INDIVIDUAL ASSIGNMENT: MACHINE LEARNING TUTORIAL ON PLAYER SELECTION USING RANDOM FOREST

SRIHARI MOHAN – 23069726

GITHUB - https://github.com/sriharimohan/Machine-Learning-Assignment.git

---

1. INTRODUCTION

This tutorial demonstrates how to use the Random Forest classification algorithm to predict whether a football player is likely to be selected for their national team, based on various physical, performance, and financial attributes. This analysis is how a real-world national team or club select players based on their previous performances for club and country, goal contributions, fouls, cards, etc. Analysis of players by using machine learning algorithm practically helps coaches to understand the current form of the players and to know about their past performances, all under one roof, which can help teams to get their best players at the moment for maximizing their win percentage.

---

2. ML TECHNIQUE USED: RANDOM FOREST

Random Forest is an ensemble learning method that constructs multiple decision trees and merges their results to improve accuracy and control overfitting. It's particularly useful for classification tasks like ours, where we aim to make binary predictions (Selected vs. Not Selected). Each tree in the forest votes, and the most popular class becomes the model's prediction.

Key advantages include:

- Handles large datasets with higher dimensionality.
- Maintains accuracy even when a large proportion of data is missing.
- Helps reduce variance and avoid overfitting.

---

3. DATASET AND ITS FEATURES:

I used the **FIFA 20 player dataset**, which includes comprehensive stats of professional footballers. From this dataset, we selected the following features:

- age, height_cm, weight_kg

- overall, potential
- value_eur, wage_eur
- pace, shooting, passing, dribbling, defending, physic

Engineered a binary target variable:
is_national_team_level = 1 if overall rating >= 80, else 0
This serves as a proxy for whether a player is good enough for national team selection.

After dropping rows with missing values, we split the data into training (80%) and testing (20%) sets.

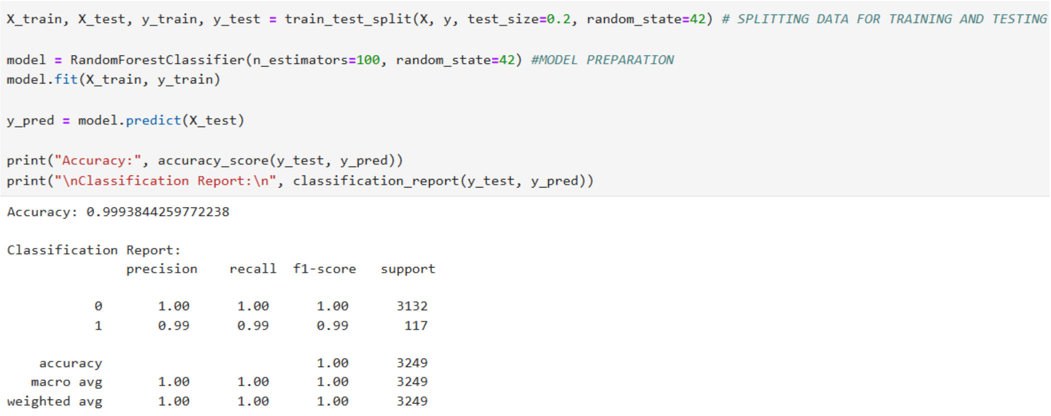
---

4. MODEL BUILDING AND EVALUATION:

RandomForestClassifier from scikit-learn is used, with 100 trees (n_estimators=100) and a fixed random_state for reproducibility.

Evaluation Metrics:

- **Accuracy Score**: Percentage of correctly predicted instances.
- **Classification Report**: Includes precision, recall, and F1-score.
- **Confusion Matrix**: Visualized to show the true positives, false positives, etc.
- **Feature Importance Plot**: Shows which features most influence the model's predictions.



