Fantasy Premier League

Abdelrahman Ahmed Mohamed Abdelfattah Omran , Ahmed Nasser Abdelkareem
Computer and systems department, faculty of engineering, alexandria university

Problem Statement

With over 6 million players, Fantasy Premier League is the biggest Fantasy Football game in the world, at this game users can choose their team which consists of real players from the Premier League each with a value and the user shouldn't exceed the amount of money in his bank and based on the actual performance of these real players the user gets points, one thing is that each weak the user has the option to replace one of his players by another and this action is very critical and has a lot of factors to take into consideration like players current form , his next matches difficulty , player's value , ownership (how many other players have this player) and many other factors (like team captain) that makes one player better than another.

Dataset

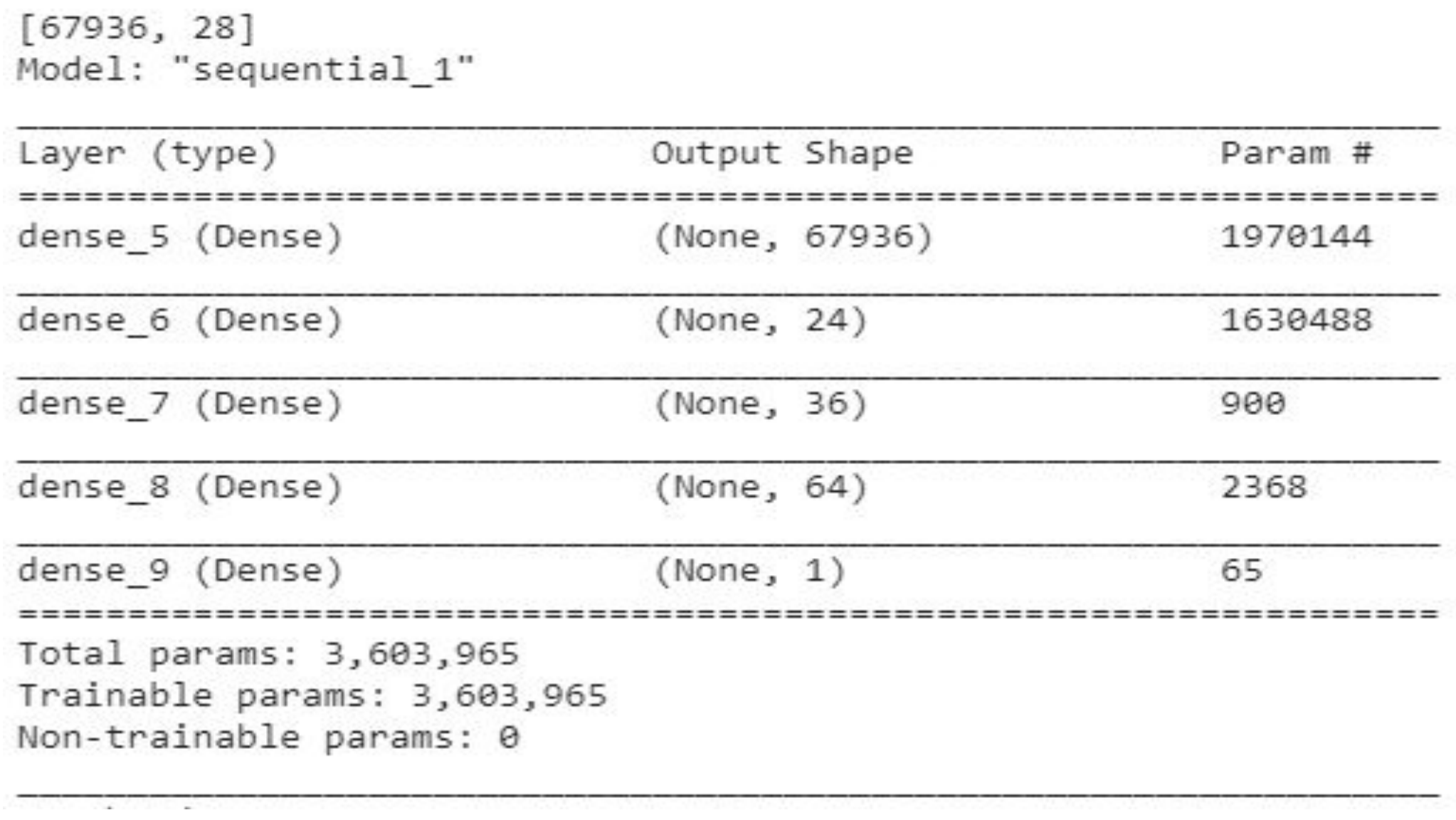
Detailed description of the Dataset:

The data folder contains the data from the previous 3 seasons as well as the current season. It is structured as follows:

- season/cleaned_players.csv : The overview stats for the season
- season/gws/gw_number.csv : GW-specific stats for the particular season
- season/gws/merged_gws.csv : GW-by-GW stats for each player in a single file
- season/players/player_name/gws.csv : GW-by-GW stats for that specific player
- season/players/player_name/history.csv : Prior seasons history stats for that specific player.

