Report justification and results

1. Problem 1:

- Approach: *problem-1* has the task of collecting data about players who play in the *Premier League* from the website *FBref.com*. It is designed to:
    - *Access the website*: Connect to FBref to obtain data.
    - *Search for the league*: Identify the Premier League page.
    - *Get team data*: Browse the list of teams, then access each team page.
    - *Extract player data*: Collect various types of statistics, including:
        - Standard Stats.
        - Goalkeeping.
        - Shooting.
        - Passing.
        - Goal and Shot Creation.
        - Defensive Actions.
        - Possession.
        - Miscellaneous Stats.
    - *Save results*: The final data is saved to a CSV file (results.csv).
- Results: After running the program, a *data table* containing detailed information about players in the *Premier League* will be saved to the results.csv file, including:
    - *Standard Stats*: Player name, Nation, Team, Position (Pos), Age, Matches Played (MP), Starts, Minutes played (Min), Goals scored (Gls_st), Assists (Ast_st), Yellow cards (CrdY_st), Red cards (CrdR_st), Expected goals (xG_st), Expected assists (xAG_st), Progressive carries (PrgC_st), Progressive passes (PrgP_st), Progressive passes received (PrgR_st), Goals per 90 minutes (Gls_st_per90), Assists per 90 minutes (Ast_st_per90), Expected

goals per 90 minutes (xG_st_per90), Expected assists per 90 minutes (xAG_st_per90).

- *Goalkeeping Stats*: Goals against per 90 minutes (GA90_gk), Save percentage (%) (Save%_gk), Clean sheet percentage (%) (CS%_gk), Penalty save percentage (%) (Save%_gk_pen).
- *Shooting Stats*: Shots on target percentage (%) (SoT%_sh), Shots on target per 90 minutes (SoT/90_sh), Goals per shot (G/Sh_sh), Average shot distance (Dist_sh).
- *Passing Stats*: Completed passes (Cmp_pas), Pass completion rate (%) (Cmp%_pas), Short pass completion rate (%) (Cmp%_pas_S), Medium pass completion rate (%) (Cmp%_pas_M), Long pass completion rate (%) (Cmp%_pas_L), Key passes (KP_pas), Passes into final third (1/3_pas), Passes into penalty area (PPA_pas), Crosses into penalty area (CrsPA_pas), Progressive passes (PrgP_pas).
- *Goal and Shot Creation Stats*: Shot-creating actions (SCA_gsc), Shot-creating actions per 90 minutes (SCA90_gsc), Goal-creating actions (GCA_gsc), Goal-creating actions per 90 minutes (GCA90_gsc).
- *Defensive Actions Stats*: Tackles (Tkl_def), Tackles won (TklW_def), Challenges attempted (Att_def), Challenges lost (Lost_def), Blocks (Blocks_def), Shots blocked (Sh_def), Passes blocked (Pass_def), Interceptions (Int_def).
- *Possession Stats*: Touches (Touches_pos), Touches in defensive penalty area (Def Pen_pos), Touches in defensive third (Def 3rd_pos), Touches in middle third (Mid 3rd_pos), Touches in attacking third (Att 3rd_pos), Touches in attacking penalty area (Att Pen), Dribbles attempted (Att_pos), Successful dribble percentage (%) (Succ%_pos), Dispossessed percentage (%) (Tkld%_pos), Ball carries (Carries_pos), Total progressive carrying distance (PrgDist_pos), Progressive carries (PrgC_pos), Carries into final third (1/3_pos), Carries into penalty area (CPA_pos), Miscontrols (Mis_pos), Dispossessed (Dis_pos), Passes received (Rec_pos), Progressive passes received (PrgR_pos).

- *Miscellaneous Stats*: Fouls committed (Fls_mis), Fouls drawn (Fld_mis), Offsides (Off_mis), Crosses (Crs_mis), Ball recoveries (Recov_mis), Aerial duels won (Won_mis), Aerial duels lost (Lost_mis), Aerial duel win percentage (%) (Won%_mis).