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42) # SPLITTING DATA FOR TRAINING AND TESTING

model = RandomForestClassifier(n_estimators=100, random_state=42) #MODEL PREPARATION
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.9993844259772238

Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      3132
           1       0.99      0.99      0.99       117

    accuracy                           1.00      3249
   macro avg       1.00      1.00      1.00      3249
weighted avg       1.00      1.00      1.00      3249
```

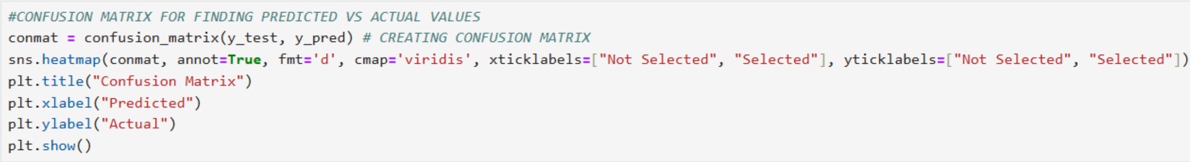
The Random Forest Classifier achieved an outstanding test accuracy of approximately 99.94%, with both classes demonstrating near-perfect precision and recall. Despite a significant class imbalance, the model reliably predicted national team selection with high confidence. These results indicate that the selected features and the model are highly effective for this classification task.

---

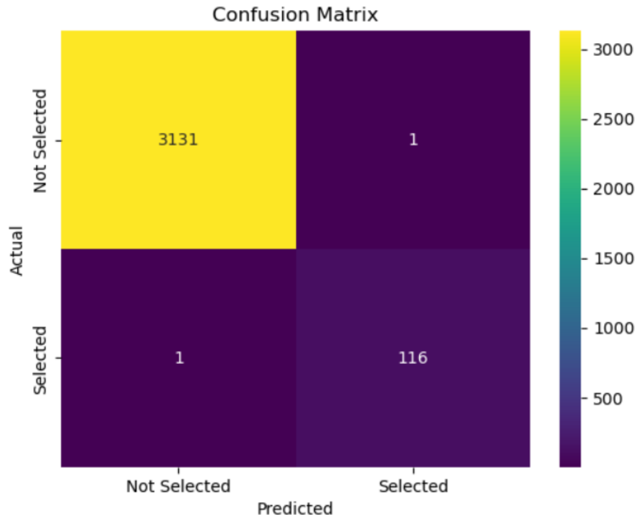
5. ANALYSIS OF RESULTS

Let's look at the plot we got from the results for deeper understanding.

CONFUSION MATRIX:

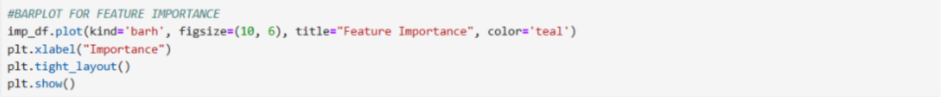


```
#CONFUSION MATRIX FOR FINDING PREDICTED VS ACTUAL VALUES
conmat = confusion_matrix(y_test, y_pred) # CREATING CONFUSION MATRIX
sns.heatmap(conmat, annot=True, fmt='d', cmap='viridis', xticklabels=["Not Selected", "Selected"], yticklabels=["Not Selected", "Selected"])
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

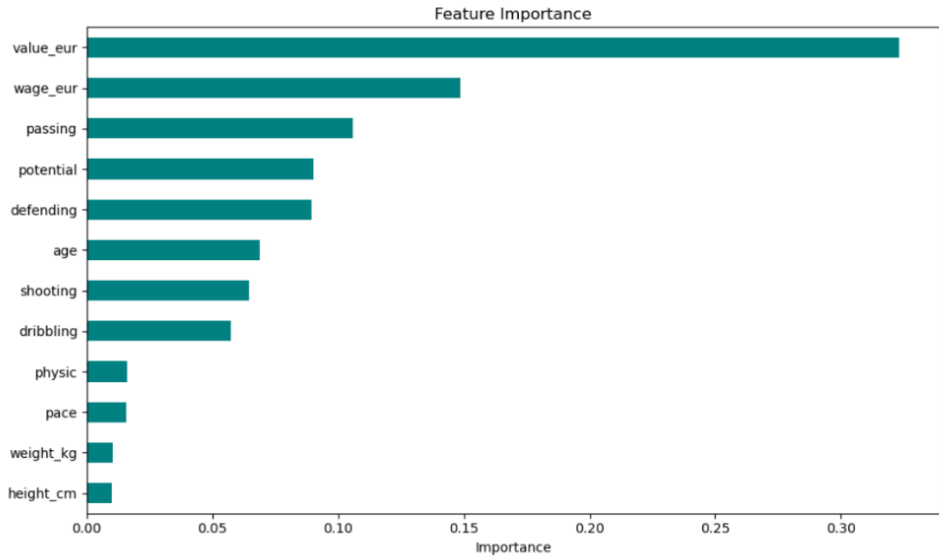


The confusion matrix shows that out of 3249 players, the Random Forest model made only 2 incorrect predictions. It correctly predicted 3131 players as not suitable for national selection and 116 players as suitable. This confirms the model's high accuracy and precision, making it highly effective for predicting player selection at the national team level.

