Football players dataset report

**Defenision:**

In this project, a machine learning approach was applied to analyze a dataset of football players, focusing on predicting their market value and wages. The project involved the development of regression models aimed at accurately estimating the market value of players based on various attributes such as age, position, nationality, and performance metrics.

Additionally, the market values were categorized into five distinct classes: very low, low, medium, high, and very high. Similar categorizations were applied to the players' wages. Classification models were then developed to predict these categories. By categorizing and modeling both market value and wage, the project explored the relationships between a player's attributes and their financial standing within the football industry.

**Problem:**

The football transfer market is a dynamic and competitive arena where player valuations are influenced by a wide range of factors, such as performance, age, potential, and prevailing market trends. Accurately assessing a player's market value is essential for clubs, scouts, and agents to make informed decisions during transfers, contract negotiations, and talent acquisition. The challenge lies in developing reliable models that can predict a player’s market value and wage based on these multifaceted factors, ensuring that stakeholders have a data-driven approach to managing player investments and optimizing financial outcomes.

**Methods:**

The project was developed using Python. Pandas, a popular Python library for data manipulation and analysis, was extensively used throughout the project and Google Collab, a cloud-based Jupiter notebook environment, was used as the development platform.

Data Preparation and Feature Engineering:

* Converted "value euro" to millions for easier interpretation.
* Created age brackets and potential vs overall rating features.
* Selected relevant features for modeling
* Handled missing values using SimpleImputer.

Regression Models for Predicting "value euro":

We've used several models to predict the market of the players:

* Random Forest Regressor.
* K-Nearest Neighbors (KNN) Regressor.
* Support Vector Regression (SVR).
* Linear Regression.
* Decision Tree Regressor.

For each model:

* Used a preprocessing pipeline with SimpleImputer and StandardScaler.
* Evaluated performance using Root Mean Squared Error (RMSE) and R-squared score.
* Provided example predictions

Classification Models for Predicting "value euro" categories:

We categorized "value euro" into bins (Very Low, Low, Medium, High, Very High), by using the following models:

* Decision Tree Classifier.
* K-Nearest Neighbors (KNN) Classifier.
* Support Vector Machine (SVM) Classifier.
* XGBoost Classifier.
* Naive Bayes Classifier.

We used SelectKBest for feature selection, evaluated performance using accuracy scores and checked for overfitting by comparing training and test accuracies.

Classification Models for Predicting "wage euro" categories:

We've used the same methods in the "value euro" prediction to predict the players' wage.

**Description of the dataset:**

Our dataset contains information about football players, with around 18,000 rows and 51 columns. Each row represents an individual player and includes various attributes such as:

Personal data – name, date of birth, height, weight. (10 columns)

Physical info – jumping, vision, mark, acceleration. (17 columns)

Player rating – potential, rating, skills, moves. (6 columns)

Financial info – wage, value player, release clause. (3 columns)

National info – national jersey, position, team. (3 columns)

Technical info – curve, volleys, crossing. (12 columns)

