Msc Data Analytics in Football

Sports Data Campus
UCAM (Catholic University of Murcia)

Player Injury Risk Classification Model

Module 12: Masters' Final Project
Tutor: Pablo Sanzol

Author: Harry Woodnutt

Table of Contents

1 Executive Summary

For this project I have created a prediction model that will allow key decision makers in medical departments at football clubs to understand the potential injury risk a given player carries for the following season. The model is trained on historical injury records for each individual player and biological data such as age and height. It has the capacity to include other data sources also, in particular we should look to Load Data in future to understand the impact this has on injury likelihood.

Medical departments would find value and use for this prediction model as it has a range of applications within Football Clubs, including; understanding the existing squad and their propensity for injury in the upcoming season, but also in assessing potential new recruits in terms of transfer targets and the injury risk they carry. By categorising each player as a ‘Low’, ‘Medium’, ‘High’ injury risk, where Low = < 15 days missed in the following season, Medium = 15-60 days missed in the following season, and High = 60+ days missed in the following season, we provide an intuitive an easy to use output for key decision makers.

This clear, data-driven assessment of injury likelihood supports informed decisions to optimise player availability, reduce injury-related disruptions, and guide strategic planning within professional football clubs.

.

2 Introduction

The world of football player recruitment is arguably one of the most financially high stakes areas in sport, with inflated transfer fees and misguided, impulsive decision making leading to inefficient spending. This has led to a much needed transformation to player recruitment. The application of data led recruitment processes has allowed the successful clubs of today to navigate the transfer window with much more efficiency, finding effective players for reasonable transfer fees, rather than paying highly inflated fees for overvalued talent. The successes of clubs like Liverpool, Brighton and Brentford all as early adopters of data led recruitment has seen them flourish whilst clubs that are too slow to react have fallen behind.

Seemingly now, most clubs take this data focussed approach to transfers. This can be credited with the general improvement in transfer success that followed. In this project, I look to also account for the impact potential injuries can have on the success of transfers too - historically injury has been viewed as a case of luck or an ‘uncontrollable’, however as we understand more in this field we understand how recurring injuries and load can become predictors of injury, hence the increase in load management we see in the sport today.

What do we mean when we discuss a successful or failed transfer? Typically it can be boiled down to the % of games played in the following seasons post transfer. In the book titled ‘How to Win the Premier League’, Ian Graham pinpoints Injury as one of the main reasons transfers don’t work out. Graham's research indicates that **less than 50% of Premier League signings over €10 million start more than 50% of games in their first two seasons**. This statistic underlines the challenges clubs face in ensuring that high value signings remain fit and available for selection. Given the greater focus on data led recruitment in terms of finding players of the necessary quality to improve the team, I believe there should also be some attention paid to the injury risk that player carries also, and that aspect shouldn’t be left as uncontrollable.

By utilising statistical packages via Python (SKlearn), I have used a Supervised Machine Learning technique that acts to classify players based on a mix of categorical and numerical features, called a Random Forest Classifier. Later on in the report I will dive deeper into the reasons this model was best for my use case and my initial trail of thought before landing on this model.

Objectives/Process

1. Collect Data from Transfermarkt
    - The first step was to collate a dataset of all potential features that could be used to predict Injury.

2. Organise / Clean Dataset
    - Given we were scraping, effort had to be applied to reformat and clean the data in to a valid format, this including a Target Variable column for each player to assist in supervised learning

3. Train / Test Model
    - Given the clean dataset, we then split into a train and test dataset to allow the model to learn what features affect the estimation the most, and then to test what it had learnt against the test dataset

4. Evaluating Model Performance
    - Once tested, it was important to understand how successfully the model predicted the risk category each player fell into.

5. Examples and Application
    - The current transfer window (Summer 2025) has seen numerous high value transfers rumoured, in particular, that of Alexander Isak to Liverpool FC. Let's investigate how the model classifies that player and whether Liverpool FC should tread carefully before outlaying £100m+ worth of transfer fees.

3. Technological Architecture

In the data analysis eco-system there are a vast choice of softwares, packages, coding languages and more than I could use to assist in this project. My general approach was to use whatever software or tool that was best fitted to that element of the analysis in my opinion, to avoid being forced into a single tool for the sake of cleanliness. Therefore, you will see how in this project I move from R to Python coding language for specific purposes, and utilise the different packages that are available in each. It is worth noting this is not the only way I could have approached this task in terms of these tools or softwares or models, which we will discuss later.

For this project I used a 2-step approach in terms of the tools and coding language I utilised, each stage allowing me to best approach the task required. Initially, my project was based in R using R Studio for effective data collection processes and cleaning processes. Once cleaned, I used Python and Google Collab to utilise the SKlearn package in order to train, test and evaluate my model effectively. I will now dive a little deeper into those packages - for specific code please access the relevant R and Python files in my Github Project.

