Adil A Kumar, Jake Arendsen, Ivy Zhou, Zihan Wang

ISOM 671: Managing Big Data

Dr. Panos Adamopoulos

16 February 2020

Using Big Data to Analyze FIFA

1. Introduction

Soccer (referred to as football by the rest of the world, because they like logical names) is by far the world’s most popular sport, with an estimated 3.5 billion fans worldwide (Das 2020). Given this popularity, it should come as no surprise that the top sports-related video game franchise, EA Sports’s FIFA franchise, are soccer games (Wikipedia). Given the freeform nature of this project, we thought it would be interesting to perform analysis on both the FIFA games and real world FIFA, to see what exactly we can learn from each independently and together. In particular, we want to attempt to cluster players based on factors from multiple datasets, attempt to reverse engineer FIFA’s overall ratings using real world statistics, and utilize big data tools to explore our data and build visualizations. As a team, we really valued the opportunity to explore the data freely and present things that appeared interesting as we came across them.

1. Data

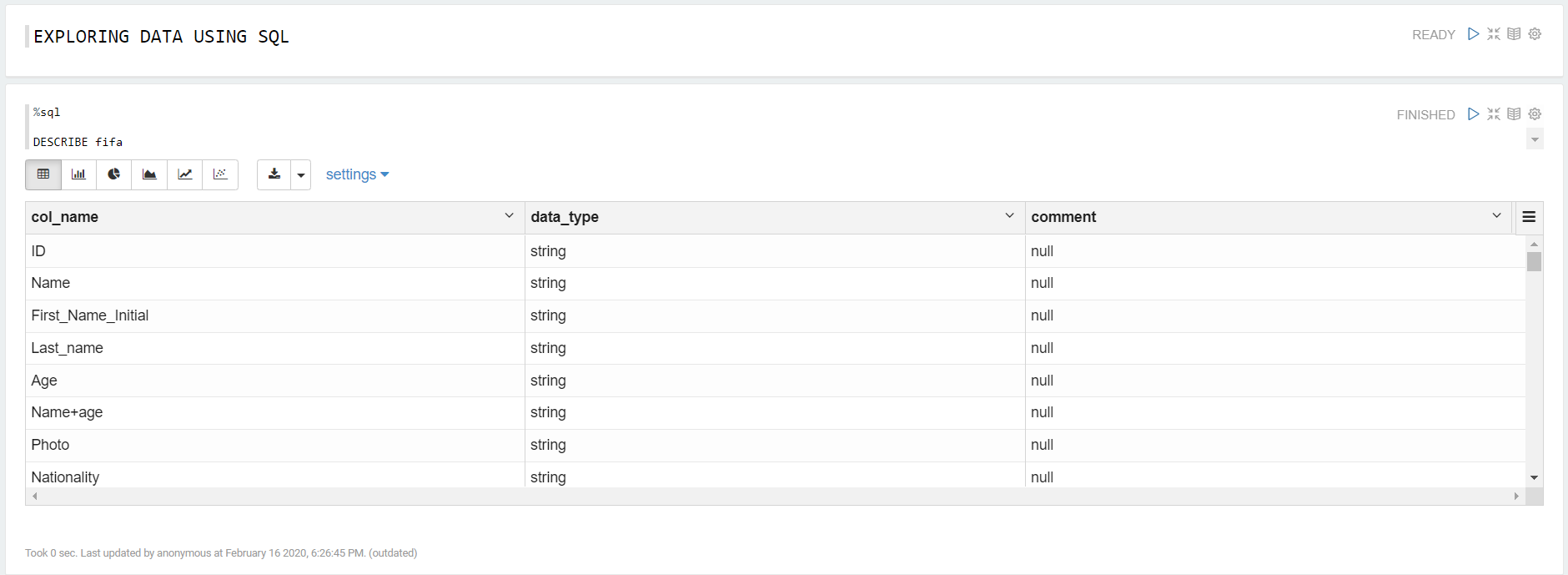
Our data comes from two distinct sources. For FIFA data, we found a dataset on Kaggle which contained information regarding a player’s statistics, abilities, and personal information within the video games. For us, particularly useful information included a player’s overall rating, which we would do much of our analysis with, and basic personal information which would be very useful for joining our dataset. Additionally, individual attribute ratings (a player’s speed, passing, shooting, etc.) could be used to help validate some of our analysis.