The results.csv file contains a list of players with all the above parameters, helping to analyze or use for deeper reporting on players in the *Premier League*.

2. Problem 2:

A, Problem-2a:

- Approach: *problem-2a* has the task of processing data from the results.csv file (results of problem-1) to find *the top three players and the bottom three players* for *each statistical indicator* in the data. The implementation process includes:
  - *Reading the data:* The CSV file containing information about Premier League players is loaded into a DataFrame.
  - *Data processing:*
    - Remove columns containing non-numeric information (Player, Nation, Team, Pos).
    - Convert N/a data to 0.0 to ensure calculations don't produce errors.
    - Convert the values of numeric columns to numeric data types (int64, float64).
  - *Sort and select players:*
    - Sort each statistical column by value from low to high.
    - Take *the three players with the lowest values* and *the three players with the highest values* for each indicator.
  - *Write results to file:* Save the list of players with *top & bottom 3* for each indicator to the Top_and_Bottom_3_statistics.txt file.
- Results: After running the program, the Top_and_Bottom_3_statistics.txt file will contain a list of *best and worst players* for each statistical indicator in the data, including:

- Number of goals, number of assists.
- Defensive performance, passing.
- Advanced metrics such as *xG (expected goals)*, *CS% (clean sheet percentage for goalkeepers)*, *Tkl_def (number of tackles)*, etc.
- Special indicators for *passing*, *saves*, *dribbling* and many other factors.

This file helps easily *analyze strengths & weaknesses* of each player in the league, supporting statistics, evaluation, or making more in-depth assessments.

B, Problem-2b:

- Approach: *problem-2b* is designed to *calculate important statistics* for each team in the *Premier League*. It processes data from the results.csv file (results of problem-1) to determine:
  - *Mean value* -- The average value of each indicator within the team.
  - *Median value* -- The midpoint of the data, helping to eliminate the influence of outliers.
  - *Standard Deviation* -- Measuring the variability of each indicator within the team.

The key steps include:

- *Reading data* from the results.csv file (results of problem-1).
- *Removing unnecessary columns*: Player, Nation, Age, Pos, keeping only data by team.
- *Processing data*: Converting N/a values to 0.0 and forcing data types to ensure calculations don't encounter errors.
- *Calculating statistics* for each team.
- *Saving results*: The calculated data is saved to the results.csv file.
- Results: After running the program, the results.csv file will contain:
  - List of teams in the *Premier League*.

- Mean, median, and standard deviation of each indicator (e.g., goals, assists, shot accuracy percentage, number of tackles, etc.).
- An overview of the stability or performance disparity between teams.

This file is used to *analyze* and *compare the strength* of teams, as well as identify which teams play consistently and which teams have large performance fluctuations.

C, Problem-2c:

- Approach: *problem-2c* has the task of *creating histograms* to visualize *attacking* and *defensive* indicators of players in the *Premier League*. Specifically, it performs the following steps:
  - *Reading data* from the results.csv file (results of problem-1) containing player information.
  - *Data processing:*
    - Converting relevant data columns (Gls_st, Ast_st, xG_st, Tkl_def, Int_def, Blocks_def) to numeric types.
    - Filling missing values (NaN) with 0.0 to avoid errors when drawing charts.
  - *Drawing histograms:*
    - Creating *6 charts*, divided into *2 rows*, *3 columns*.
    - Each chart represents an indicator:
      - *Attacking*: Goals (Gls_st), Assists (Ast_st), Expected Goals (xG_st).
      - *Defensive*: Tackles (Tkl_def), Interceptions (Int_def), Blocks (Blocks_def).
    - Using *different colors* for each chart for easy differentiation.
    - *Adding titles*, axis labels, and grids to enhance readability.
  - *Saving results:* The chart is saved as histogram_premier_league.png for viewing.
- Results: After running the program, the result is:

- *An image containing 6 histograms*, visualizing Premier League players' performance by *attacking & defensive indicators*.
- Easy identification of the distribution of *goals*, *assists*, *expected goals*, *tackles*, *blocks*, *interceptions* among players.
- Support for *performance analysis* to find players with strong attacking play or solid defense.

