarding a player's performance using various machine learning models. It was found that KNN model performed the best in terms of accuracy and robustness.

- However, it is important to note that sports injuries predictions are uncertain and can be influenced by various factors such as a player's mental conditions, climatic change(demographic specific), previous injuries, Athletic load and physical demand. Therefore, the results obtained from this project should be taken with caution and used as a reference point for further research and analysis.

-

Discussing the limitations of project, it's widely acknowledged in machine learning that larger datasets lead to improved results and accuracy. Unfortunately, in this project, the dataset was relatively small, resulting in certain models not performing as expected. But, after gathering data set from the football.reference website for 5-6 weeks, data got a bit bigger. When I used machine learning on this somewhat larger dataset, the accuracy of models went up. but after literature survey of various papers, it is found that injury prediction can be much better with the help of ML if we consider high number of variables and their inter-relationship because using the cut-off values and studying only linear interactions between isolated variables cannot successfully identify injury predictors, so in order to get most accurate prediction, more comprehensive data with inter-related variables and more complexed models should be applied. and to overcome limitations, predictive analysis should be utilised alongside explanatory methods.

References :

[1] Rommers, N., Rössler, R., Verhagen, E., Vandecasteele, F., Verstockt, S., Vaeyens, R., … Witvrouw, E. (2020). A machine learning approach to assess injury risk in elite youth football players. *MEDICINE AND SCIENCE IN SPORTS AND EXERCISE*, *52*(8), 1745–1751. https://doi.org/10.1249/mss.0000000000002305

[2]A. Naglah et al., "Athlete-Customized Injury Prediction using Training Load Statistical Records and Machine Learning," 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Louisville, KY, USA, 2018, pp. 459-464, doi: 10.1109/ISSPIT.2018.8642739.

[3]Borg, G. A. (1982). Psychophysical bases of perceived exertion. *Medicine and science in sports and exercise*, *14*(5), 377-381.

[4] Russell, W. D. (1997). On the current status of rated perceived exertion. *Perceptual and motor skills*, *84*(3), 799-808.

[5]Effective injury forecasting in soccer with GPS training data and machine learning,**PLOS ONE, 13(7), e0201264 - July 2018, https://doi.org/10.1371/journal.pone.0201264**

[6] https://www.pro-football-reference.com/players/injuries.htm
-reference website for the dataset used in the project.

[7]https://github.com/Prateek525/ML-project/blob/main/injury%20prediction.ipynb
(GitHub repository)

APPENDIX

code

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('football latest.csv')
    df.info(), df.sample(10)

    df.shape

    df.head()

    df.tail()

    plt.figure(figsize=(12, 6))

    sns.countplot(x='Pos', hue='Status', data=df)

    plt.title('Distribution of Injuries by positions and Status')

    plt.show()

    plt.figure(figsize=(20, 6))

    sns.countplot(x='Injury Comment', hue='Status', data=df)

    plt.title('Distribution of Injuries by positions and Status')

    plt.xticks(rotation=45)

    plt.show()


    categorical_columns = df.select_dtypes(include='object').columns

    for column in categorical_columns:

        plt.figure(figsize=(10, 3))

        sns.countplot(x=column, data=df, palette='viridis')

        plt.title(f'Count Plot of {column}')

        plt.xticks(rotation=45)
    plt.show()

selected_injury_comments = ['Ankle', 'Knee', 'Hamstring', 'Shoulder']

if not all(isinstance(value, str) for value in selected_injury_comments):
    raise ValueError("All elements in selected_injury_comments must be
strings")

# Filter out non-string values and perform case-insensitive comparison
filtered_df = df[df['Injury
Comment'].astype(str).str.lower().isin(map(str.lower,
selected_injury_comments))]

plt.figure(figsize=(12, 6))
sns.countplot(x='Pos', hue='Injury Comment', data=filtered_df)
plt.title('Distribution of Injuries by Team and Status')
plt.show()

from sklearn.preprocessing import LabelEncoder

categorical_columns = ['Pos', 'Tm', 'Practice Status', 'Injury Comment']
label_encoder = LabelEncoder()
for column in categorical_columns:
    df[column + '_encoded'] = label_encoder.fit_transform(df[column])

df['Status'].fillna('1', inplace=True)
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

df = pd.read_csv('football latest.csv')

df = df.drop(['Player'], axis=1) # Dropped unrequired columns

label_encoder = LabelEncoder()
df['Tm'] = label_encoder.fit_transform(df['Tm'])
df['Pos'] = label_encoder.fit_transform(df['Pos'])
df['Status'] = label_encoder.fit_transform(df['Status'])
df['Practice Status'] = label_encoder.fit_transform(df['Practice Status'])
df['Injury Comment'] = label_encoder.fit_transform(df['Injury Comment'])

X = df.drop(['Status'], axis=1)  # Features
y = df['Status']  # Target variable

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Creating a Decision Tree classifier
model = DecisionTreeClassifier(random_state=42)

model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

print('\nClassification Report:')
print(classification_report(y_test, y_pred))

model.fit(X_train, y_train)
from sklearn.ensemble import RandomForestClassifier
y_pred = (model.predict(X_test)).astype(int)

accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print('\nClassification Report:')
print(classification_report(y_test, y_pred))

from sklearn.ensemble import GradientBoostingClassifier


model = GradientBoostingClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print('\nClassification Report:')
print(classification_report(y_test, y_pred))

from sklearn.linear_model import LogisticRegression



model = LogisticRegression(random_state=42)



model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print('\nClassification Report:')
print(classification_report(y_test, y_pred))

from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X, y)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

print('\nClassification Report:')
print(classification_report(y_test, y_pred))
print(df)
dataplot = sns.heatmap(df.corr(), cmap=sns.color_palette("ch:s=.25,rot=-.25", as_cmap=True),vmin = -1, vmax = 1, annot=True)

from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import pandas as pd
X = df.drop(['Status'], axis=1)
y = df['Status']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = XGBClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print('\nClassification Report:')
print(classification_report(y_test, y_pred))
```