FEATURE IMPORTANCE USING BARPLOT:

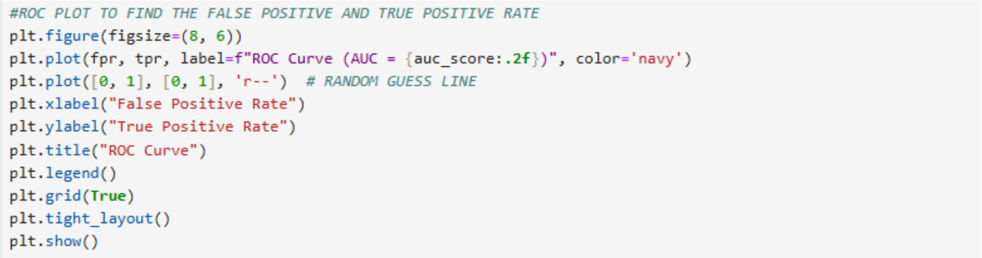


```
#BARPLOT FOR FEATURE IMPORTANCE
imp_df.plot(kind='barh', figsize=(10, 6), title="Feature Importance", color='teal')
plt.xlabel("Importance")
plt.tight_layout()
plt.show()
```

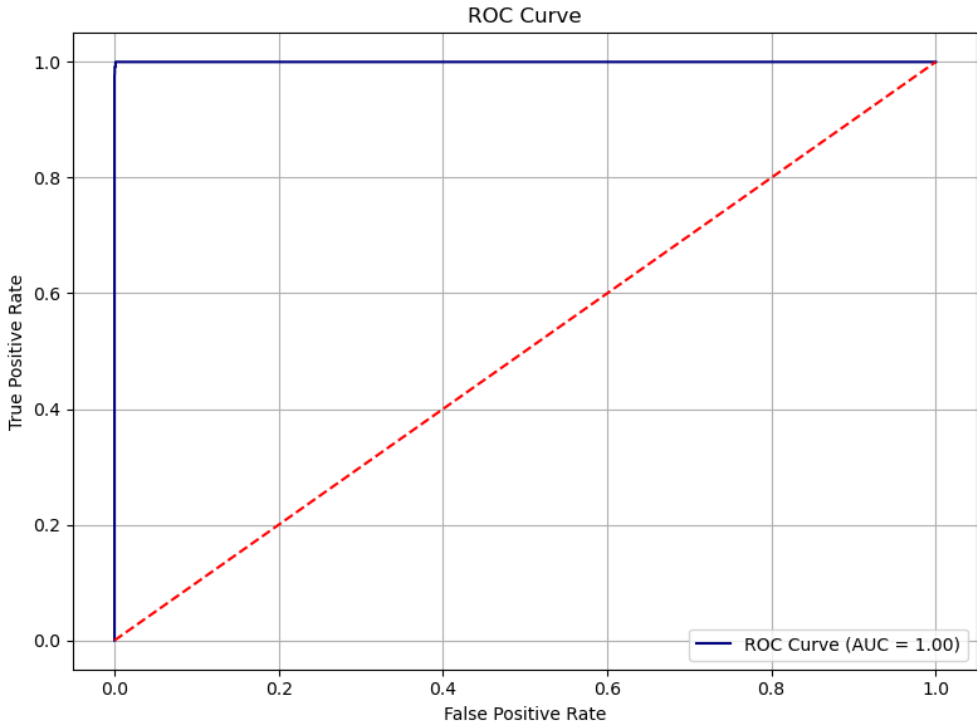


The Feature Importance plot indicates that value_eur and wage_eur are the most influential features in predicting a player's likelihood of being selected for the national team. This aligns with real-world football dynamics, where highly valued and well-paid players tend to be high performers and more likely to receive national call-ups. Skill-based metrics such as passing, defending, and potential also played significant roles. Interestingly, physical attributes like height_cm and weight_kg had minimal impact, suggesting the model prioritized technical and market-based indicators over raw physical stats.