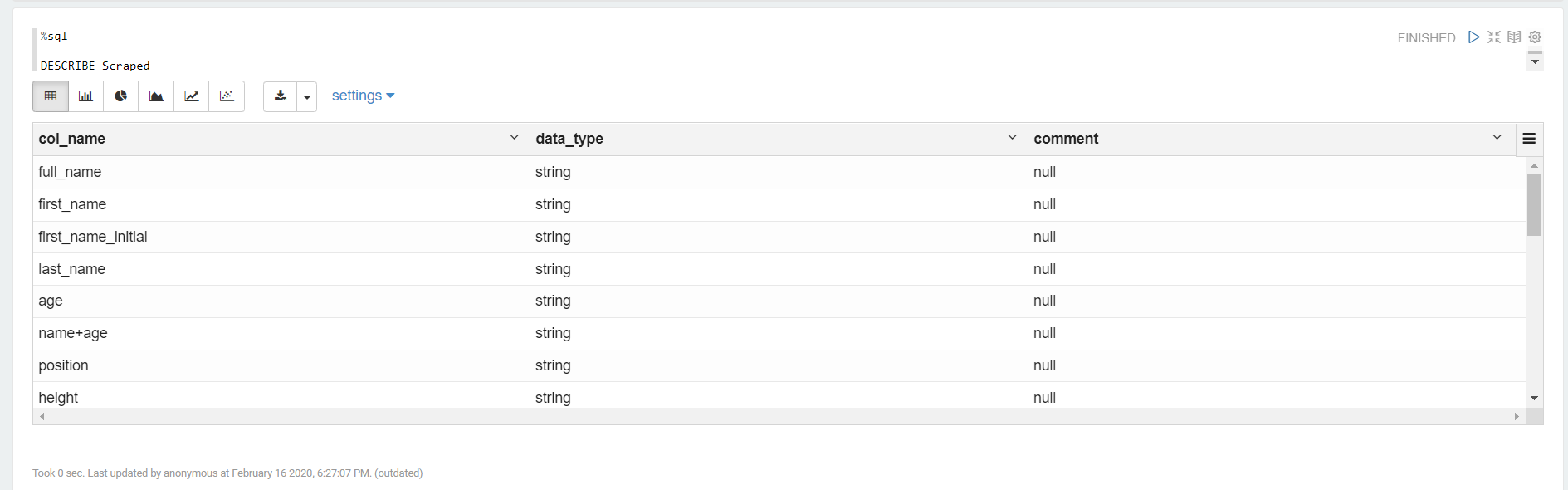
For our real world player statistics, we had to utilize FBref, the professional soccer division of Sports-Reference, a well known collection of numerous sports statistics. However, the website lacks a feature to easily pull the data yourself, so we had to manually scrape the data from the website using Python. To this end, code provided to us in Sports Analytics by TA Wooyong Jo proved incredibly useful. His code was originally intended to scrape Basketball-Reference, so it had to undergo numerous modifications to be utilized in our setting. Among other things, the biggest challenges we had to account for is that FBref is much larger, less consistent, and the standard statistics table is more complex, and thus harder to pull. We did manage to develop a program that pulled statistics for all players, thus giving us real world data. However, the program is not super fast and had some difficulty running, so we were only able to get a sample of the sample we tried to collect. As such, we believe that this is a part of the project we could have improved greatly with more time and effort, as potentially using big data tools to assist in scraping could be very beneficial.

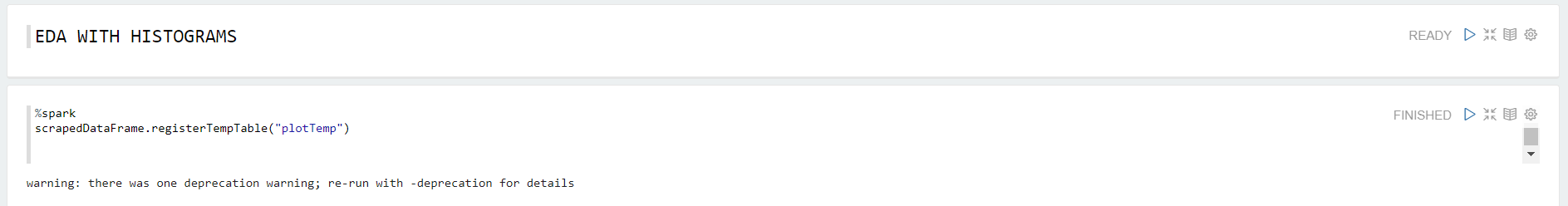
The two data sets share similar kinds of information; however, they are quite different in size. Both data include personal information, the teams, and the metrics on performance. The FIFA data has over 17,000 pieces of records each with around 90 attributes and the web scraping data has over 13,000 records each with nearly 20 attributes. Compared to the FIFA data which was already in an analyzable format, the web scraping data needed data cleansing, and a main reason is that it records multiple lines for a single player in different teams or seasons. To be more consistent with the game data which collected in 2018, duplicate records were removed to keep only one record for each player in 2018 or season 2017 - 2018. Null values were also dropped to make sure the data was ready to be analyzed.