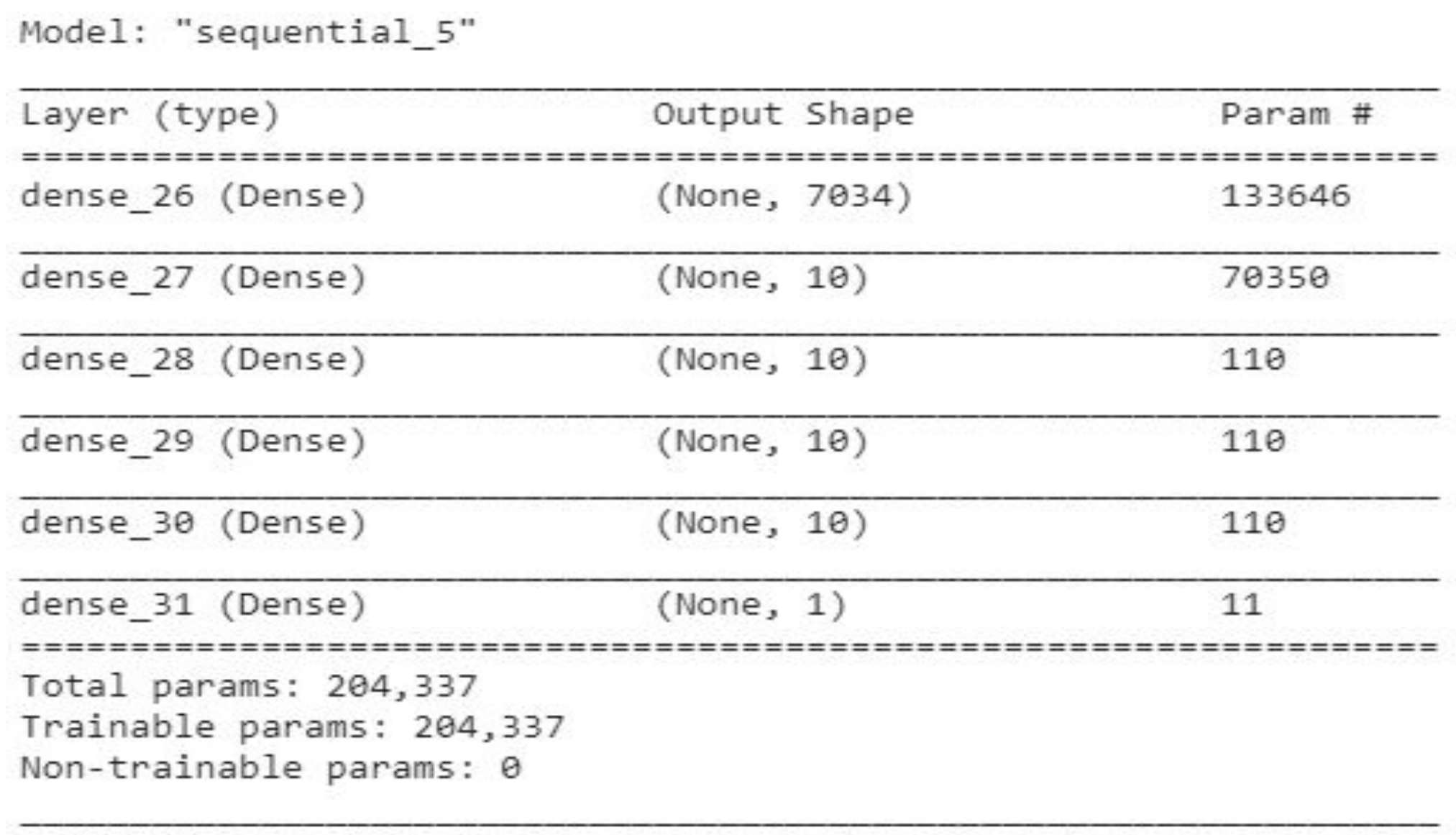
Models

Goalkeeper model :