**Explanation of the research:**

**First section:**

1. תמונה שמכילה טקסט, גופן, צילום מסך

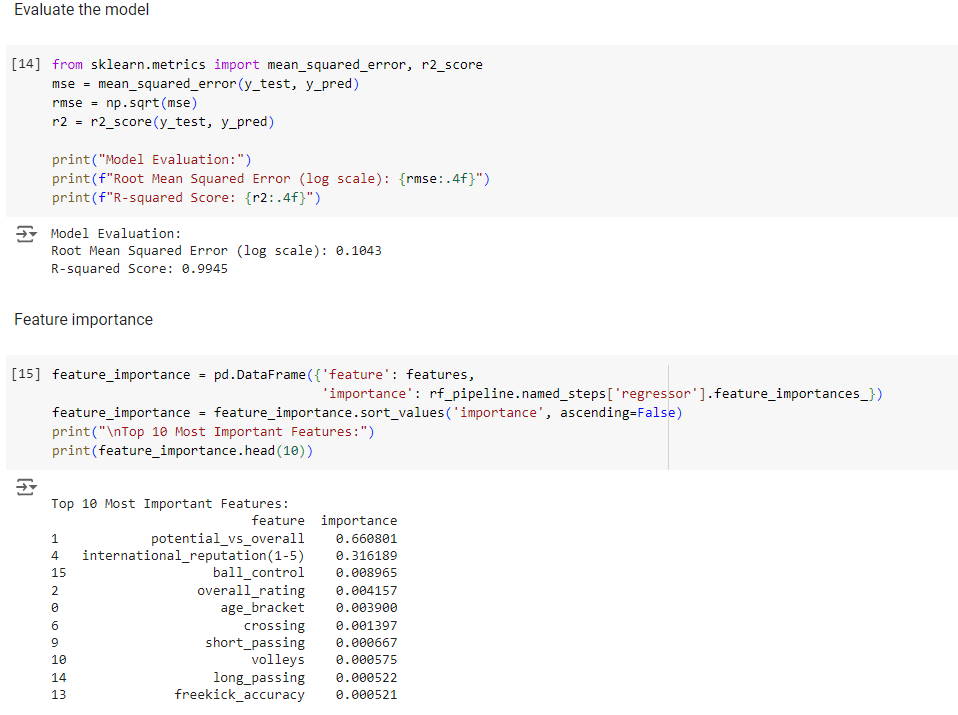
   התיאור נוצר באופן אוטומטיLoad the data set of football players data:
2. תמונה שמכילה טקסט, צילום מסך, גופן, קו

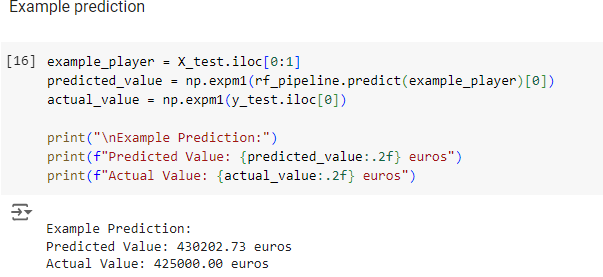
   התיאור נוצר באופן אוטומטיConvert the player's market value making the values more convenient and readableתמונה שמכילה טקסט, גופן, צילום מסך

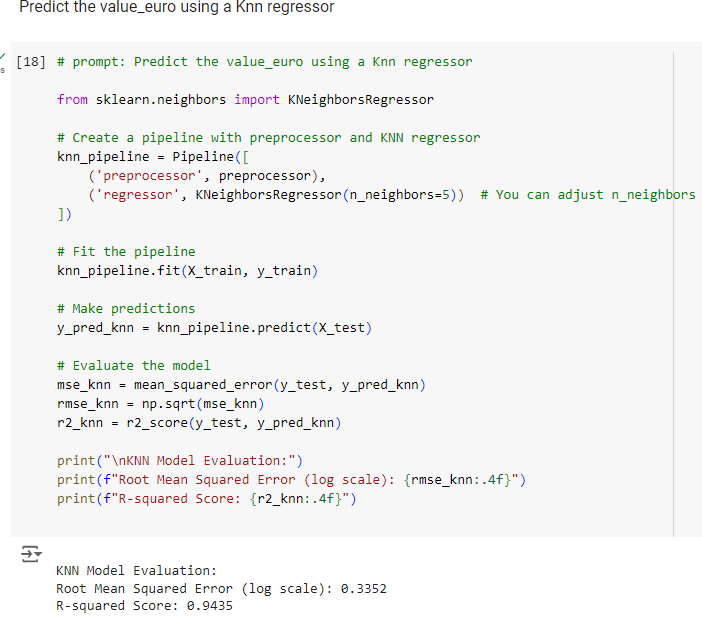
   התיאור נוצר באופן אוטומטי.
3. Feature engineering, feature selection, and preprocessing steps for a machine learning model predicting football players' market values. It starts by categorizing players' ages into different brackets, represented by labels 0, 1, 2, and 3, to capture various career stages. A new feature, potential\_vs\_overall, is created by calculating the difference between a player's potential and current overall rating, indicating potential growth. Next, relevant features are selected, including age\_bracket, potential\_vs\_overall, overall\_rating, wage\_euro, international\_reputation(1-5), and release\_clause\_euro, along with other technical attributes related to specific skills like crossing, finishing, and dribbling. The target variable, value\_euro, is then prepared by filling any missing values with the median value. Finally, a log transformation is applied to the target variable to normalize its distribution and reduce skewness, making it more suitable for regression modeling.
4. Splitting the dataset, creating a preprocessing pipeline, and fitting a machine learning model for predicting football players' market values. It begins by splitting the data into training and testing sets, with 20% of the data reserved for testing, ensuring the model can be evaluated on unseen data (14,363 rows for training set and 3,591 rows for test set). Preprocessing pipeline is built to handle numerical and categorical features separately.