ROC CURVE:

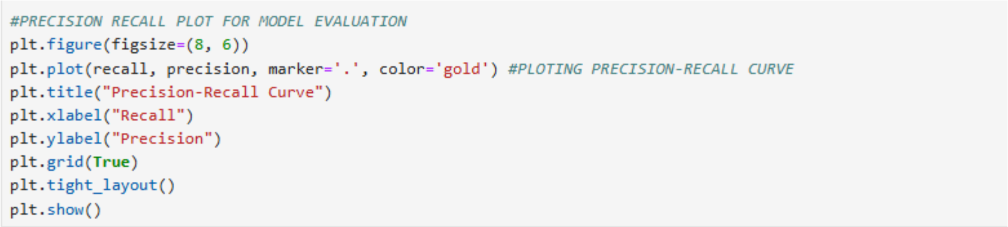


```
#ROC PLOT TO FIND THE FALSE POSITIVE AND TRUE POSITIVE RATE
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label=f"ROC Curve (AUC = {auc_score:.2f})", color='navy')
plt.plot([0, 1], [0, 1], 'r--')  # RANDOM GUESS LINE
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

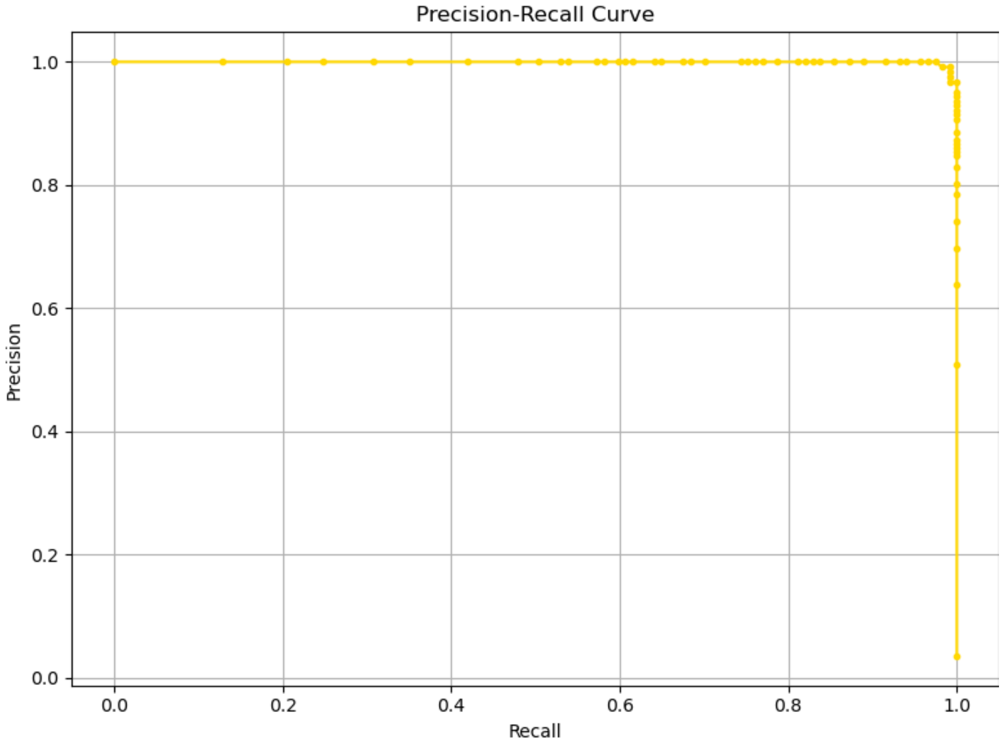


The ROC curve demonstrates the classification performance of the model across all thresholds. With an AUC (Area Under the Curve) score of 1.00, the model exhibits perfect discrimination capability — it correctly distinguishes between players selected for the national team and those not selected. This indicates exceptional predictive performance, with minimal false positives or false negatives.

PRECISION-RECALL CURVE:



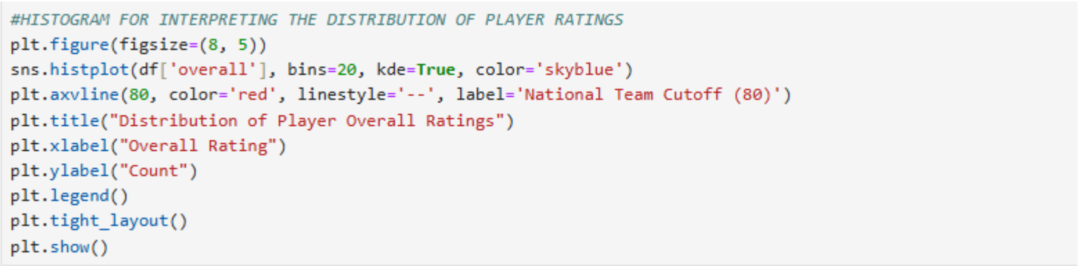
```
#PRECISION RECALL PLOT FOR MODEL EVALUATION