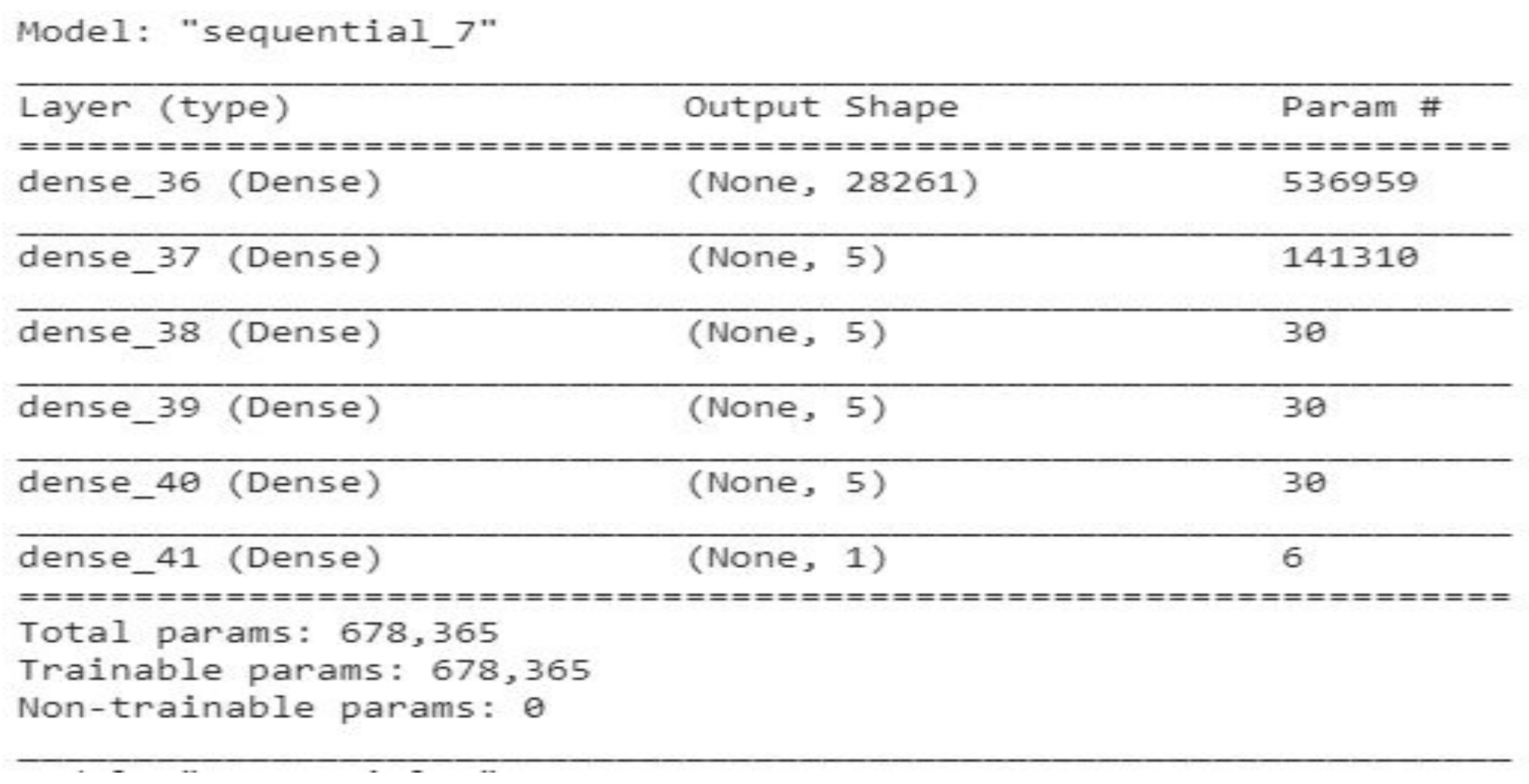
```
[67936, 28]
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_5 (Dense)              (None, 67936)             1970144
_________________________________________________________________
dense_6 (Dense)              (None, 24)                1630488
_________________________________________________________________
dense_7 (Dense)              (None, 36)                900
_________________________________________________________________
dense_8 (Dense)              (None, 64)                2368
_________________________________________________________________
dense_9 (Dense)              (None, 1)                 65
=================================================================
Total params: 3,603,965
Trainable params: 3,603,965
Non-trainable params: 0
_________________________________________________________________
```

def model :