R Studio

When building a prediction ML model, one of the key determining factors is the availability of data that could be used to predict the target variable, ie, the injury category for each player. Therefore, my choice for R studio / R was mainly around the existing packages that could allow me to scrape key football data sites most readily. I had already identified Transfermarkt as a key source of information around injury records which would provide the main predictive features of my model, therefore it made sense to utilise the package 'worldfootballR' which allows for the simple scraping of football related data from Transfermarkt.com.

WorldFootballR

This library comes pre-built with functions allowing the user to access varying football related data from a variety of sites including; Transfermarkt, FBref and more. This meant I didn't have to build my own web scraping tool from scratch. However, as we will see later in the report, issues with specific functions in WorldFootballR meant I was restricted in terms of potential data I could accrue, specifically in the area of 'Load Data' which I had identified as a key predictor of injury risk going forward. It also had issues regarding the collection of Bio Data, forcing me to create my own scraping tool for such information.

Dplyr / Readr / Stringr
Once the data had been collected, it was important to choose relevant packages that could help to clean and organise the data in a format that was relevant for the Modelling packages I would use later to create my classification model.

Purrr / Rvest / Httr
As mentioned above, I did end up producing my own scraping code to access relevant Bio data (Age, Height etc). These packages allowed me to develop my own web scrape tool where the worldfootballR package was either not working or unfit for the specific purpose.

Python / Google Collab:
Python contains a package called SKLearn which I was particularly keen to use for the modelling stages of this project. I chose this library due to my familiarity and easy access to statistical methodologies, like those I intended on implementing in this project.

Sklearn
Sklearn allows the process of preprocessing, model access, and model evaluation metrics for ML purposes much more simple than needing to build such code myself. This allows non-expert level users who still want to apply these statistical techniques the access to do so. In particular, SKlearn offers access to Linear Regression models and Classification models which would be tested and used in this project.

Pandas / Numpy
Once we have the necessary dataset, feature engineering becomes an important part to add value to the model created, therefore these packages used for data manipulation and numerical computations allow that.

Seaborn / Matplotlib
Whilst a model in itself is very useful, without proper data visualisation and plotting it can be difficult to properly understand what the results are telling us, or communicate to stakeholders areas of interest. Here we use Seaborn / Matplotlib to make that process simple and clear.

4. Methodologies and Techniques Employed

4.1 **Business Understanding**

The objective and key driver of this project was to find a way of categorising individuals based on their historical injury records and biological data as either a Low, Medium or High risk for injury in terms of the upcoming season. This would allow management and medical teams to understand their squad in terms of potential injury risk, and help avoid recruiting players with a high chance of injury moving forward. The approach outlined ensures we address this common issue with an easily interpretable output for end users, that they can then translate into their decision making process when it comes to squad building and load management.

4.2 **Data Understanding**

As part of my approach, it was important to accumulate a large list of potential features that could impact our end target - in order to make the prediction as accurate as possible. Omitting key data points from the model could mean the model finds it challenging to accurately predict injury risk moving forward. As a result, I accumulated a large dataset of which it was important to appropriately explore and understand before continuing.

The process involved collecting relevant data and then performing exploratory data analysis to understand the data better, and any limitations it may have. It also meant the assessment of the fullness of the data and the quality of the data, to ensure whatever the model was being trained on was high quality data that it could produce a valid output from.

4.3 **Data Preparation**

Once the data had been collected properly, it required cleaning and re-processing. The data scraped from Transfermarkt often came in non-standard formats or combined numerous data points within one field, and therefore it was important to properly clean the data first. Also in order to properly create a prediction model, the data must be in a specific format with features and a target variable per row of data, this is to ensure the Supervised Machine Learning technique can properly be trained and tested.

- Data Scraping: Using a combination of existing libraries (worldfootballR) and my own web scraping code.
- Data Cleaning: It was important to ensure consistency across different features, valid data types and deal with missing and null values.

- Data Standardisation: This involved standardising numerical values so that columns with larger values didn't overly affect the prediction in comparison to smaller features.
- Joining Datasets: Given we loaded different data from Transfermarkt, it was important to then join them together to ensure all features were present per row of data.
- Feature Engineering: We often wanted to view features such as Muscular Injury Count across the last 1/2/3 seasons, therefore we had to infer from the injury field that they were muscular, and then conditionally sum based on the year they occurred.
- Train vs Test Data: Once the final dataset was complete, it was important to split between Train and Test data to ensure the model had a large enough dataset to train itself on and then to measure itself against.

4.4 **Modelling**

In terms of modelling, I initially opted for a Regression Model, however on reflection decided the output from a Classification model (ie, Random Forest Classifier) was more beneficial for my use case. These ML techniques allow the model to learn in a supervised manner what a successful estimation looks like in terms of the target variable, it can then apply this learning to a series of unseen data (test dataset) to measure its effectiveness.

Model Selection: **Random Forest Classification**

Reasoning: This model was used for its ability to handle numerical and categorical datatypes, and its format of providing a clear categorical (classification) output.