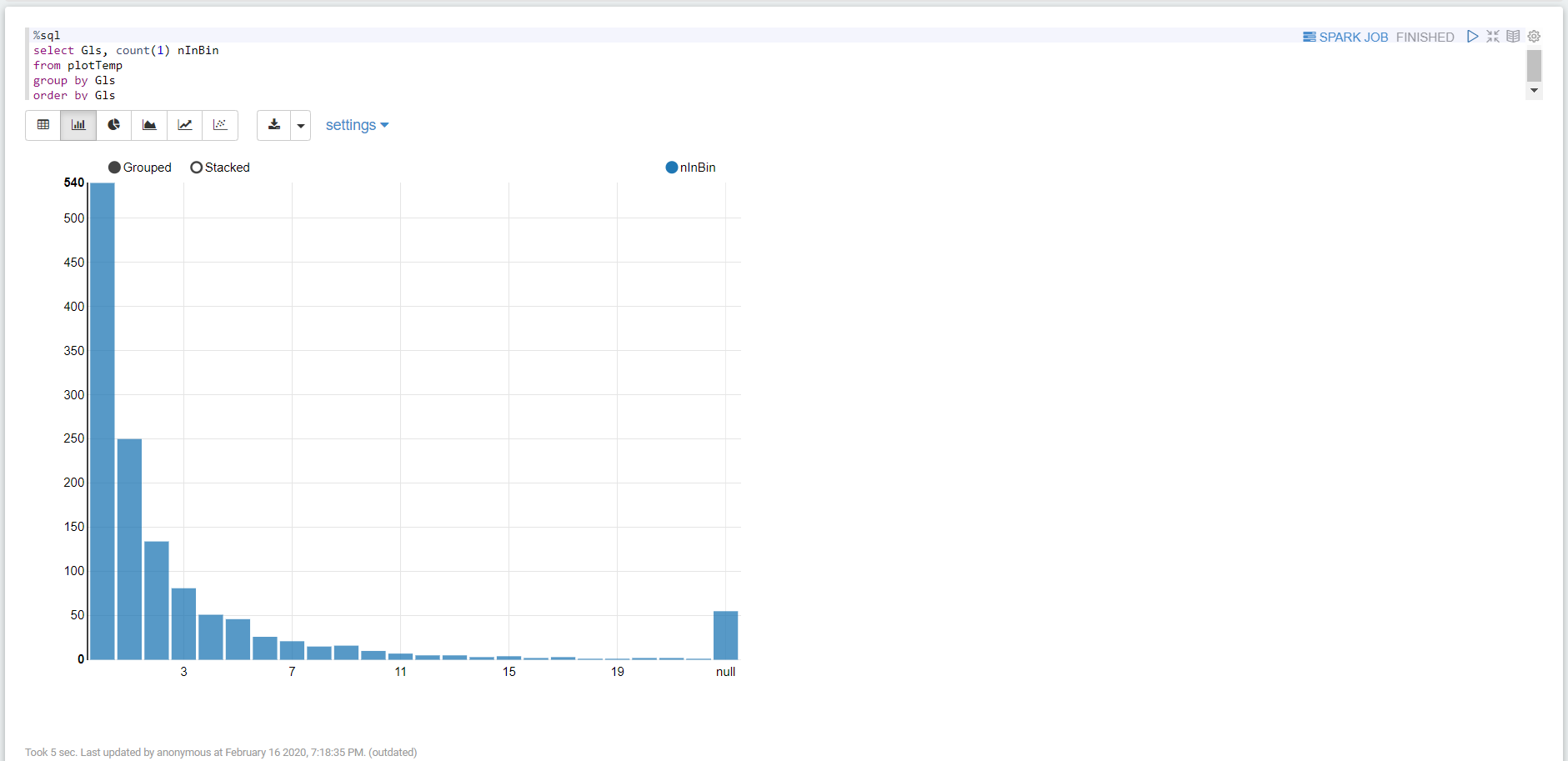
After cleaning the dataset we scraped, we joined it with the FIFA dataset. First we cleaned and filtered the dataset we scraped. Since we are focusing on FIFA 19, in which statistics are based on seasons that ended in 2018, we filtered the scraped data by season. After filtering, we found the player names are in different formats in the two dataset: players names in FIFA dataset are not full names, instead, their first names got abbreviated. Thus, we are not able to join the datasets using the name of each player. To solve this problem, we created a new key column: “Last Name + Age” to identify each player. To make sure that each player has a unique key, we checked the key column and the corresponding player and found that there is no case of duplicates. This is partly due to the size of the dataset. After cleaning, filtering and joining the dataset, we found there are 90 players left for us to proceed to the next step. Obviously, we recognize that we ideally would have performed this with more players and accept this as a shortcoming of our analysis.