1. Evaluation and interpretation of a Random Forest regression model (rf\_pipeline) for predicting football player values. Starts by making predictions on the test set and then evaluates the model's performance using Root Mean Squared Error (RMSE) and R-squared score. The RMSE provides a measure of the average prediction error, while the R-squared score indicates how well the model explains the variance in the target variable.

Finally, the code provides an example prediction for a single player from the test set.



1. Using another regression models in order to compare which one of them gives the most accurate score.

A screenshot of a computer program

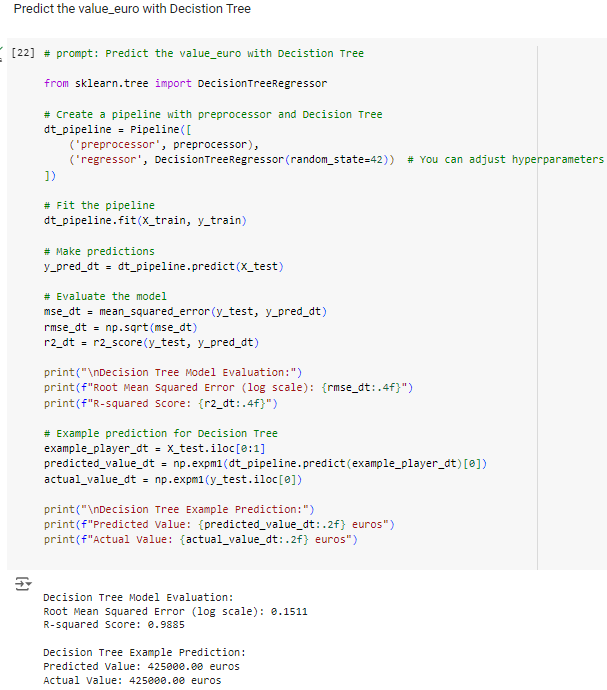
Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