Note: Despite being considered initially, It was deemed too difficult to request a Linear Regression model to estimate the 'Expected Number of Days Missed in the Following Season', given the natural variation and wide array of features that would be required to predict to that level of accuracy. This was decided having evaluated the effectiveness of the model using R2.

4.5 **Evaluation**

In order to properly evaluate the quality of the model, I utilised precision, recall, f1-score and support from the SKlearn package. From this, I was able to examine the performance of this type of model and where it worked best.

5. Work Development

In this part of the report, we look at the technical development of the project. This involves getting into some more of the granular detail in terms of steps taken to create this Injury Risk Categorisation Model. The development approach involved some steps taken that should be mentioned, including; data collection, data preparation, exploratory, data analysis. statistical methods employed, model training, and model evaluation. We will also dive into each feature in more depth to understand the individual contribution that feature had on the prediction classification model.

5.1 **Data Collection**

When considering data collection it was important to first understand the potential features, and then decide the best source of that information. I split my potential features into 3 categories; Bio, Injury Record, and Load data. Bio and Injury record data was most readily available from Transfermarkt, via the R package worldfootballR and its functions. Specifically, 'tm_player_bio' and 'tm_injury_history_record'.

I used data from the past 4 seasons (ie, since 2021) from the top 5 European leagues to create my dataset. Initially, I used various worldfootballR functions to establish a list of all teams within the top 5 european leagues and then used that list to establish a new list of all the players currently playing in the top 5 european leagues. This allowed me to feed the new list of transfermarkt URL's directly into the worldfootballR provided 'tm_player_injury_history' function which returned a new row for every single injury related to the players in the top 5 leagues.

Unfortunately, I couldn't repeat the same process for the Biological data from Transfermarkt as the 'tm_player_bio' function appeared to be faulty. Therefore, I established my own web scraping tool I saved as 'get_player_bio_info_table'.

5.2 **Data Cleaning**

Once that data was collected, it had to be cleaned to contain only relevant information and ensure each data point was in the correct format. The main aim of this step was to ensure the final dataset contained a single row per player with all the necessary features as columns. A summary of steps taken for each data set is as follows

- Filter to injuries since 2021 (given we were only interested in injuries in the past 3 years)
- Splitting 'DateofBirth' into Date of Birth and Age (where Age was a predictive feature)

- Standardising Positions into Position_1 (Def, Mid, Att) and Position_2 (Centre Back, Right Back, Left Back) for proper comparison.
- Converting height into a purely numerical data point (rather than 1.88m)
- Filling nulls in specific data points, for example, using the average height of the position_1 of a player if they were missing that data point.
- Aggregating injury data per player

With the cleaned Bio and Injury history data from Transfermarkt, it was important to join these datasets together using 'transfermarkt_url' which was present in each dataset as the joining key. This meant we could finally view a single row per player with columns for all their predictive features.

Finally, In order to properly utilise the categorical data points, it was important to encode them first. This involves creating unique features for every category option, with each being flagged as 'TRUE' if the individual had that option, or 'FALSE' if not. For example, 'Foot' would be split into 'Left_Foot' and 'Right_Foot' with whatever option being relevant flagged as 'TRUE'.

This process, undertaken in R using the functions and libraries mentioned before, allowed me to clean and organise the data into a format that was ready for Sklearn to model. At this point I switched away from R Studio to Google Collab to complete the remaining steps.

5.3 **Exploratory Data Analysis**

With this new organised dataset, it was important to fully understand the content of the dataset before beginning the modelling stage in the exploratory data analysis. I used a variety of techniques to get an overview into the cleanliness and availability of each data point, to ensure we didn't include any poor quality data in the model. Such techniques include;

1. **.info()**

- Displays each column in the dataset, showing the null count and data type of each

```
RangeIndex: 833 entries, 0 to 832
Data columns (total 31 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   player_name                                 833 non-null    object
 1   injury_count_last_3_seasons                 833 non-null    int64
 2   days_missed_last_3_seasons                  833 non-null    int64
 3   injury_count_last_2_seasons                 833 non-null    int64
 4   days_missed_last_2_seasons                  833 non-null    int64
 5   injury_count_last_1_seasons                 833 non-null    int64
 6   days_missed_last_1_seasons                  833 non-null    int64
 7   muscular_injury_count_last_3_seasons        833 non-null    int64
 8   muscular_days_missed_last_3_seasons         833 non-null    int64
 9   muscular_injury_count_last_2_seasons        833 non-null    int64
 10  muscular_days_missed_last_2_seasons         833 non-null    int64
 11  muscular_injury_count_last_1_seasons        833 non-null    int64
 12  muscular_days_missed_last_1_seasons         833 non-null    int64
 13  skeletal_injury_count_last_3_seasons        833 non-null    int64
 14  skeletal_days_missed_last_3_seasons         833 non-null    int64
 15  skeletal_injury_count_last_2_seasons        833 non-null    int64
 16  skeletal_days_missed_last_2_seasons         833 non-null    int64
 17  skeletal_injury_count_last_1_seasons        833 non-null    int64
 18  skeletal_days_missed_last_1_seasons         833 non-null    int64
 19  tendon_ligament_injury_count_last_3_seasons 833 non-null    int64
 20  tendon_ligament_days_missed_last_3_seasons  833 non-null    int64
 21  tendon_ligament_injury_count_last_2_seasons 833 non-null    int64
 22  tendon_ligament_days_missed_last_2_seasons  833 non-null    int64
 23  tendon_ligament_injury_count_last_1_seasons 833 non-null    int64
 24  tendon_ligament_days_missed_last_1_seasons  833 non-null    int64
 25  days_missed_current_season                  833 non-null    int64
 26  Height                                      833 non-null    float64
 27  Position_1                                  833 non-null    object
 28  Position_2                                  833 non-null    object
 29  Foot                                        833 non-null    object
 30  Age                                         831 non-null    float64
dtypes: float64(2), int64(25), object(4)
```