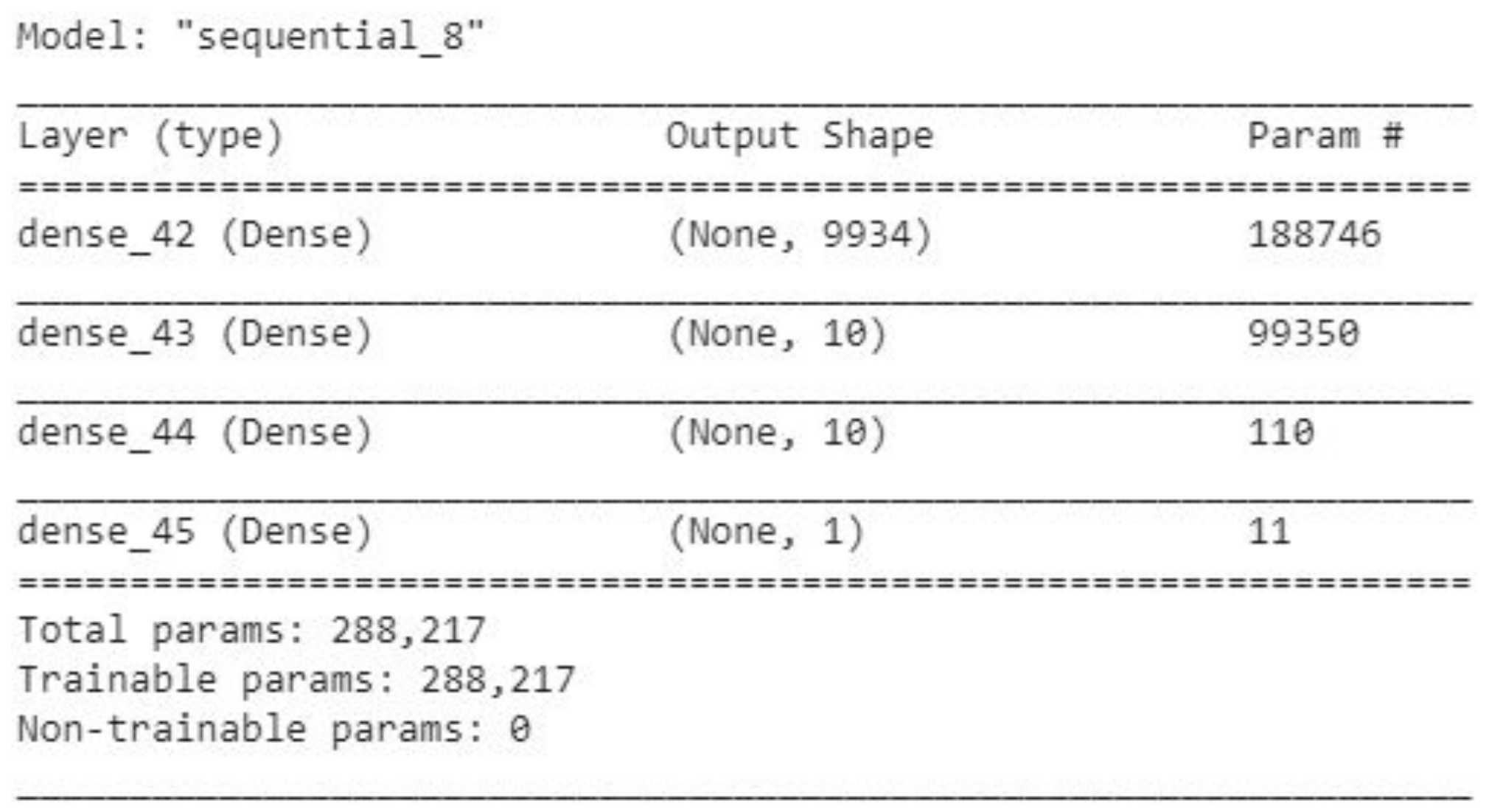
```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_26 (Dense)             (None, 7034)              133646
_________________________________________________________________
dense_27 (Dense)             (None, 10)                70350
_________________________________________________________________
dense_28 (Dense)             (None, 10)                110
_________________________________________________________________
dense_29 (Dense)             (None, 10)                110
_________________________________________________________________
dense_30 (Dense)             (None, 10)                110
_________________________________________________________________
dense_31 (Dense)             (None, 1)                 11
=================================================================
Total params: 204,337
Trainable params: 204,337
Non-trainable params: 0
_________________________________________________________________
```

midfield model



```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_36 (Dense)             (None, 28261)             536959
_________________________________________________________________
dense_37 (Dense)             (None, 5)                 141310
_________________________________________________________________
dense_38 (Dense)             (None, 5)                 30
_________________________________________________________________
dense_39 (Dense)             (None, 5)                 30
_________________________________________________________________
dense_40 (Dense)             (None, 5)                 30
_________________________________________________________________
dense_41 (Dense)             (None, 1)                 6
=================================================================
Total params: 678,365
Trainable params: 678,365
Non-trainable params: 0
_________________________________________________________________
```