D, Problem-2d:

- Approach: *problem-2d* has the task of *identifying the team with the best performance* in the season based on a set of statistical indicators. It performs the following steps:
  - *Reading data* from the results.csv file (results of problem-2b), containing average statistics for each team.
  - *Categorizing indicators:*
    - *Attack* **(Atk)**: Number of goals (Gls_st), number of assists (Ast_st), expected goals (xG_st).
    - *Defense* **(Def)**: Number of tackles (Tkl_def), number of interceptions (Int_def), number of blocks (Blocks_def).
    - *Possession* **(Pos)**: Pass success rate (Cmp_pas), number of progressive passes (PrgP_pas), number of touches (Touches_pos).
  - *Data normalization*: Converting the value of each indicator to the same scale (from 0 to 1) for easier comparison.
  - *Calculating composite scores:*
    - *Attack* score accounts for **40%** of total score.
    - *Defense* score accounts for **30%** of total score.
    - *Possession* score accounts for **30%** of total score.
  - *Ranking teams* by composite score to determine *the team with the best performance in the season*.
  - *Saving results* to the Team_stat_leaders_and_record.txt file.
- Results: After running the program, the results include:

- *List of teams* ranked by composite score from highest to lowest.
- An overview of the *best performing teams*, based on *attacking performance*, *defense*, *and ball possession*.
- The team with the *best performance* in the season, clearly displayed with their composite score.

This file helps *compare strength between teams*, find the team with the strongest attacking style, the most solid defense, or the team with the best match control.

3. Problem 3:

A, Problem-3a:

- Approach: *problem-3a* uses the *Elbow Method* to find the optimal number of clusters (K) in *K-Means clustering*. Its main objectives are:
  - *Reading data* from the results.csv file (results of problem-1), which contains statistical information about players.
  - *Data preprocessing:*
    - Selecting columns containing numerical values.
    - Normalizing data using *StandardScaler* to ensure features have the same scale, helping the clustering algorithm work better.
  - *K-Means clustering:*
    - Running K-Means with number of clusters from *1 to 10*.
    - Recording *intra-group variability* (Inertia) for each cluster number K.
  - *Drawing the Elbow chart:*
    - X-axis: Number of clusters (K).
    - Y-axis: Value of *intra-group variability* (Inertia).
    - Finding the Elbow point, where variability begins to decrease more slowly (this is the optimal K value).
  - *Saving the chart* as Find_the_optimal_k.png.

- Results: After running the program, the results include:
    - *An Elbow chart*, helping to determine the most suitable number of clusters in player data analysis.
    - Results supporting the selection of *optimal K*, helping classify players based on statistical characteristics.
    - A scientific approach to grouping players with similar playing styles.
- Comments on results: Based on the *Elbow Method* chart from Find_the_optimal_k.png, the Elbow point appears at K **= 3**, meaning this method suggests that *3 groups* is the optimal number of clusters to classify players. This is the level where intra-group variability (inertia) decreases sharply, but then the rate of decrease slows down as the number of clusters increases. The significance of 3 groups may be *outstanding attacking players* with notable goal-scoring and assist abilities, *strong defensive players* possessing good tackling, interception, and blocking skills, and *good ball control players* with high pass rates and effective ball retention.

B, Problem-3b:

- Approach: *problem-3b* uses *KMeans clustering* combined with *Principal Component Analysis* (PCA) to classify players based on their statistics. The main objective is:
    - *Data processing:*
        - Reading data from the results.csv file (results of problem-1), filtering out columns containing player statistics.
        - Normalizing data using StandardScaler to ensure all indicators have the same scale.
        - Using PCA to reduce the dimensionality of the data to *2 principal components*, making visualization easier.
    - *Clustering with KMeans:*
        - Identifying *3 player groups* (according to results from the Elbow method).