2. .describe()

- Shows key figures per column to understand the spread and variation of the data points, helping to identify any outlying or erroneous data.

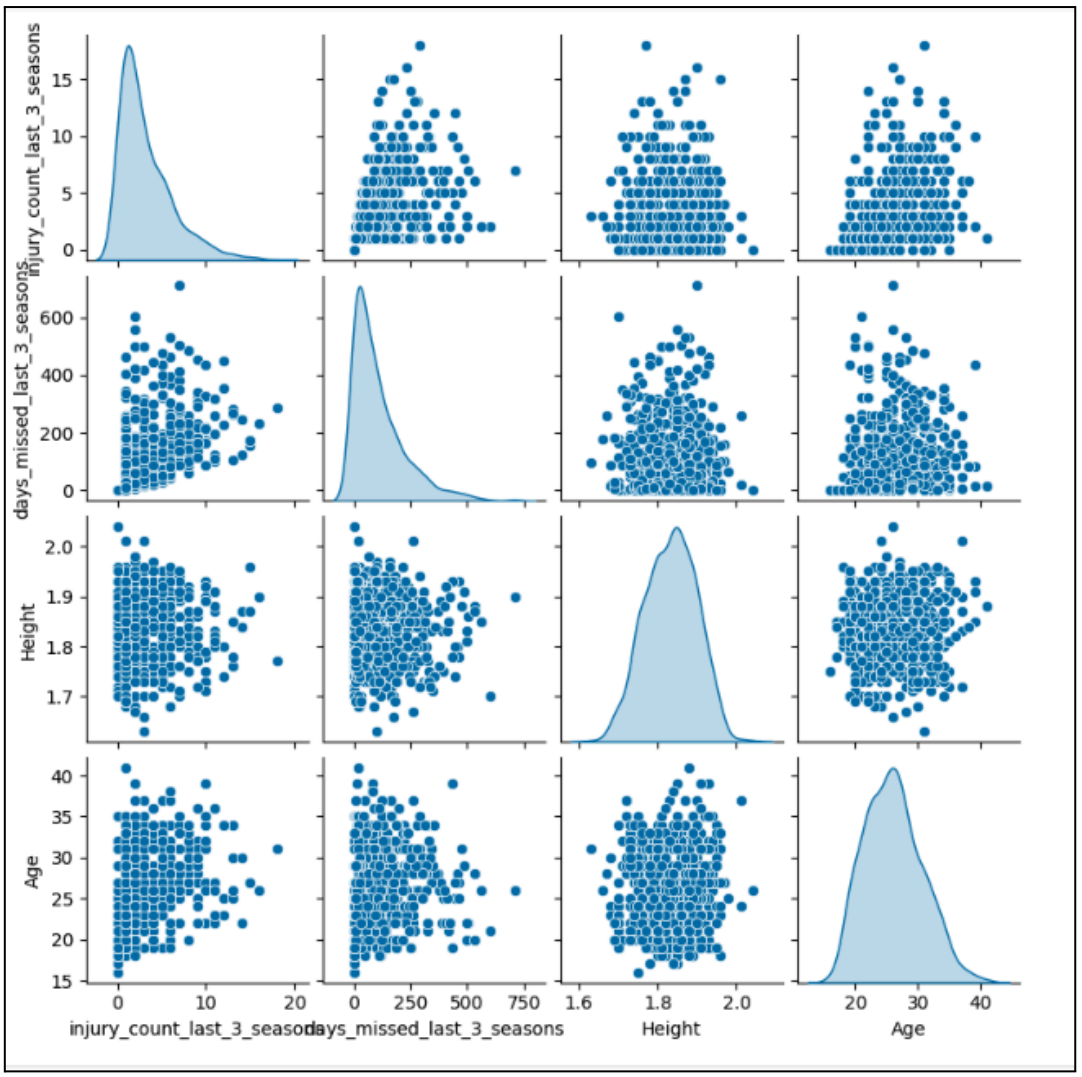
```
[53] pd.set_option('display.float_format', '{:.2f}'.format)
     injury_dataset.describe()
```