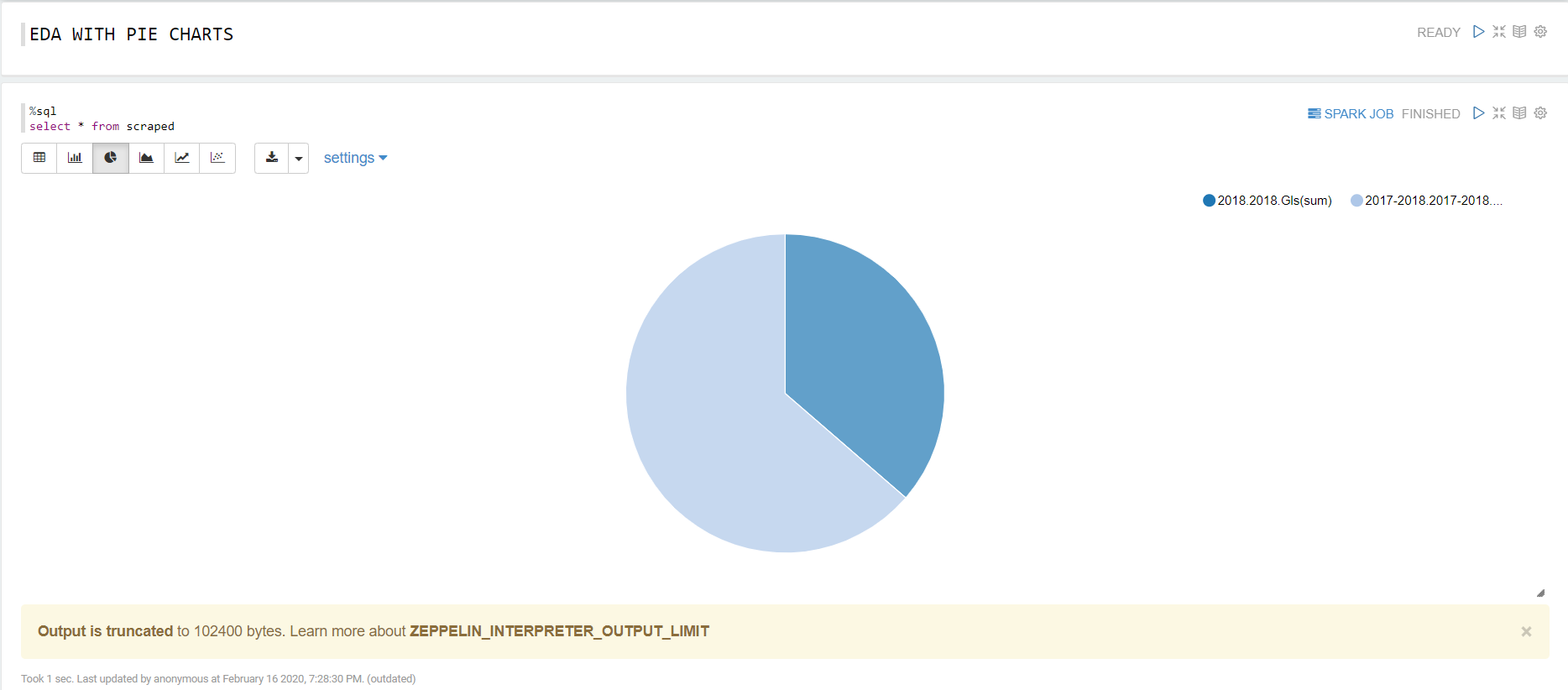
1. EDA

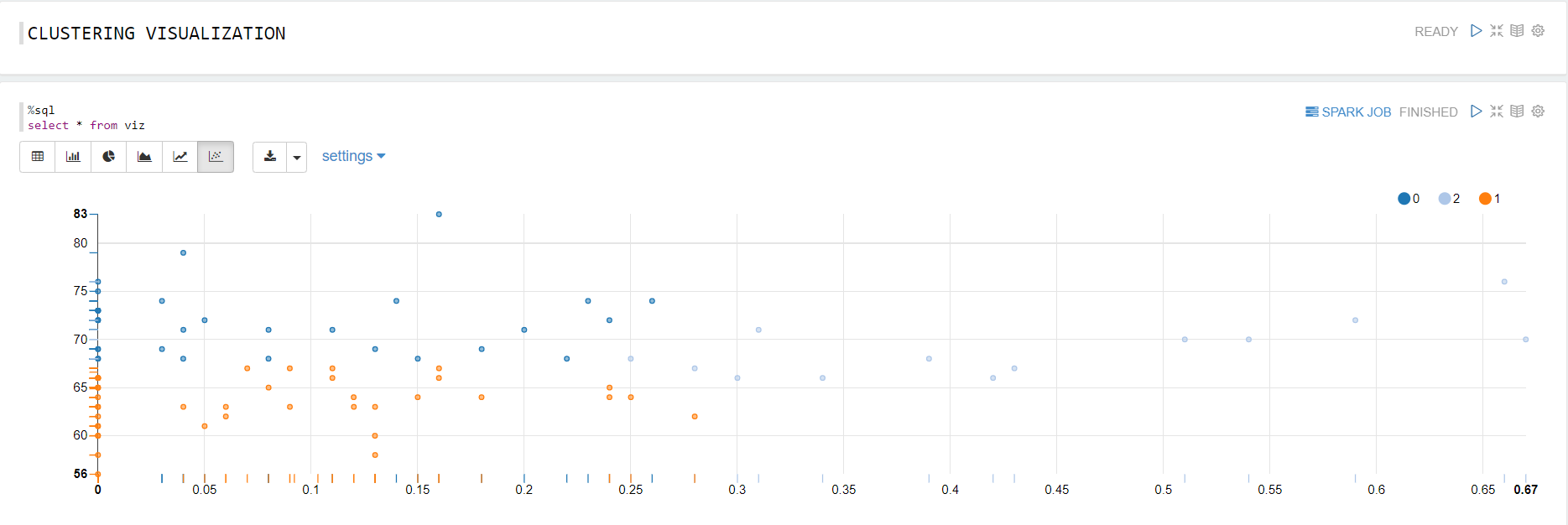












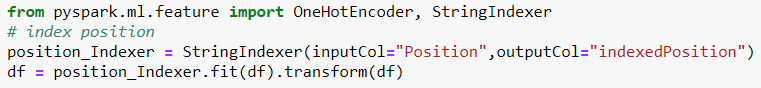
1. Analysis

After data preparation and understanding, we can proceed to data analysis. We seperated the analysis into two parts: Prediction and Clustering. In prediction analysis, we attempt to use a player’s real-world statistics to predict their FIFA rating. In clustering analysis, we attempted to cluster players based on their FIFA ratings to see if there’s any “typical” players.

1. Prediction

Each player in FIFA 19 has an overall rating as well as scores for the key stats. These stats are combined with a player's international recognition to calculate the player's overall rating. Higher rating generally corresponds to better player performance on the field. In this project, our joined data contains an overall rating by FIFA for each player who is active during the 2018 season, on a scale from 0 to 100. Well-known good players are generally rated between 85-100, while other players are rated within a range of 50-80. Apart from the FIFA overall rating, we also have real world player statistics including Position, Goal Count, Goals Per 90 Minutes, Assists, Penalty Kicks, Number of Yellow Cards, Number of Red Cards, etc. With all the information on hand, our first goal is to predict the FIFA score given a player’s real world statistics.

We used Pyspark to transform the data, build models and evaluate models. First, we transformed the categorical columns using StringIndexer.



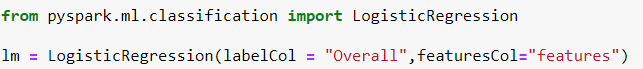
Then we assembled the predicting features: Goal Count, Goals Per 90 Minutes, Assists, Penalty Kicks, Number of Yellow Cards, Number of Red Cards,etc.