plt.figure(figsize=(8, 6))
plt.plot(recall, precision, marker='.', color='gold') #PLOTING PRECISION-RECALL CURVE
plt.title("Precision-Recall Curve")
plt.xlabel("Recall")
plt.ylabel("Precision")
plt.grid(True)
plt.tight_layout()
plt.show()
```



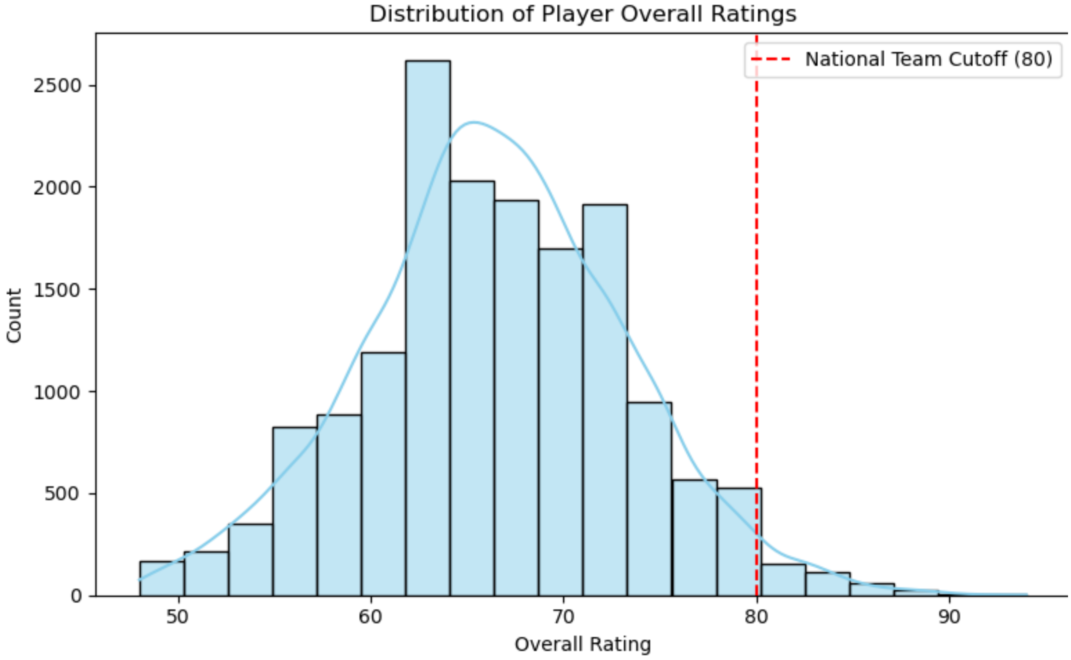
The Precision-Recall Curve illustrates the model's ability to identify selected players with high confidence. The curve remains flat near a precision score of 1.0 throughout, indicating that the model makes very few false positive errors. Even as recall increases, the precision remains high, showcasing strong overall model performance in correctly selecting national team candidates while minimizing incorrect selections.

DISTRIBUTION OF PLAYER RATINGS IN HISTOGRAM:

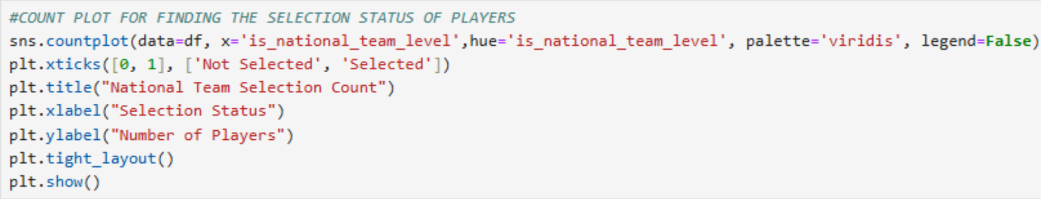