	injury_count_last_3_seasons	days_missed_last_3_seasons	injury_count_last_2_seasons	days_missed_last_2_seasons	injury_count_last_1_seasons
count	833.00	833.00	833.00	833.00	833.00
mean	3.23	107.26	2.20	79.64	1.19
std	3.04	116.90	2.17	99.50	1.29
min	0.00	0.00	0.00	0.00	0.00
25%	1.00	18.00	1.00	5.00	0.00
50%	2.00	69.00	1.00	45.00	1.00
75%	5.00	159.00	3.00	113.00	2.00
max	18.00	709.00	12.00	648.00	7.00

8 rows × 27 columns

3. **sns.pairplot()**

- Creates a quadrant of charts that show how each numerical variable is related to one another, I picked some key features for comparison.



5.4 **Feature Engineering**

Once fully understanding and accounting for any data issues, we could begin to design more complex features that our initial dataset doesn't contain.

For example, per injury we could see the exact injury the player suffered, e.g. Hamstring strain. In isolation this datapoint was so varied it wasn't providing much value, therefore I decided to group the injuries into 'Muscular', 'Skeletal', and 'Ligament/Tendon' to provide more detail into the injury type and how that may affect future injury risk. Once that grouping was complete, it allowed the creation of more interesting metrics such as 'Muscular Injury Count past 3 Seasons', rather than grouping all injuries into a single type. This led to some

interesting results around the type of injuries in the past that are more likely to contribute to injury risk in the future.

Next, we decided to create unique features for injury counts and days missed in each of the past 3 seasons, allowing us to display the relevance of recent injuries versus older injuries in our prediction. By grouping by season we could easily calculate features such as 'Days Missed Last Season' and 'Days Missed Last 3 Seasons' to shine a light on the impact this difference carries too.

Also, it was important to properly design our target variable to train our dataset on. Initially, I used the exact 'number of days missed' for a season as the target variable, however, upon reflection of our initial regression type model, it was decided creating bins based on the original target variable 'days_missed_current_season' was more suitable to fit our new Classification style model.

- High: Days Missed in Season >= 60
- Medium: 15 <= Days Missed in Season < 60
- Low: Days Missed in Season < 15

5.5 **Model Selection**

As mentioned before, my initial plan was to use a Linear Regression model to attempt to predict the exact number of days missed in the following season. Here I will touch on the evaluation scores of that model and the reasons behind the change to a Classification Model.

Regression Model Evaluation Scores:

```
from sklearn.metrics import mean_squared_error, r2_score

predictions = model.predict(X_test)


mse = mean_squared_error(y_test, predictions)
print("MSE:", mse)

rmse = np.sqrt(mse)
print("RMSE:", rmse)

r2 = r2_score(y_test, predictions)
print("R2:", r2)
```

```
MSE: 3898.587643335393
RMSE: 62.43867105676892
R2: 0.04917546170801779
```

Above are the evaluation scores of the initial model produced, we can see

- MSE = 3898
- RMSE = 62.43
- R2 = 0.049

When considering an R2 score, the higher the score along the range of 0-1 the greater the prediction capabilities of the model. For example, a 0.8 R2 score means the model in question accurately accounts for 80% of the variation in the target variable. Here, our score of 0.049 implies that this model is performing poorly, and only just above that of if we were to just use a model that always predicts the mean average of the target variable. Therefore, we can conclude the effectiveness of the model is very low and it has no real world application in its current state.

There are potential ways to improve this score, by adding Load Data for example or other data points that may improve the prediction power of the model. That being said, it is mostly just too difficult to request a model to accurately predict the exact number of days missed in the following season using the features provided. This is because ultimately Days Missed via Injury is not a linear relationship with these features. After understanding this point, it was important to consider other model types and how they may be more beneficial for my use case.

For example, a **Classification model (RandomTreeClassifier)**. This model is a better suit for many reasons including

1. Classification
    - By classifying into broad buckets, the model has a wider range for success.
2. Categorial Data
    - This model works much better with categorical data too, meaning our Bio data such as Age, Height, Foot can be used as features.

There are however trade offs that come with this change in approach, most noticeably in the output. The classification model is set up to bucket an individual as High / Medium / Low based on their profile and injury record. Therefore, we lose the granularity that came with the regression model in terms of exactly how many days this player was to miss. In practicality however, this trade off is less important than it may appear. In reality, the medical department probably doesn't need to predict the exact number of days a player will miss, they just need to know a simple categorisation to base their decisions off. Also, given the lower accuracy of that model, it is unlikely the decision maker would even consider the output knowing it is largely inaccurate. A broad categorisation is most likely enough to influence decisions in this area - therefore our classification model is best for our use case.