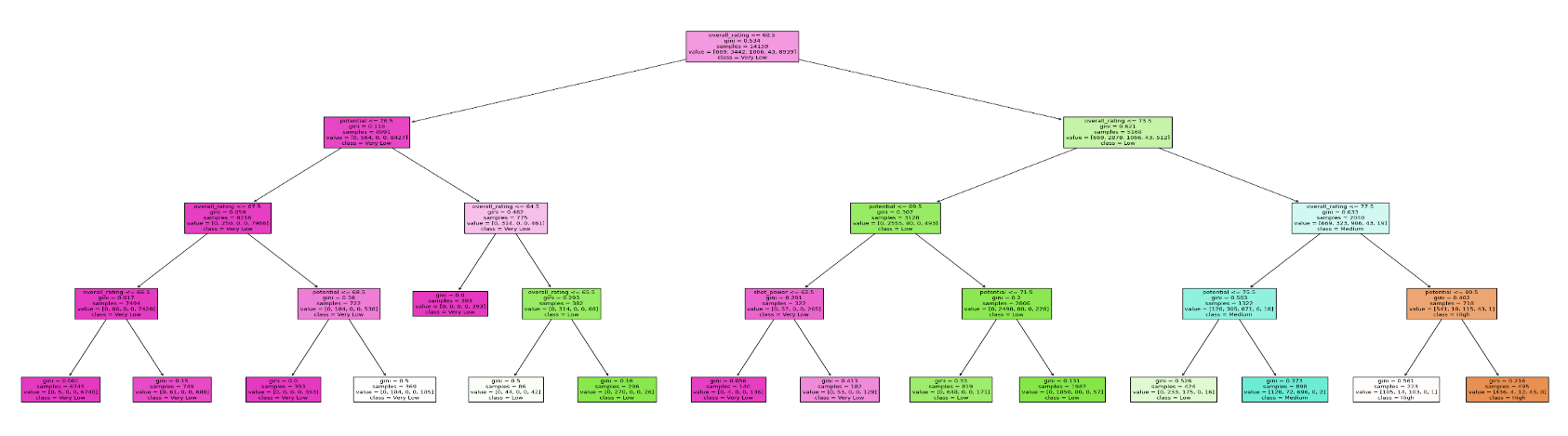


**Second section:**

1. Predicting the market value of a player by using classification models. Firstly, we prepared the data by categorizing the "value euro" target class to 5 labels – ‘Very Low’, ‘Low’, ’Medium’, ‘High’, and Very High’.

תמונה שמכילה טקסט, צילום מסך, גופן

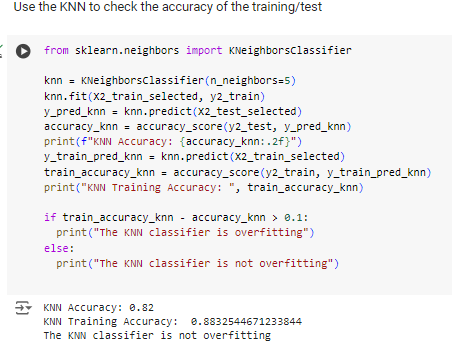
התיאור נוצר באופן אוטומטיWe are splitting the target from the rest of the features and then replacing the target with the bins and label we created. From that we are using Selector to get the best 10 features and then apply the train & test on them.

1. תמונה שמכילה טקסט, גופן, צילום מסך

   התיאור נוצר באופן אוטומטיCreating a decision tree by using sklearn
2. Evaluate the performance of the classifier and check for overfitting.



1. Applying more classification models to check which one of them gives the best accuracy score.



A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Third Section:**

1. Predicting the wage value of a player by using classification models. Firstly, we prepared the data by categorizing the "wage euro" target class to 5 labels – ‘Very Low’, ‘Low’, ’Medium’, ‘High’, and Very High’.