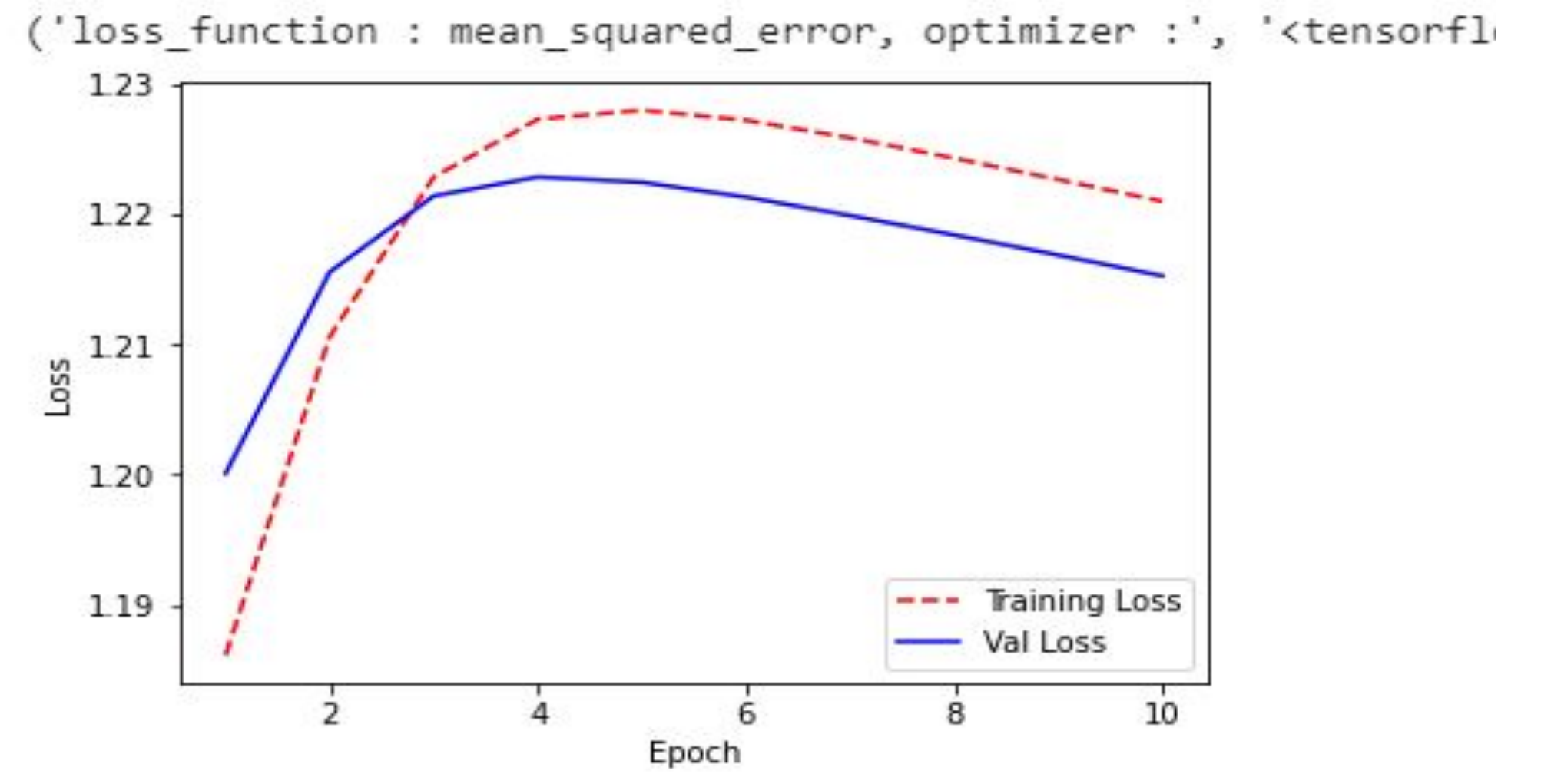
strikers model ::



```
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_42 (Dense)             (None, 9934)              188746
_________________________________________________________________
dense_43 (Dense)             (None, 10)                99350
_________________________________________________________________
dense_44 (Dense)             (None, 10)                110
_________________________________________________________________
dense_45 (Dense)             (None, 1)                 11
=================================================================
Total params: 288,217
Trainable params: 288,217
Non-trainable params: 0
_________________________________________________________________
```