The histogram of player overall ratings reveals a slightly right-skewed distribution with most ratings clustered between 60 and 75. A vertical line at 80 marks the national team selection threshold, indicating that only a small fraction of players qualifies. his visual supports the observed class imbalance, where highly rated players suitable for national selection are relatively rare compared to the general player population.



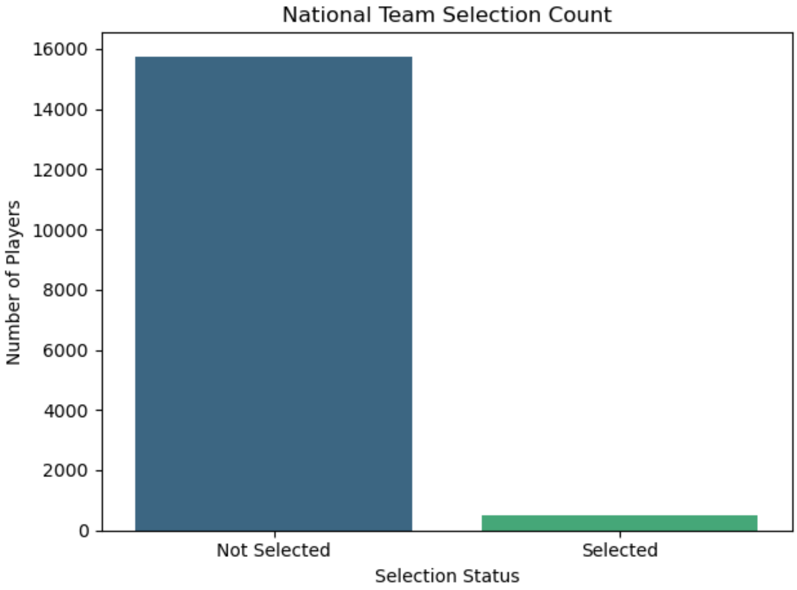
```
#HISTOGRAM FOR INTERPRETING THE DISTRIBUTION OF PLAYER RATINGS
plt.figure(figsize=(8, 5))
sns.histplot(df['overall'], bins=20, kde=True, color='skyblue')
plt.axvline(80, color='red', linestyle='--', label='National Team Cutoff (80)')
plt.title("Distribution of Player Overall Ratings")
plt.xlabel("Overall Rating")
plt.ylabel("Count")
plt.legend()
plt.tight_layout()
plt.show()
```



SELECTED COUNT USING COUNT PLOT:

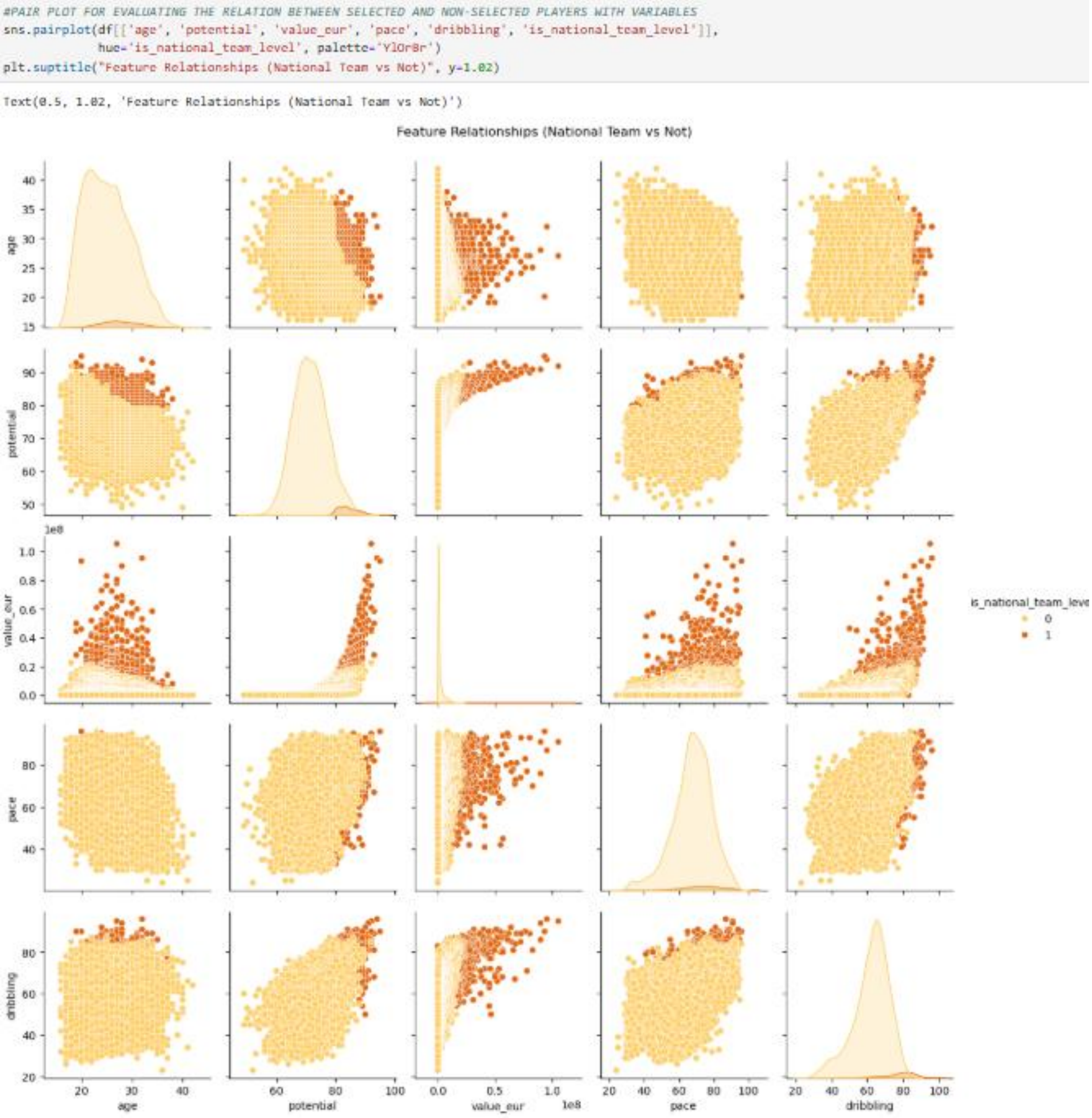