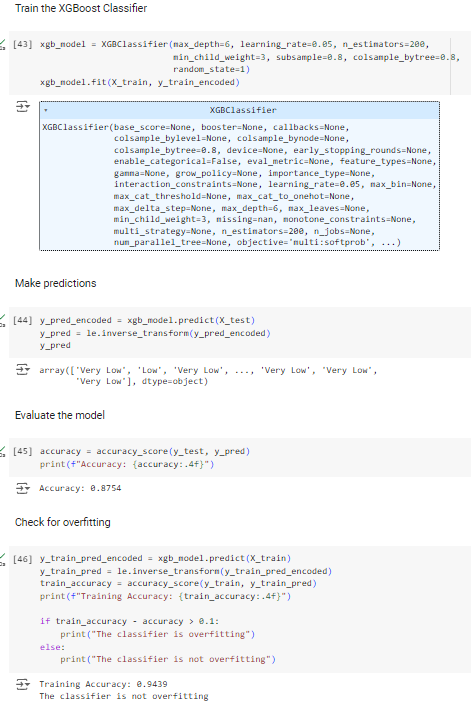
תמונה שמכילה טקסט, צילום מסך, גופן

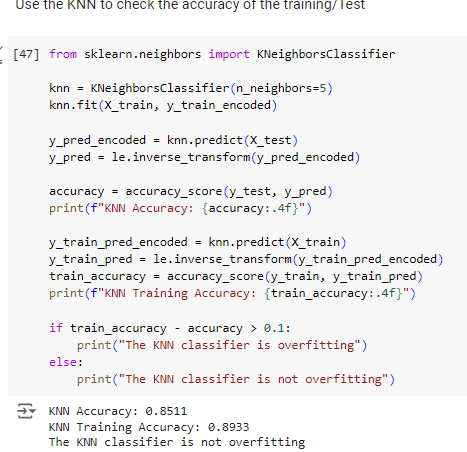
התיאור נוצר באופן אוטומטיWe are splitting the target from the rest of the features and then replacing the target with the bins and label we created. From that we are using Selector to get the best 20 features and then apply the train & test on them

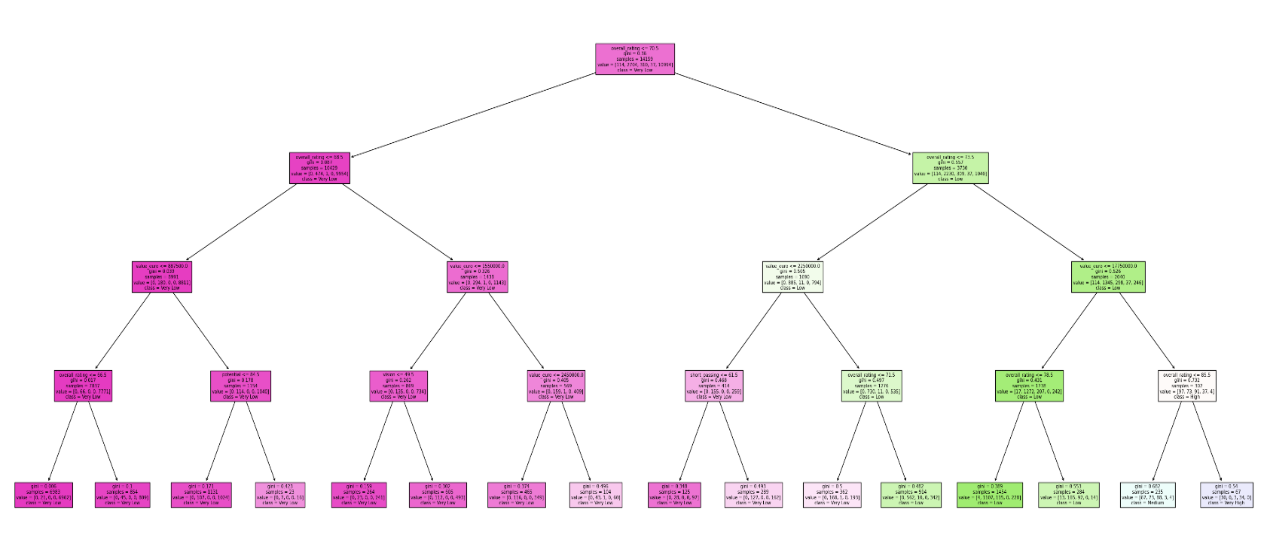
1. We using LabelEncoder from the sklearn. preprocessing module in Python to convert categorical labels into a numerical format that machine learning algorithms can work with. The train & test remains with 80%-20% (14,363 rows for training set and 3,591 rows for test set).

A screenshot of a computer program

Description automatically generated

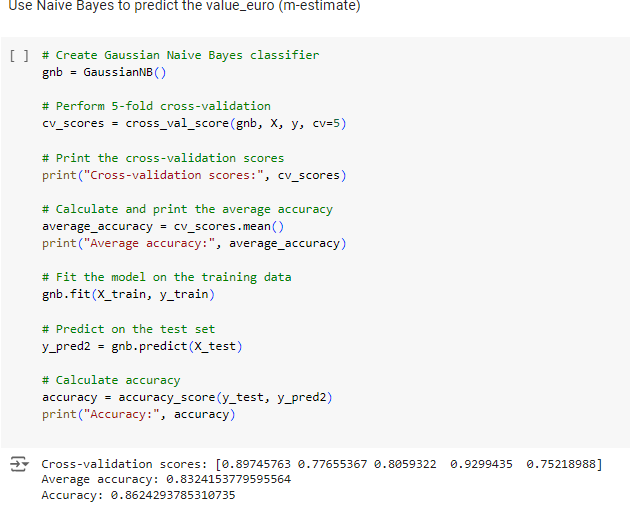
1. Evaluate the performance of the classifiers and check for overfitting.



