Compass Analytics Sports Analytics Hackathon User Guide

In collaboration with Databricks

Friday, October 13, 2023

Table of Contents

Context and Preamble Description *3*
File descriptions 3
Data Description 3
Data Dictionary 3
Hackathon Goal *3*
Guidance for different profiles *4*
Data Engineering 4
Data Analyst 6
Data Scientist 6
Submission Outputs *8*
Evaluation Criteria *8*
Quick Links and Reference *9*

Context and Preamble Description

You are a beginner in an NBA fantasy league where the goal is to most effectively predict the plus/minus score line of games with the most accuracy. You are competing against NAB fanastics who spend night and day watching and researching basketball. You are lucky if you watch some of the games when the playoffs come on. You have heard of the power of data science and statistical analysis to predict results so to gain a competitive edge and increase your chances of winning the league you have decided to take a data-driven approach instead of using only intuition. You have found NBA play-by-play data since 1996 to 2023 aid you in your journey. Your fantasy league predicts the score margin at the start of each quarter. You believe you can gain an edge for last quarter predictions and make up the difference in basketball knowledge. Using your analytics skills you will overcome their basketball expertise, take home the league trophy and have bragging rights forever!

File descriptions

- Play_by_play_YYYY-YY.parquet (YYYY-YY is the season eg. 2021-22)
    - Contains all the play-by-play data
- Example_game_data.xlsx
    - Example of a single game from the dataset for reference

Data Description

- Each file contains play-by-play data for the given season. Some of the data is at the play-by-play level while some other data is at the game level
- Context for each step of the hackathon process is provided in the data dictionary file below. Please review the data dictionary extensively as it will be very useful

Data Dictionary

- Data_Dictionary_Hackathon.xlsx

Hackathon Goal

The goal of the hackathon is to effectively predict the NBA Score margins for the 2022-2023 season using the historical data. The idea of the project is to get experience working with all aspects of an analysis from the point of view of a data engineer, a data analyst and a data scientist.

For the Data Engineer profile, the main goal is to take raw data stored as Parquet files in an S3 bucket and move it into Delta Live Tables within Unity Catalog in Databricks. This will give you an idea of the tools to migrate data from external locations, process the data using PySpark and generate clean data to be used for data analysis and data science.

For the Data Analyst profile, the main goal is to generate SQL queries to create datasets from the SQL warehouse from contextualized (gold-level) data. Using the data you will create a lakehouse dashboard on top of the warehouse to generate insights into player stats, team stats and season trends.

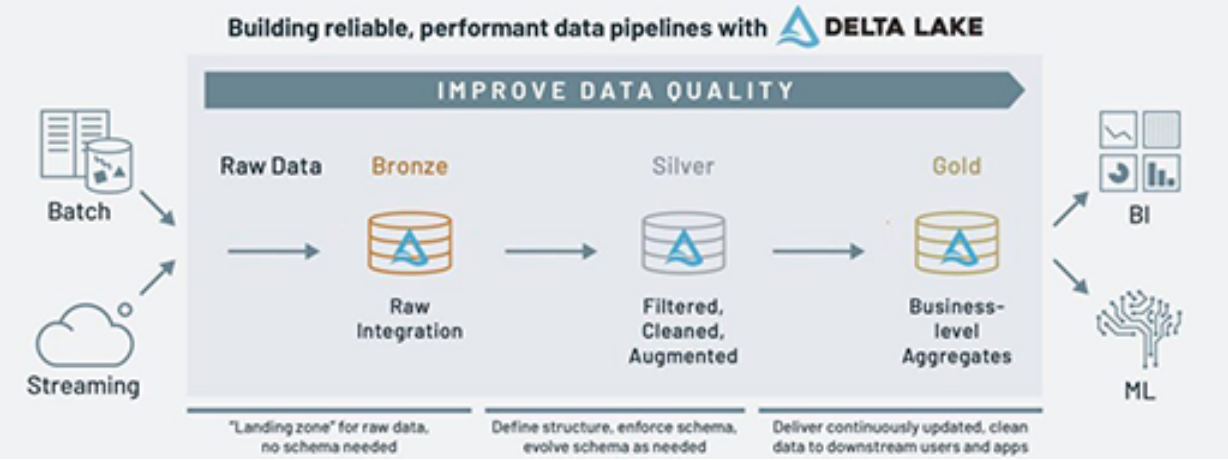
For the Data Scientist profile, the main goal is to perform exploratory data analysis on contextualized data, create resuasble datasets with feature engineering and perform various regression modelling techniques to predict the NBA Score margins.

Workspace URL: https://dbc-edd6d92b-cd2f.cloud.databricks.com

Guidance for different profiles

Data Engineering

The goal of the data engineer profile is to take the raw parquet data files in an S3 bucket (s3://databricks-compass-hackathon-main-external-location/Montreal_Hackathon_2023/NBA_Games/) and clean the data using delta live tables in Databricks. The idea is to follow the Medallion Architecture (https://www.databricks.com/glossary/medallion-architecture) which brings the data from raw data all the way to the gold layer.