5.6 **Model Training**

Once we had established our model of choice and our dataset, we had to train and test our model. To do so, the dataset was split into 4 distinct sets:

- X_train: Training Data of the Features
- X_test: Testing Data of the Features
- Y_train: Training Data of the Target Variable (risk category)
- Y_test: Testing Data of the Target Variable (risk category)

This is a pivotal step in allowing the model to first learn and then test itself against unseen data. We then also encoded categorical variables into numerical variables to allow the model to account for them properly.

```
[1] from sklearn.model_selection import train_test_split

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=0)
```

Initially, I didn't use any 'class balancing', but having investigated the number of rows in each category before prediction, Low Risk contained many more examples than High Risk (119 vs 59). This is too imbalance, and therefore the model tended favourably to the Low category. Therefore, we included 'class_weight="balanced" to ensure the model accounted for this variation in its training.

5.7 **Classification Model Evaluation Metrics**

```
[34] from sklearn.metrics import classification_report

     print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
High	0.64	0.63	0.63	59
Low	0.75	0.82	0.78	119
Medium	0.72	0.61	0.66	72
accuracy			0.72	250
macro avg	0.70	0.69	0.69	250
weighted avg	0.71	0.72	0.71	250

Now we have trained the classification model, we can view how this performs and compare that to the original Regression model.

1. Overall accuracy = 0.72 (72%)
    - This implies the model correctly predicts the class 72% of the time
    - Given we have 3 classes, random guessing would provide a 33% success rate and therefore this offers a better than random prediction

We can also investigate how each individual category performed in terms of correct classification (Low, Medium, High).

- High: Precision = 0.64
    - This implies that of all players predicted as High, only 30% truly were

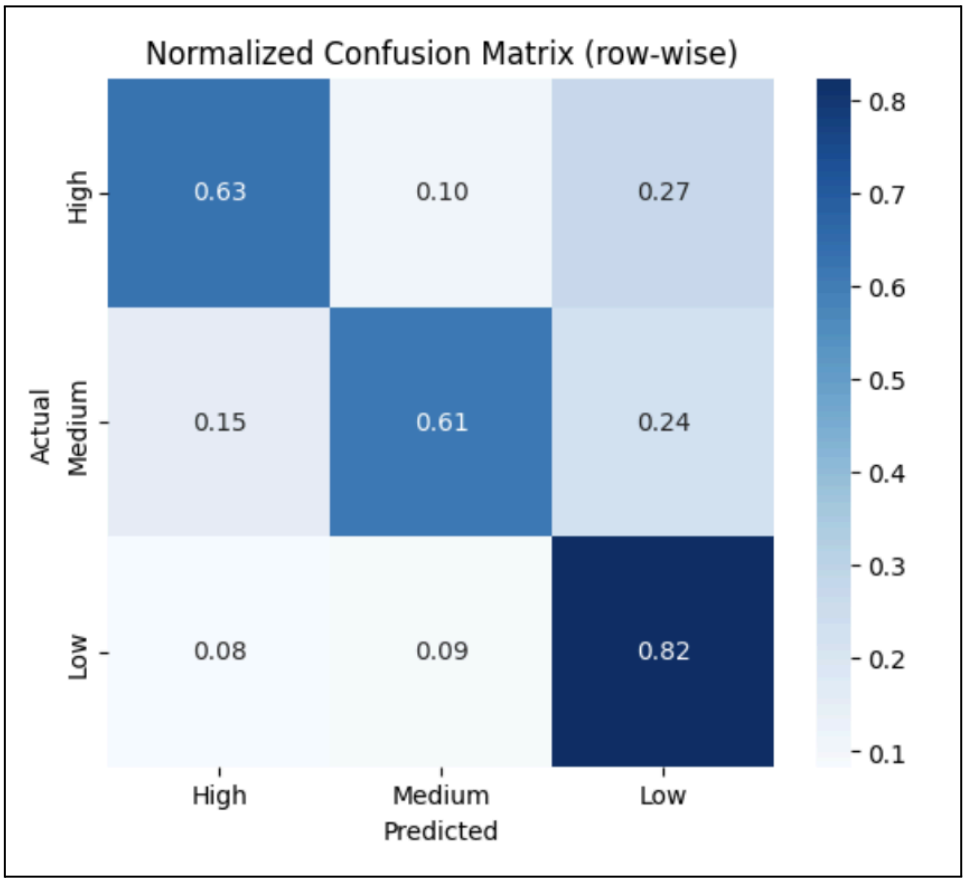
- Medium: Precision = 0.72
    - Ie, 37% of those predicted Medium were correct

- Low: Precision = 0.75
    - This is clearly the highest performing category.

Here, despite using different evaluation metrics, we can see clearly that this model outperforms the regression version we touched on earlier (both can be seen in GitHub). Our regression model only accounted for roughly 5% of variance in the target variable, yet the classification model predicts correctly 72% of the time.

The potential downside of the classification model is that you lose the granularity in the output, ie, it can't predict the number of days but only the category of risk. That being said, in modern medical departments it is assumed this category type approach would be more applicable in their decision making process, giving an indicator of potential risk rather than claiming to predict the exact number of days they will miss - especially when you account for "freak" accidents that cannot be accounted for.

We can also use a Confusion Matrix to explain how accurate a Classification Model is:



This confusion matrix shows the recall figures of each class and where incorrect predictions are most likely. We see our top left, centre, and bottom right squares as the highest proportions, this is positive as it shows that the correct prediction happens the majority of the time. Outside of this, we should consider the top right square which seems high also, 0.27, this means that 27% of actual High Risk players were misclassified as low risk. Therefore, if used in the recruitment decision, it is possible this model would allow some High Risk players in despite being predicted to be Low Risk.

5.8 **Feature Contribution / Importance**

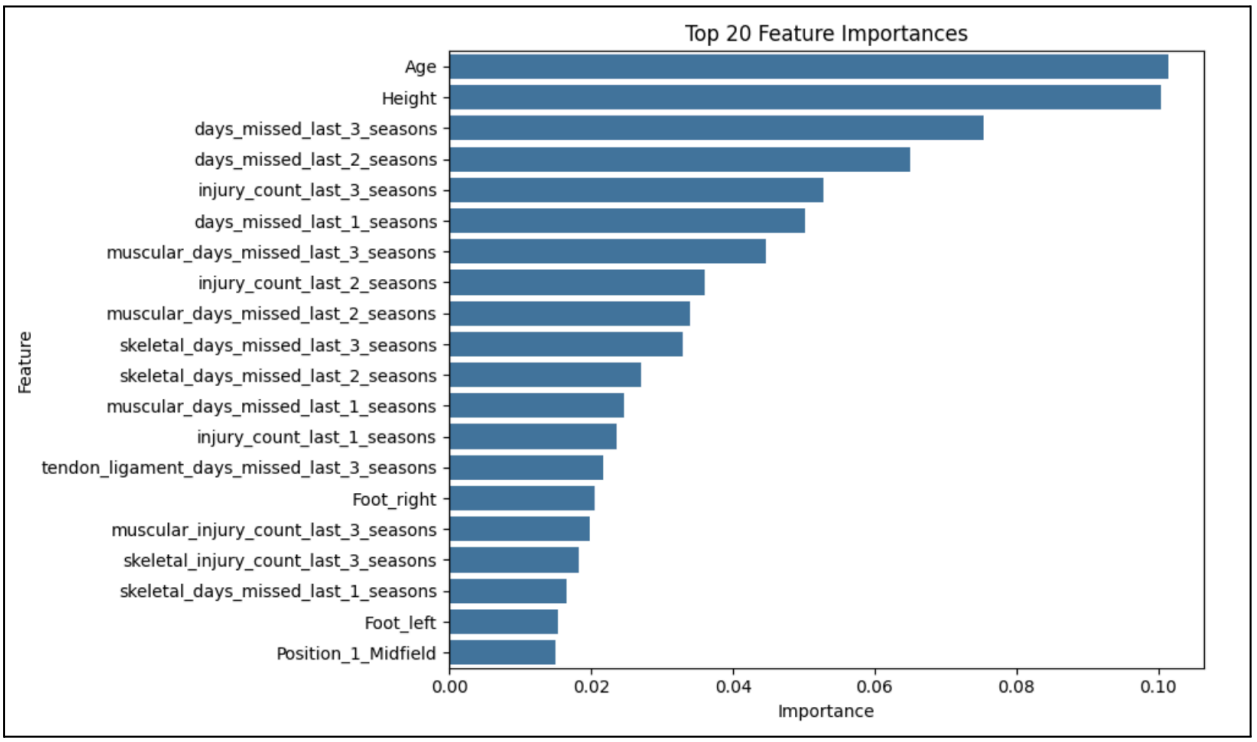
We can now investigate the individual contribution our features make in terms of predicting the output. Here, a greater coefficient implies greater impact on the risk category.

feature	importance
Age	0.101357
Height	0.100421
days_missed_last_3_seasons	0.075425
days_missed_last_2_seasons	0.064967
injury_count_last_3_seasons	0.052735
days_missed_last_1_seasons	0.050170
muscular_days_missed_last_3_seasons	0.044695
injury_count_last_2_seasons	0.035977
muscular_days_missed_last_2_seasons	0.034008
skeletal_days_missed_last_3_seasons	0.032909
skeletal_days_missed_last_2_seasons	0.026990
muscular_days_missed_last_1_seasons	0.024722
injury_count_last_1_seasons	0.023665
tendon_ligament_days_missed_last_3_seasons	0.021672
Foot_right	0.020569
muscular_injury_count_last_3_seasons	0.019795

Age: This follows some logic as injuries become more frequent and typically take longer to recover from as players age.
Height: This doesn't follow as expected, however there is some argument height could be correlated with 'higher risk' positions such as Central Defense. We would look to investigate this feature in future work.
Days Missed Last 3 Seasons: This follows as hypothesised, a poor injury history record correlates with a higher chance of injury moving forward.

Other Interesting Notes:
Days Missed over Injury Count: Days missed appear as a more relevant feature here than explicit injury count.
Muscular Injuries: Of the injury categories, a history of muscular injuries tends to be the best predictor of future injury risk. This is understandable as skeletal injuries tend to be one off situations and less likely to recur.

This figure shows a better visualisation of each feature used in the model and its relevant in determining the final categorisation:



6. Results Discussion

Now that we have a trained model predicting at relatively strong levels of accuracy, it is time to apply this to a real world situation and application.

Alexander Isak is highly rumoured to be a part of a £100m+ transfer away from Newcastle United to Liverpool FC. Liverpool FC as pioneers would certainly have identified Isak from a combination of traditional scouting methods and data led analysis to support the transfer decision. But as we mentioned in the introduction, injury is one of the key factors as to why transfers fail which may not be being considered as much as it should be.

So, let's create a new row of data for Alexander Isak's current profile and allow the model to predict his risk category for the upcoming season.