```
#COUNT PLOT FOR FINDING THE SELECTION STATUS OF PLAYERS
sns.countplot(data=df, x='is_national_team_level',hue='is_national_team_level', palette='viridis', legend=False)
plt.xticks([0, 1], ['Not Selected', 'Selected'])
plt.title("National Team Selection Count")
plt.xlabel("Selection Status")
plt.ylabel("Number of Players")
plt.tight_layout()
plt.show()
```



The class distribution plot confirms a significant imbalance in the target variable. The vast majority of players (over 15,000) are not selected for national teams, while only a small minority (around 600) are selected. This imbalance presents a challenge for classification models, as it can bias predictions toward the majority class. Proper evaluation metrics beyond accuracy, such as precision-recall and ROC-AUC, are therefore critical.

PAIRPLOT FOR FEATURE RELATION:



```
#PAIR PLOT FOR EVALUATING THE RELATION BETWEEN SELECTED AND NON-SELECTED PLAYERS WITH VARIABLES
sns.pairplot(df[['age', 'potential', 'value_eur', 'pace', 'dribbling', 'is_national_team_level']],
             hue='is_national_team_level', palette='YlOrBr')
plt.suptitle("Feature Relationships (National Team vs Not)", y=1.02)
```

```
Text(0.5, 1.02, 'Feature Relationships (National Team vs Not)')
```

The pair plot reveals distinct feature patterns between selected and non-selected players. Selected players tend to exhibit higher potential, value, dribbling, and pace. Market value especially stands out as a strong differentiator, with national team players having considerably higher valuations. These visual trends support the selection of these features for the Random Forest model and indicate that they carry predictive signals relevant to national team selection.

6. CONCLUSION

This project shows how machine learning, specifically Random Forest, can support talent identification in football. Coaches and scouts can use such models to identify promising players based on data-driven insights. This type of analysis can be really helpful for real-world teams and clubs to know their team strength and prepare for the tournament based on their precise tactics by selecting players who comply with the manager's vision to tactically provide the team with best results. This type of analysis may comply with real world development up to some extent after all, it is a machine prediction and not a super power like human brain.

7. REFERENCES

- Scikit-learn documentation: https://scikit-learn.org/
- FIFA 20 player dataset (Kaggle): https://www.kaggle.com/stefanoleone992/fifa-20-complete-player-dataset
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324
- Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432