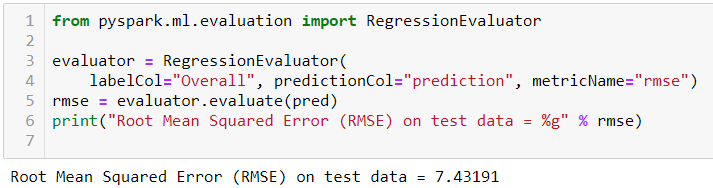
Next, we built a pipeline and tried numerous different models.



Logistic Regression Example:



We applied different models to our dataset but due to the limitation of the size, the performance of the prediction is very limited. The best model we found is the logistic regression model, which gives us the best RMSE(7.43191).

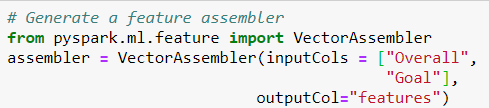


In addition, by observing the prediction and actual overall rating, we found the difference between prediction overall rating and the actual overall rating at all test cases are between 5-10.

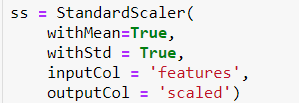
1. Clustering

Different players and teams have different playing styles in the field. A team could change styles based on what kind of players are playing. So our second goal is to cluster FIFA players based on their FIFA overall rating and their real world statistics to see if there are any “typical” players.

Similar to the process of building a predictive model, clustering also needs transformation, assembling features, building a pipeline and filling model processes. However, unlike prediction, feature scaling is necessary before clustering because we need to make each feature equally significant for distance calculation. We created a feature assembler to assemble features. Considering a good visualization of the clustering, we assembled two features: Overall rating and Goal per 90 minutes.



We then scaled the features, using mean and standard deviation when scaling.



Afterwards, we created a KMeans estimator, setting the featuresCol to “scaled” and setting 3 clusters.