A screenshot of a computer

Description automatically generatedA screenshot of a computer code

Description automatically generated



**Results**

Regression Results

| **Model** | **R2** | **RMSE** | **Actual Predict** | **Model Predict** |
| --- | --- | --- | --- | --- |
| **RandomForest** | 0.993 | 0.1 | 42.5 | 43.2 |
| **KNN** | 0.943 | 0.33 | 42.5 | 48.3 |
| **SVR** | 0.972 | 0.23 | 42.5 | 37.2 |
| **Linear** | 0.955 | 0.29 | 42.5 | 37.0 |
| **Decision Tree** | 0.988 | 0.15 | 42.5 | 42.5 |

Player Value Classification

* XGBoost performs best with 0.99 accuracy on training and 0.95 on test data.
* DecisionTree follows closely with 0.911 on training and 0.91 on test data.
* KNN and SVM show lower performance.
* All models show a little mismatches between the test and train accuracy (Overfitting quite well here)

Cross-Validation Results

* 5-fold cross-validation was performed.
* Accuracy scores range from 75% to 92%.
* Average accuracy is 0.83, with overall accuracy of 0.85.
* There's some inconsistency in model performance across different data subsets.

Wage Value Classification

* XGBoost again performs best with 0.944 on training and 0.875 on test data.
* KNN, DecisionTree, and SVM all perform similarly on test data (around 0.85-0.87).
* All models show a little mismatches between the test and train accuracy (Overfitting quite well here)

Cross-Validation Results:

* 5-fold cross-validation was performed.
* Accuracy scores range from 75% to 92%.
* Average accuracy is 0.83, with overall accuracy of 0.86.
* There's some inconsistency in model performance across different data subsets.

**Analysis:**

1. Regression:
   * RandomForest performs best with the highest R2 (0.993) and lowest RMSE (0.1).
   * Decision Tree follows closely, with very accurate predictions.
   * KNN, SVR, and Linear regression show good performance but with higher error rates.
   * All models show high R2 values, indicating they explain a large portion of the variance in the data.
2. Classification:
   * XGBoost consistently outperforms other models for both player value and wage value classification.
   * There are signs of overfitting across all models, which may need to be addressed.
   * The cross-validation results show some inconsistency, indicating that the model's performance varies depending on the subset of data used.

**Conclusions:**

1. Best Models:

For regression: Random Forest is the best choice for predicting value euro.

For classification: XGBOOST consistently outperforms other models for both value euro and wage euro categories.

2. Predictability of value euro vs wage euro:

value euro appears to be slightly more predictable than wage euro:

Classification for value euro achieves higher peak accuracy (0.95 vs 0.875 for wage euro).

Regression models for value euro show extremely high R² values.

However, both value euro and wage euro are highly predictable, with

all models performing well.

3. Model Consistency:

Classification models for both value euro and wage euro show good consistency between training and test accuracies, indicating good generalization.

The M-Estimate cross-validation results are nearly identical for both tasks, suggesting similar underlying patterns.

4. Feature Importance:

The high predictability of both value euro and wage euro suggests that the features in your dataset are strongly correlated with these target variables.

5. Model Selection:

For regression tasks, ensemble methods (Random Forest) outperform single models.

For classification tasks, boosting methods (XGBOOST) perform best,

but other models also show good results.

6. Robustness:

The consistent performance across different models and tasks suggests that the dataset is robust and well-structured for these