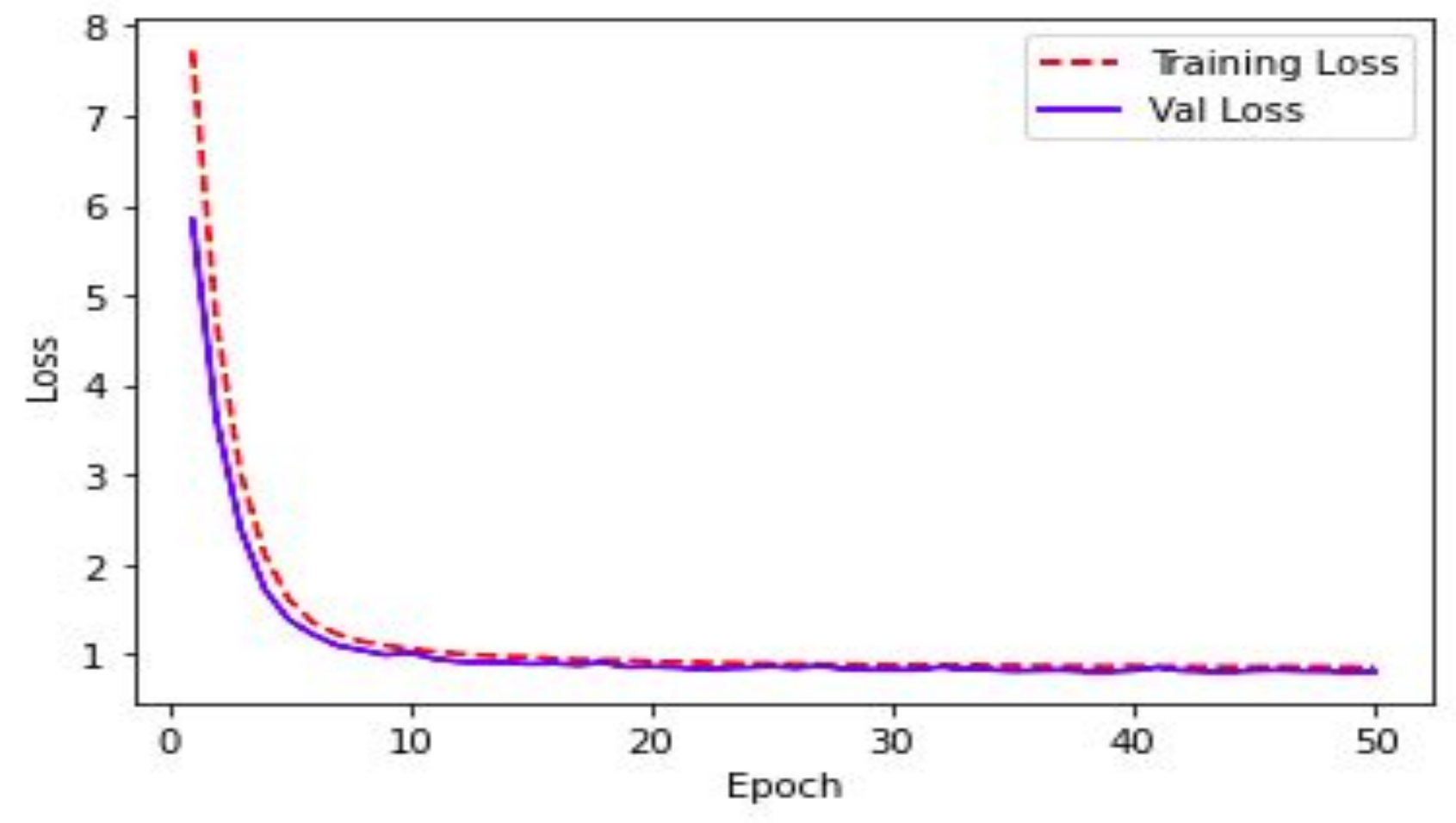
Results

- **Base model loss:**

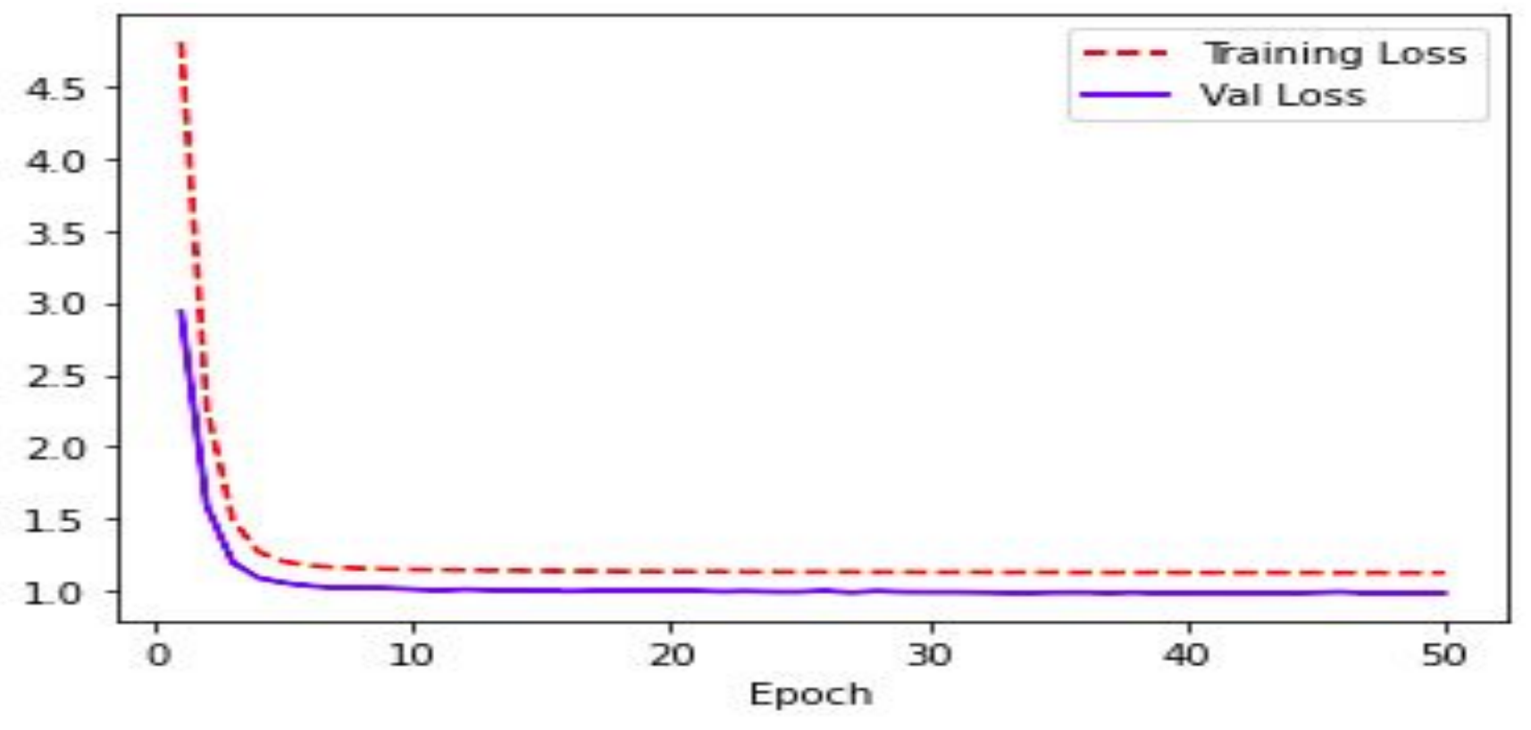


```
('loss_function : mean_squared_error, optimizer :', '<tensorfl
```