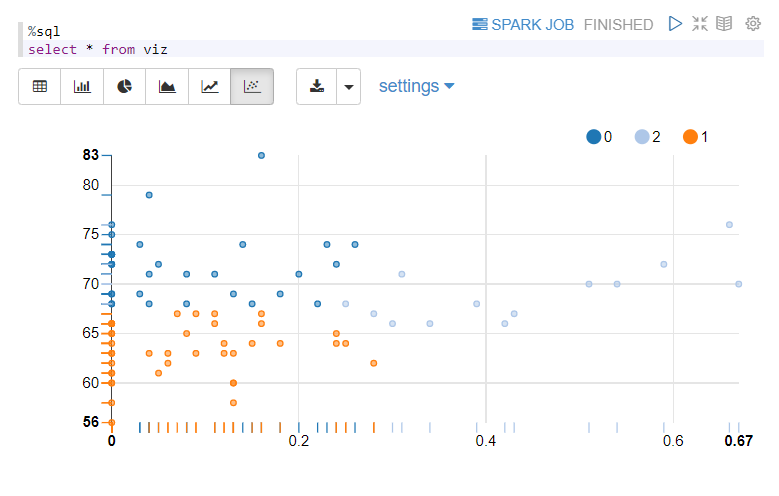
The next step was building a pipeline with a sequence of three stages: assembler, StandardScaler and the KMeans estimator. These stages are run in order, and the input DataFrame is transformed as it passes through each stage.



Following that, we built chain methods of the pipeline (and the resulting pipeline model) to create an RDD of predictions using only one line.



After clustering, we output the clustered data into a csv file and generate visualization using Zeppelin.

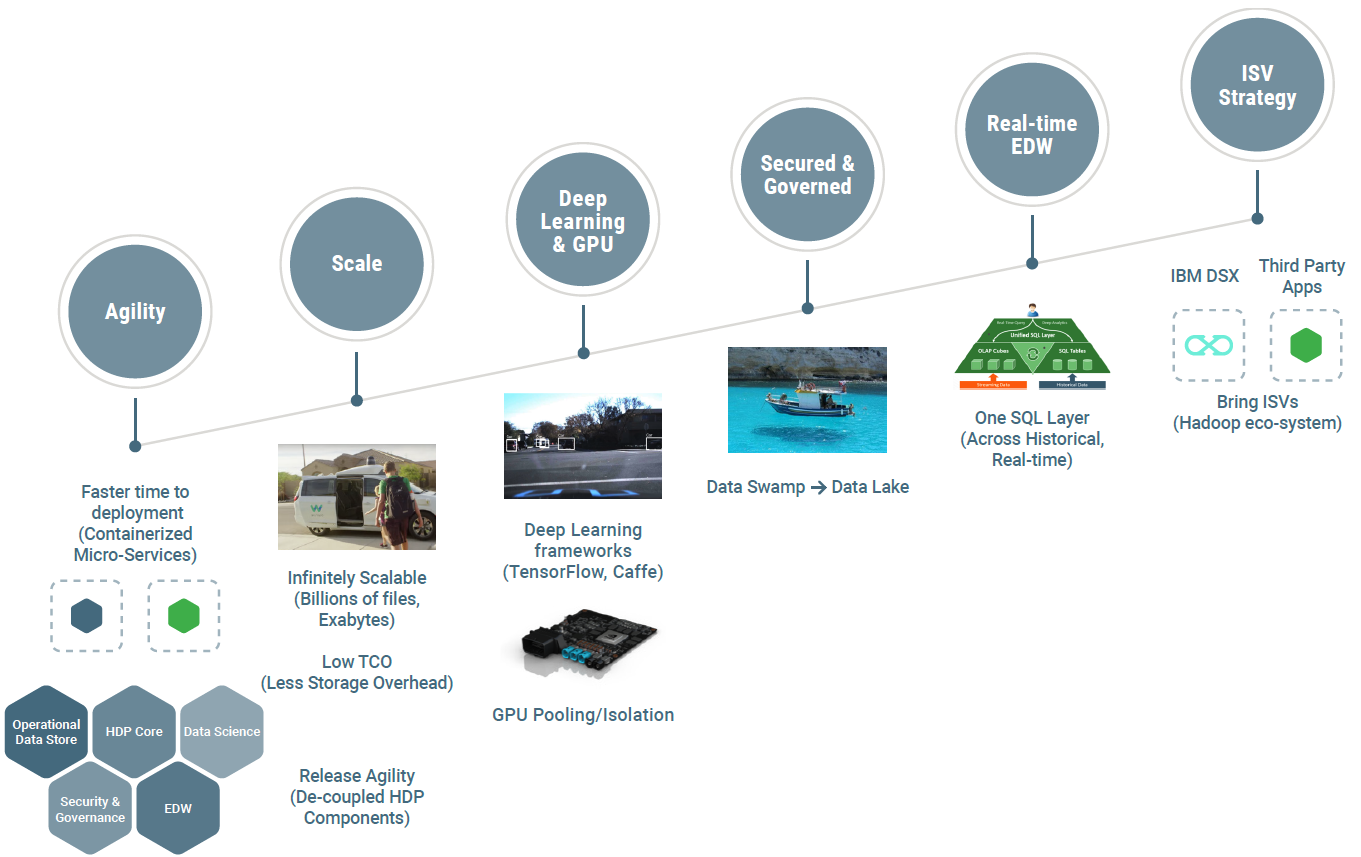


We found there are three types of “typical” players: Good Offensive Players, Good Defensive Players, and Mediocre Players.