```
# Convert dictionary to DataFrame
isak_df = pd.DataFrame(isak_data)

# Ensure columns match the model's training data
missing_cols = set(X.columns) - set(isak_df.columns)
for col in missing_cols:
    isak_df[col] = 0  # add missing columns if any

# Reorder columns to match training features
isak_df = isak_df[X.columns]

# Predict risk category
predicted_risk = model.predict(isak_df)

print(f"Predicted risk category for {isak_df.get('player_name', ['Alexander Isak'])[0]}:
```

```
Predicted risk category for Alexander Isak: High
```

As you can see from the snippet above, the model predicts Alexander Isak to be a 'High' risk category player for the upcoming season. That implies that Isak may miss over 60+ days in the following season - with a model that predicts roughly 70% accuracy. Specifically when looking at the High category from the confusion matrix the model predicts at a slightly lower 63% accuracy, potentially with the addition of load data we could get a more accurate prediction.

Therefore, this model may actually act to caution the transfer of Alexander Isak to Liverpool, or even reject it, depending on how the model is used in the decision making process. Particularly when spending massive volumes of money on transfer fees, like in this instance, the buying club can't afford to take risks when it comes to player unavailability due to injury and so this prediction should be accounted for.

So looking at Isak's profile, what is it specifically about him that contributes to his 'High' risk injury categorisation? According to Transfermarkt, Isak has missed 206 days in the past 3 seasons, with 172 of these days as a result of muscular injury. Looking more recently too, Isak has 3 separate injuries contributing to 43 days missed. Looking back at the feature importance chart, we can see these are highly important features in the model - particularly the days missed last 3 seasons which is particularly high for Alexander Isak.

7. Conclusions and Future Work

Conclusions
As we can see from our evaluation metrics, we have a model that can provide a decent estimate in terms of the Risk Category players fall into for injury, however before being applied in the real world we should look to improve its predictive ability as mentioned in the future work below. That being said, we have a solid framework and statistical process that with development can provide key insight to medical departments particularly when making decisions on recruitment. The categorisation and easy translation into a range of 'Expected Days Missed' makes it an intuitive tool for medical professionals to use and apply in their decision making process.

Limitations / Future Work
Our model provides a solid starting point and structure, however, given the evaluation of our model and its performance there is certainly scope to iterate and improve the predictive ability of the model. Specifically, as mentioned previously, we should look to incorporate a greater range of features that may help in prediction but are currently omitted.

The main area I would look to address is that of 'Load' that currently isn't accounted for in the model. Features such as minutes played in the past 1/2/3 seasons are likely to help improve the prediction model, as that is often an early indicator of injuries to come. Perhaps we should also look at minutes played before a certain age too, there is often debate around overworking younger players and not allowing them to fully develop physically before asking them to play 40/50 games per season. The theory is that players playing too many minutes at a younger age tend to pay for it in the back end of their career, whether that is through early retirement or through a greater injury risk.

We could also look to incorporate more detailed load event data than previously mentioned, in terms of sprints, turns, duels, lands and tackles. Typically, it is in the high intensity movements where muscular injuries occur, whilst actions like Tackles and Lands can lead to other types of injury such as Ligament or Skeletal. Players who perform more of these actions may result in a higher injury likelihood.

Another area would be to expand the training data set. Given I was limited to open data sources and web scraping, the analysis is limited to top 5 European leagues and subsequently only players where we have rich enough data. Working internally at a club may allow for richer and wider training data to be provided, allowing a more accurate prediction model than what we have.

Finally, in terms of adoption for key stakeholders as a tool they use regularly in their day to day operation, an intuitive user interface should be developed to make it usable by non technical stakeholders. Currently, whilst the output of the model 'High', 'Medium', 'Low' is easy to interpret and apply, the actual process of using the model may be too complex and require technical support. Perhaps a basic app where the user can apply the model to potential transfer recruits to easily determine the risk category is the best case.