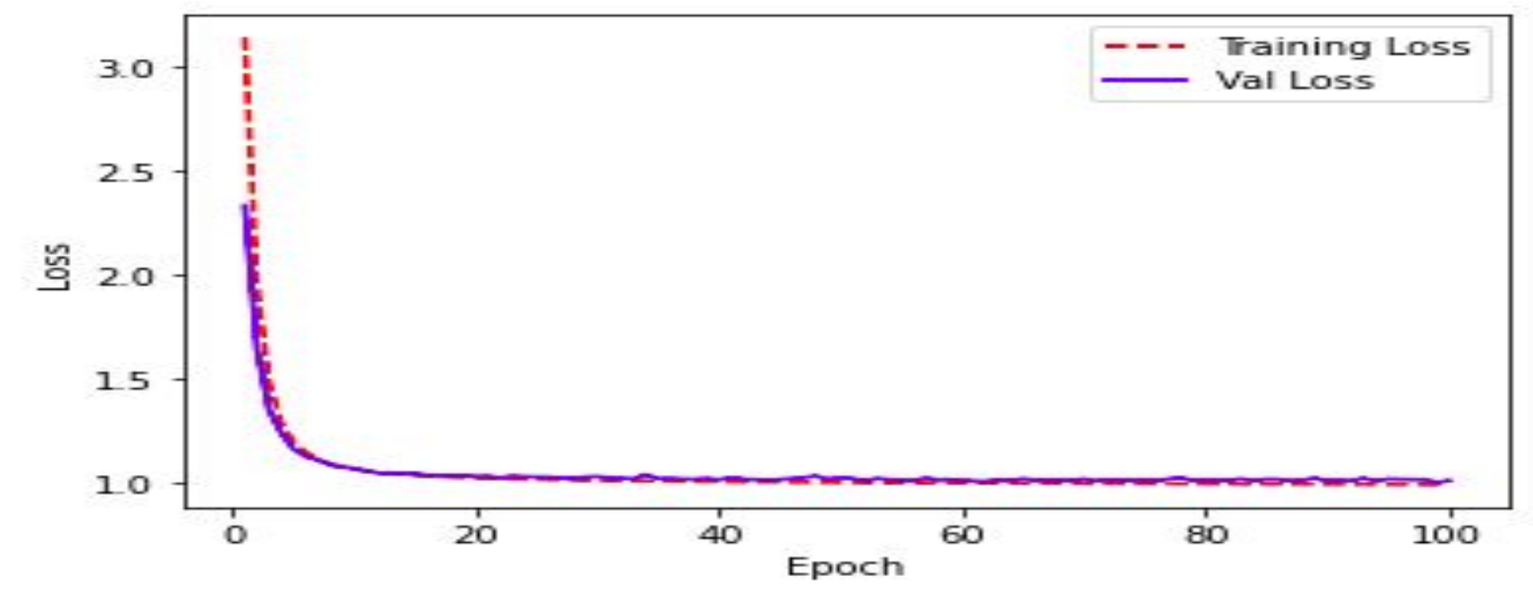
- **The loss for each model :**
  - GK



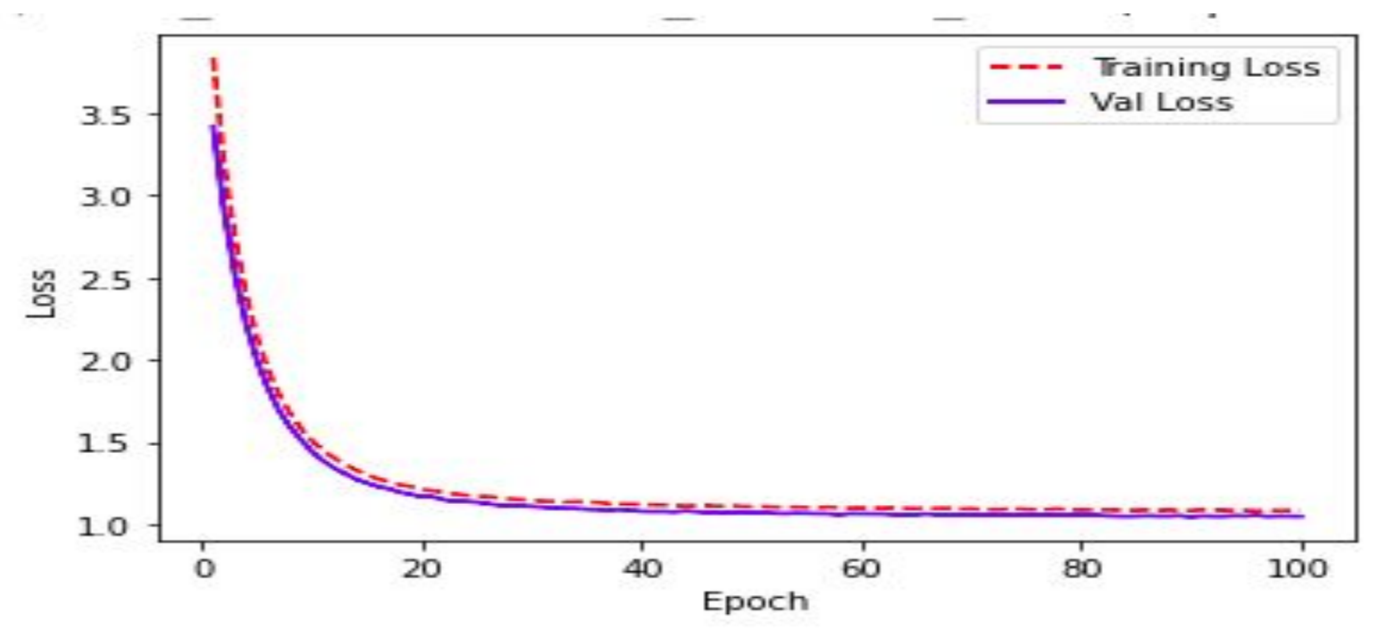
  - Def



  - Mid



  - Stk

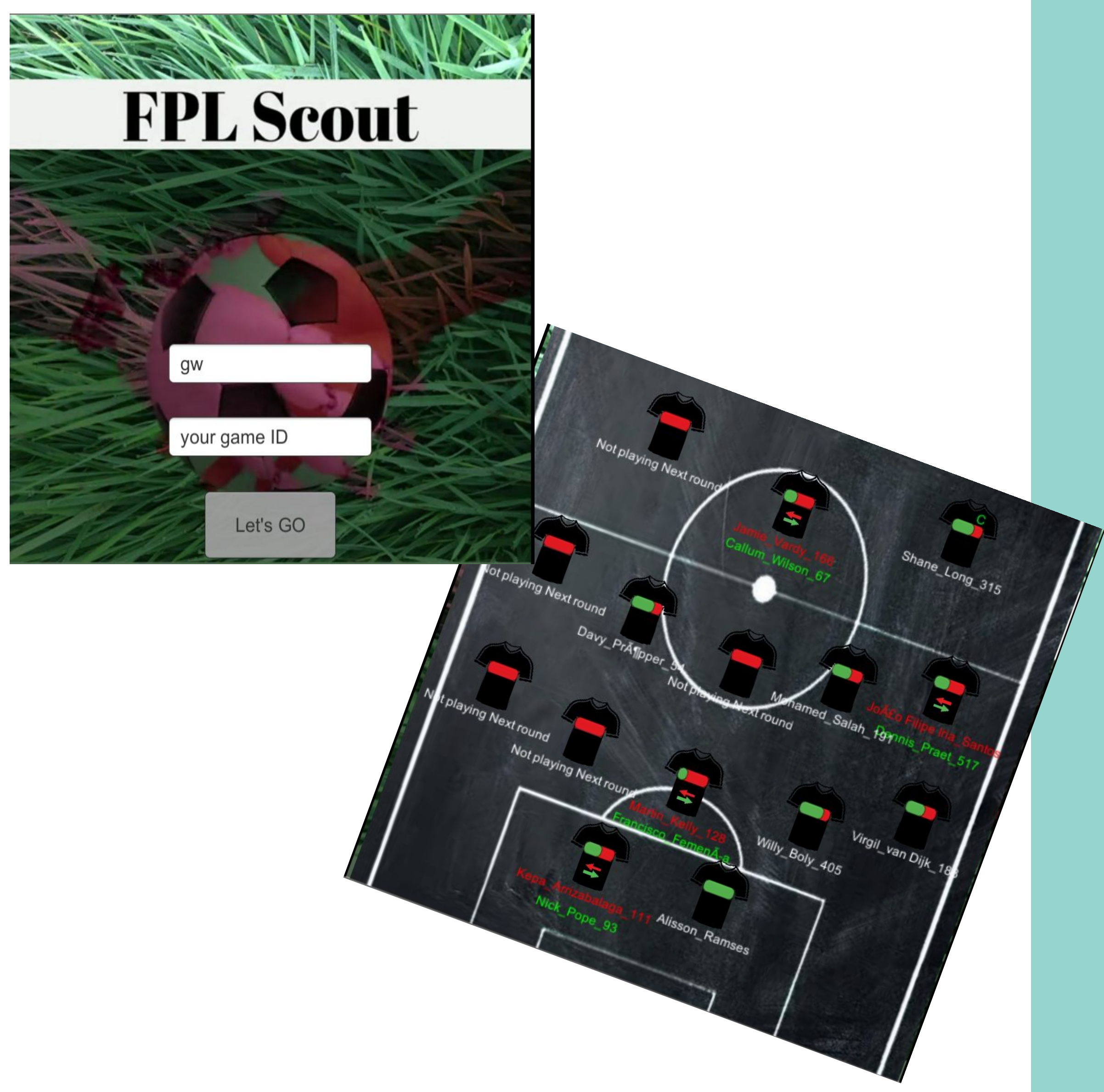


Conclusion

- **Future Work**
  - Study the weights of each feature to determine the best Input for each model per position.
  - Publish The game.

- **Lessons**
  - Check if the train and test data have the same shape and from the same distribution.

- **our game**
  - we deployed the model to a server and created a unity game that fetch the predictions for substitutions and suggest it to the user :



References

[1] Dataset

[2] Website