Good Offensive Players (2) typically have a comparatively high FIFA overall rating and elite offensive statistics. They are key players on a team and the main scoring power of the team. They take the lead to attack the opponent team and thus are more active in goal attempts and assists. Good Defensive Players (0) typically have comparatively high FIFA overall rating and weak offensive statistics. They form the shield of the team, playing the role of defender against the opponent team. They typically stay closer to their own teams’ goal and do not always have a chance to offensively interact with the ball, which leads to their weak offensive statistics. Apart from good offensive players and good defensive players, there is also a group of Mediocre Players (1) with a low-average FIFA overall rating and comparatively weak offensive statistics. This group contains more players than the other two. Players in this group are not in the top tier either in FIFA overall rating or in personal offensive stats. They are likely role players or bench players for their current squads, in a role where they do not get lots of playing time and are easily replaceable. It should be noted that the clustering is obviously impacted by “standard” soccer statistics, which are very offensive in nature. Additionally, we are assuming EA Sports is good at their job and the ratings are somewhat reflective of a player’s abilities. We feel relatively confident assuming this, as we had some success reverse engineering their formula using real world statistics.

V. Big Data Platform Overview

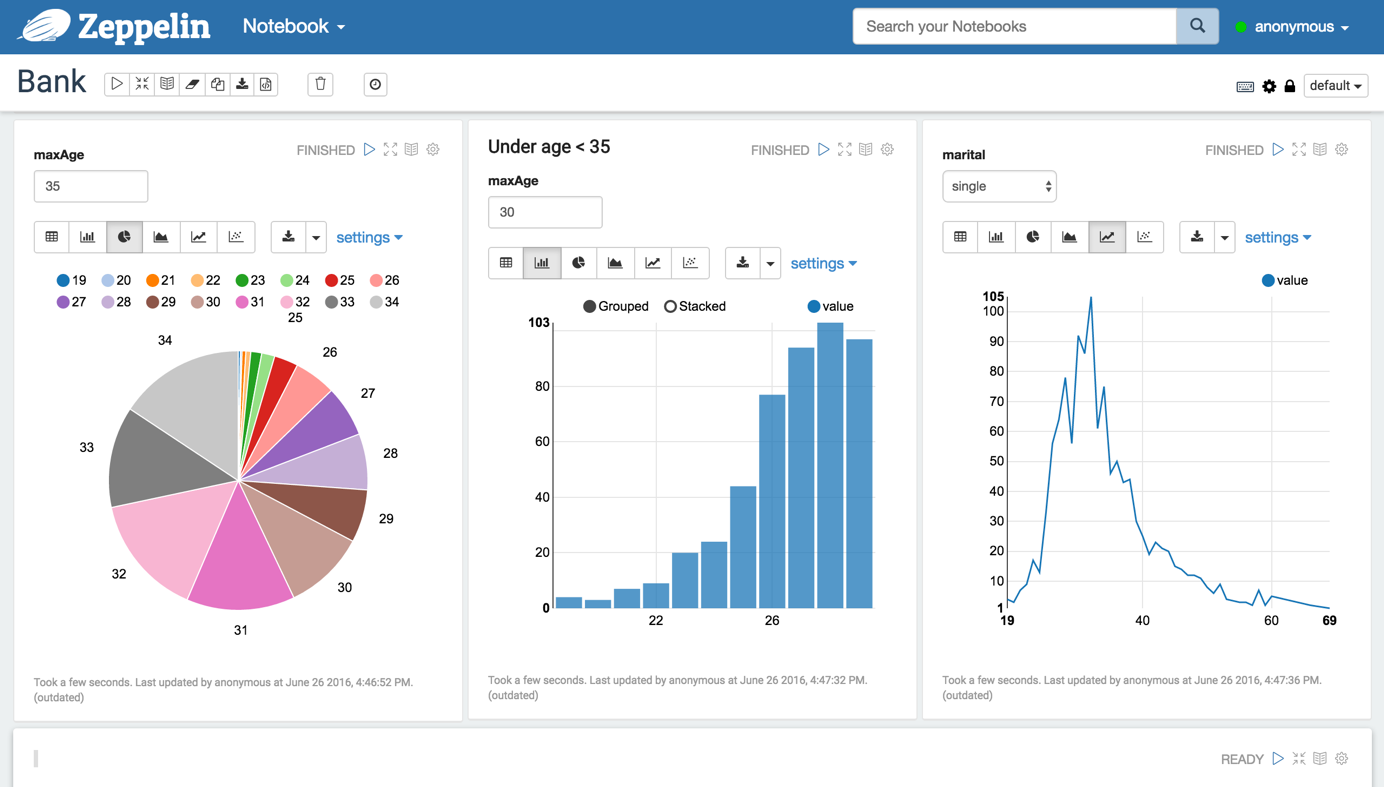
Our big data application relied on a tool from the HortonWorks Data Platform (HDP), a Hadoop distribution somewhat similar to the run we run in class, but with much more applications and tools available. It launches through one’s browser to pull u[ the virtual machine that has access to all of HDP’s big data tools. It was designed in part to help assist with the issues indicated below, in a graphic from HDP website.