The data follows the following steps:

- Bronze: Ingest raw data from source systems into delta tables with no schema required
- Silver: Perform data augmentation, ensure data quality and enforce schema of data
- Gold: Perform aggregations on data for downstream users such as in BI dashboards to gain insights

Data Engineering Tasks:

Create the following delta tables in your team's schema by replicating the tables and schema found in hackathon.example_schema. The following list of tables should be generated:

- nba_games_bronze
- Overall_game_silver
- Play_by_play_silver
- Play_by_event_gold
- Play_by_event_team_gold
- Play_by_player_gold
- Play_by_player_per_season_gold
- Team_stats_season_away_gold
- Team_stats_season_home_gold

Bronze Table:

Use the montreal_hackathon_2023 external location (s3://databricks-compass-hackathon-main-external-location/Montreal_Hackathon_2023/NBA_Games) to ingest the data from the

S3 bucket into databricks (https://docs.databricks.com/en/storage/amazon-s3.html). Use the dlt module to implement a delta live table pipeline using Python (https://docs.databricks.com/en/delta-live-tables/python-ref.html)

The data stored is in parquet format and the external location is mentioned above. Include inferSchema=True and header="true. Add meta data information specifiying a comment relating to the actions performed in the query and the quality level of the data.

Silver Tables:

Overall_game_silver:

Separate out the game level data by selecting all of the columns defined in the data dictionary in the Overall Game data. Drop all duplicates using the corresponding PySpark command

Play_by_Play_silver:

Select all of the play-by-play columns defined in the data dictionary as well as the game_date, home_team (concatenate the columns home_team_city and home_team_name together) and away_team (concatenate the columns away_team_city and away_team_name together).

Perform the following data cleaning steps to move data to silver level:

- Cast all columns to the appropriate data type following the data dictionary
- Concatenate home_description and visitor_description to create play_description
- Use regexp_replace to replace "TIE" with "0" and cast to integer
- Apply a window function to fill in the score_margin, score, away_score, home_score and team_leading (Hint: use the following stack overflow article https://stackoverflow.com/questions/36019847/pyspark-forward-fill-with-last-observation-for-a-dataframe)
- Fill in missing values for the columns generated using one of 0, "0 - 0", or "Home"
- Order results by game_id then event_num in ascending order
- Drop any duplicates

Use a DLT expectation to ensure that score_margin_complete does not contain any empty rows

Gold Tables:

Play_by_*:

Create the following 3 tables which start with Play_by_* by grouping on the following columns and aggregating to get the count. (Use groupBy, agg with F.count and F.expr, and alias to rename). The expected columns are shown in the data dictionary

- Play_by_event_gold: "season" and "event_type"
- Play_by_event_team_gold: "season", "event_type" and "home_team"
- Play_by_player_gold: "season", "event_type" and "player1_name"

Team_Stats_season*:

Create the following 2 tables which start with Team_Stats_season by grouping on the following columns and getting the count and the average of the results. (Use groupBy, agg

with F.count and F.expr, F.avg and alias to rename). The expected columns are shown in the data dictionary.

- Team_stats_season_home_gold: "Home_team_name", "Season"
- Team_stats_season_away_gold: "Away_team_name", "Season"

Data Analyst

If you were not able to complete the data engineering steps during the first half of the hackathon please use the example_schema data provided at hackathon.example_schema to create your data sources below.

Dashboard: Descriptive Statistics

Utilize the gold tables created in the data engineering steps to gain insights into player stats, team stats and number of events.

SQL Manipulation in Queries Tab:

Create a table in your schema (eg. Hackathon.your_team.event_type_lookup) called event_type_lookup from the data in the Event_Type_Lookup excel tab to update the event_type ids with descriptions in english.

SQL Manipulation in Data Tab of Lakehouse Dashboard:

Create a SQL query to get a data source called Season_Events_Counts by joining the event_type_lookup with the play_by_event_gold data. Perform the same steps to generate the table called Player_Season_Event_Counts.

Create a SQL query to get team_stats_season_home_gold and team_stats_season_away_gold into a single table. Join the data on the team name and the season. Filter out team names of teams which are not official NBA teams using a where clause

To test out the queries you can use the SQL Editor tab to query your data sources directly.

Visualizations in the Canvas Tab:

Utilizing the generated datasets using SQL create visualizations to best capture the relationships in the data. Please include examples of all types of features such as visualization, text box and filters.

Be creative and present the data in the most effective way to gain insights quickly into teams, players or number of events.

Machine Learning Results (Bonus):

Visualize the results of specific games of interest and demonstrate the effectiveness of the predictions. Find a game of interest and visualize the actual score margin during the game and the predicted score margin of the 4^{th} quarter

Data Scientist

Notebook 1: Data Exploration and Feature Store Creation

First step is to load the play_by_play_silver data into the notebook. To perform the data cleaning steps we are going to use the pandas_api for PySpark. First we want to perform some exploratory data analysis such as checking the data types, using the info command as well as checking the number of missing values. Now, remove all columns with a significant number of missing values, duplicate identifiers, uninformative/unusable columns and columns with duplicate information. This will remove a majority of the columns.