        - Assigning each player to a group based on their statistical characteristics.
    - *Drawing the clustering chart:*
        - Displaying player data on the axis system *PCA Component 1 & Component 2*.
        - Using colors (viridis color palette) to distinguish groups.
        - Marking *centroids* in red to see the average position of each group.
    - *Saving the chart* as KMeans_clustering.png.
- Results: After running the program, the results include:
    - *A 2D chart* showing player clustering based on their statistical data.
    - *Three clear groups*, which may represent:
        - Group 1: Attacking players (good shooting, assisting).
        - Group 2: Defensive players (good tackling, blocking).
        - Group 3: Ball control players (good passing and ball retention).
    - Easy identification of *common characteristics* of each group, helping analyze and optimize tactics.

4. Problem 4:

A, Problem-4a:

- Approach: *problem-4a* performs *data collection on transfer values* of *Premier League* players from the website *FootballTransfers.com* using *web scraping* with Selenium. The main objectives are:
    - *Reading data* from the results.csv file (results of problem-1), getting a list of players including: Player, Pos, Team, Age, and Min > 900.
    - *Setting up Selenium browser:*

        - Configuring *Chrome* browser to run *headless* (no window display).
        - *Automatically navigating* to the FootballTransfers website.
        - *Disabling notifications*, avoiding SSL certificate errors.
    - *Interacting with the website:*
        - *Closing notification pop-ups* if they appear.
        - *Moving the mouse to* "Players" section.
        - *Clicking on* "All Premier League Players" to access the player list page.
    - *Collecting data:*
        - Browsing through each player list page.
        - *Getting information* about Skill, Pot, and transfer value (ETV).
        - Saving data to the all_players list.
    - *Saving results* to results.csv.
- Results: After running the program, the results include:
    - *A list of Premier League players* with information:
        - Player, Pos, Team, Age, Min.
        - *Skill* and *Pot*, showing level of expertise.
        - *Expected Transfer Value* (ETV) from FootballTransfers.
    - *CSV file containing players meeting criteria* (>900 minutes played), useful for data analysis or machine learning models.

B, Problem-4b:

- Approach: *problem-4b* helps *build a player valuation model* based on statistical parameters. It combines data on *Min*, *Age*, *Skill*, *Pot* with actual transfer values (ETV) to predict player value.

Key steps:

- *Reading data* from results.csv (results of problem-1) containing player statistics and results.csv (results of problem-4a) containing transfer values.
- *Data processing:*
    - *Normalizing ETV* by extracting real numbers from strings.
    - *Converting Age* to numeric form (years + days/365) to increase accuracy.
    - *Separating skill and potential information* (Skill / Pot).
    - *One-hot encoding for Pos* to turn player positions into numeric features.
- *Building machine learning model:*
    - Choosing *Linear Regression* to predict player value based on input factors.
    - Applying *log transformation* (log1p(ETV)) to handle skewed distribution of transfer values.
    - Splitting the dataset into *train* **(80%)** and *test* **(20%)**.
- *Evaluating the model:*
    - Calculating *RMSE* (Root Mean Squared Error) to measure the difference between actual and predicted values.
    - Calculating R^2 *Score*, reflecting the model's goodness of fit.
- *Exporting results:* Comparing predicted values with actual values, saving to predicted_vs_actual.csv.
- Results: The results obtained after running the model will include:
    - *Predicted & actual values* of player transfer values in predicted_vs_actual.csv.
    - *RMSE ~* 0.4070 (low) shows *model has small error*, meaning player transfer value predictions are relatively accurate.
    - R^2 *Score ~* 0.7945 (high) shows the model reflects the player valuation trend well.

- Feature and model selection:
    - Feature selection: Features are selected based on *potential relationship with player value*:
        - Age: Younger players typically have higher value.
        - Min: Players who play more often are typically valued higher.
        - Skill and Pot: Higher scores indicate capability.
        - Pos: An excellent striker typically has higher value than a defender.
    - Model selection: *Linear Regression* was chosen because:
        - Player value often has a linear relationship with performance indicators.
        - Easy to interpret and suitable for small-scale data.
        - Can be improved by applying *log transformation* to handle skewed distribution.

Vũ Việt Thái – B23DCCE085