Obviously, all six of the issues are relevant and key for data scientists in their processing of big data, so we thought experience with the platform could prove vital to our future. Among the key benefits of HDP are that it’s smarter, faster, and bigger than other similar tools, allowing it to outperform its competition. Additionally, implementation of key Apache tools allows HDP to provide a wide covering of other necessities. For example, Apache Hive assists with real time database pulls, Apache Spark assists with any machine learning needs, and Apache Ranger and Apache Atlas help track data and ensure security.

To help monitor the whole Apache zoo, HDP comes with Apache Ambari, a web user interface (UI) that allows a user to monitor and access the entire Hadoop ecosystem. Ambari is a great tool for monitoring our entire system, as it has tools that can assist with Hadoop installations as well as creating, stopping, or reconfiguring Hadoop services across clusters. Ambari provides a dashboard to monitor performance across clusters and can be set up to provide alerts when problems are detected, such as disk space running low or a node failing. For the purposes of our project, Ambari served mostly as a portal to access Apache Zeppelin, which was the real star of the show.

Apache Zeppelin is a Web-based notebook that enables data-driven, interactive data analytics and collaborative documents with SQL, Scala and more. Zeppelin has features for data ingestion, data discovery, data analytics, and data visualization, making it a great place to perform exploratory data analysis (EDA) on big data. Because it’s incorporated with HDP, it’s a great place to do some intermediary exploration and examine some features for feature engineering before exporting your data from the Hadoop ecosystem. In our project, we used it to handle all of our EDA needs and create visualizations without ever exporting our data. Finally, the Apache Zeppelin interpreter concept allows any language/data-processing-backend to be plugged into Zeppelin. This hybrid functionality allows Zeppelin to be a useful tool for people of all backgrounds, as it likely already supports any language a user would potentially know. Currently Apache Zeppelin supports many interpreters such as Apache Spark, Python, JDBC, Markdown and Shell. Below is a screenshot of an Apache Zeppelin notebook, for reference.



VI. Concluding Remarks

In terms of our original goals, we found that we could cluster soccer players into three fairly distinct individual clusters related to performance and, even with incredibly simple statistics available to us, could do a decent job of predicting a player’s FIFA rating. Moreover, we appreciated the opportunity to use big data tools to freely explore data and draw conclusions and insights we find interesting. Getting exposure to higher end big data tools only makes the entire Hadoop zoo seem more interesting and exciting, and served a role in invigorating our passions to learn and explore as the semester continues along. Should one want to try and enhance our project, the first step would obviously be to improve the web scraping process and the data pulled from it. Harnessing a more powerful device through AWS would be a great first step, but even then the scripts will have to be reworked some to properly pull data and ease the join to allow more data to be kept. While we obviously noted some shortcomings within the report, we still found the process to be exciting and valuable.

Bibliography

Ambari -. (n.d.). Retrieved February 16, 2020, from https://ambari.apache.org/

Cloudera, Inc. (2019, September 5). Hortonworks Data Platform Datasheet. Retrieved February 16, 2020, from https://www.cloudera.com/content/dam/www/marketing/resources/datasheets/hdp-datasheet.pdf.landing.html

Das, S. (2020, January 21). Top 10 Most Popular Sports in The World [Updated 2020]. Retrieved from https://sportsshow.net/top-10-most-popular-sports-in-the-world/

Jo, W. (n.d.). Atlanta Thunder Case Data Collection Code.ipynb

Football Statistics and History. (n.d.). Retrieved February 16, 2020, from https://fbref.com/

Foundation, T. A. S. (n.d.). Apache Zeppelin. Retrieved February 16, 2020, from https://zeppelin.apache.org/

Gadiya, K. (2018, December 21). FIFA 19 complete player dataset. Retrieved February 16, 2020, from https://www.kaggle.com/karangadiya/fifa19

List of best-selling video game franchises. (2020, February 16). Retrieved February 16, 2020, from https://en.wikipedia.org/wiki/List\_of\_best-selling\_video\_game\_franchises