After, to get an understanding of your data set visualize the 20 games by taking the first 20 game ids from the dataset and use matplotlib to visualize

Next, we will be filtering for only plays during the regulation time to simplify the analysis as this should not impact our ability to calculate the score margin over the course of a game. Next, we want to keep only official NBA teams and convert any older team names to the most recent team name. Please refer to the data dictionary for the list of official team names and team conversions.
Check for distributions and correlations of numerical variables by selecting the integer and float columns and using the describe and corr methods.

Perform some feature engineering by creating some seasonality variables usng the game_date and season columns. Extract the month and day from the game_date by first converting the column to datetime format and getting the month and day. Next use the first half of the wraparound season (eg. 2020 if 2020-21) to get the year.

Create the column momentum_40 which will take the difference between the current score margin and the score margin 40 plays ago. Perform this within each game by creating a temporary column with the values shifted 40 plays previously then take the difference between the current margin and the shifted margin. Drop the temporary column and all rows with null values in momentum_40.

Finally, convert any included categorical variables to dummy variables and save in feature store within your schema. **The data set is very large so please filter for one of the home teams such as "Toronto_Raptors" to speed up the analysis.** Please include any other additional columns you believe will help in the analysis.

Notebook 2: Model Evaluation and Hyperparameter Tuning

In the second notebook, read in the feature store data from the created feature store and drop the unnecessary columns from the dataset such as the unique identifiers. Split the dataset into your training and testing datasets by setting all data from before 2022-2023 season or not in the 4^{th} quarter to the training set. Only keep 4^{th} quarter results as the testing set.

Use the Sci-kit Learn Library to implement a variety of regression modelling techniques such as Linear Regression and Random Forest. Please test out at least 3 different regression models which can be found here https://scikit-learn.org/stable/modules/classes.html.

To evaluate the models please use the crossvalidate function to implement 5-fold cross-validation on the training dataset and evaluate the results using r-squared, negative mean squared error and negative mean absolute error. For each model generated print out the evaluation metrics on both the training and validation sets during each iteration.

Optional: After finding the best model using the standard settings, implement hyperparameter tuning using the GridSearchCV method. Select at least two parameters to adjust in your model to find the optimal parameters. Store the best estimator for predictions.

Using the test set make predictions using the fully trained model and evaluate the results through performance metrics and creating visualizations comparing the actual 4th quarter results and the predicted results.

Save the actual results with predictions in a table within your schema.

Submission Outputs

Data Engineer:

- Notebook containing DLT (Delta Live Table Pipeline) moving data from S3 bucket through the Medallion Architecture (Bronze (Raw Data) --> Silver (Clean Data) --> Gold (Aggregated))
- Delta Live Table Pipeline with Specified Tables stored in your team schema

Data Analyst:

- Lakehouse Dashboard with visualizations of Gold Level Data for Insights
- Creative Design and Good User Experience in Canvas Tab
- SQL Queries to generate data tables in Data Tab

Data Scientist:

- Notebook(s) containing exploratory data analysis with some visualizations, appropriate numerical and categorical feature engineering, implementation of regression models using scikit-learn with proper training sets and testing sets and visualizations of actual results with predictions using seaborn or matplotlib
- Optional: Use Databricks Feature Store, perform hyperparameter tuning and creating a quick dashboard for results

Evaluation Criteria

Submissions will be evaluated on the 3 main components: Data Engineering, Data Analyst and Data Scientist Roles

Data Engineering:

- Bronze table built (ingest data from S3) - 1 point
- Silver table built (data cleaning) - 5 points
- Gold tables built (1 point for every table; total of 5 tables to be built) – up to 5 points

TOTAL DATA ENGINEERING POINTS: up to 11 points available

Data Analyst:

- Use of SQL queries to generate tables – 3 points
- Descriptive/exploratory dashboard built – 5 points
- Creative and easy-to-use dashboard filters incorporated – 1 point
- Professional and uses design best practices – 1 point
- Demonstrate ability to draw 3 insights – 1 point

TOTAL DATA ANALYST POINTS: up to 11 points available

Data Scientist:

- Lowest MAE (Mean Absolute Error) by team's ranking
  - 11 points for 1st
  - 9 points for 2nd
  - 8 points for 3rd
  - 7 points for 4th
  - 6 points for 5th
  - 5 points for 6th
  - 4 points for 7th

TOTAL DATA SCIENTIST POINTS: up to 11 points available

Bonus Points

- Use of Feature Store – 1 point
- Additional feature engineering – 1 point
- Use of external data – 1 point
- Results visualization in notebook – 1 point

TOTAL BONUS POINTS: up to 4 points available

Total available points: 37 points

Quick Links and Reference

- https://spark.apache.org/docs/latest/api/python/reference/index.html
- https://spark.apache.org/docs/latest/api/python/user_guide/pandas_on_spark/index.html
- https://pandas.pydata.org/docs/
- https://scikit-learn.org/stable/modules/classes.html#
- https://docs.databricks.com/en/data-governance/unity-catalog/index.html
- https://docs.databricks.com/en/delta-live-tables/index.html
- https://docs.databricks.com/en/dashboards/lakeview.html
- https://scikit-learn.org/stable/modules/classes.html