PROJECT REPORT

Statistical Representation of UEFA Champions League Data using Web GIS

Prepared By: Muhammad Mohid (160601005)

Submitted To: Madam Aneeqa Abrar

Table of Contents

Abstract:

The UEFA Champions League has a rich history in the footballing world. Recognized worldwide as the premium football competition where the best players in the world compete for the chance to lift the coveted long-eared silver trophy. Its' long and rich history is filled with the achievements, accomplishments, records and statistics of some of the greatest players to have ever played the sport. This project aims to represent and highlight those very records and achievements using modern Web GIS techniques. The result of this project is to create dynamic maps that can tell a story of the rise and fall of different teams throughout a specified time span of the competition. Some of the popular statistics such as goals scored and assists provided are represented statistically in the project in an eye-catching way. The project comprises of five different maps (one for each selected year of the competition). Each map represents the winner of the UCL (UEFA Champions League) for that season and the seasonal statistics are presented below. A basic User Interface has been designed to ensure proper usability and to switch between maps. The maps were developed using OpenLayers Web Mapping Library and Java Script. The GUI was developed using HTML, CSS and Bootstrap.

Tools/ Software Used:

The following software/ tools were used to perform the project. Each software/tool has been mentioned along with its' subsequent purpose in the project's methodology.

- OpenLayers: OpenLayers is a dynamic web mapping library, that has a very high reputation among developers and students. The web maps for the project were created using OpenLayers along with functionalities.
- Java Script: Web mapping and Java Script go hand in hand. All major functionalities, layer styling and features were created using JS with OpenLayers.
- HTML: Hyper Text Markup Language is the most popular language out there for developing web pages. The layout for the web page or GUI for this project was created using HTML.

- CSS: Cascading Style Sheet or CSS was used along with HTML for the styling and beautification of the GUI.
- Bootstrap: Bootstrap was used to add responsiveness and easy to handle styling features to the GUI.
- Arc Map: Arc Map was used to merge the datasets of European countries and also to serve as an organized database for the datasets in the layers.
- QGIS: QGIS was used to convert the shapefile layer to GeoJSON layer.

Methodology:

The project's methodology though relatively simple went through a lot of revision and rethinking. Many problems and difficulties were encountered in the implementation and subsequently, the project's goals and objectives had to be redefined. In this section, however, only the methodology and implementation of the project is discussed. The difficulties and problems faced are discussed in a later section. The changes made to the project, however, are discussed, in this section relative to those described in the proposal for the project.

Changes and Alterations:

As mentioned earlier, the project had to undergo some changes from the proposed methodology. The changes are discussed as follows (Note that the reasons for some changes are discussed in the problems section):

Proposed Methodology	Changes to Methodology
It was proposed initially to use Geoserver for the styling, storage and hosting of the data layers.	GeoJSON was used instead of Geoserver for the styling and representation of layers on the map. This change was made purely for academic reasons as the developer had prior experience with Geoserver but no experience with GeoJSON. This project was to be a learning experience for the developer in the usage of GeoJSON

Dot Proportional maps would be used to represent the statistics (Goals, Assists etc.)	This was changed to representing icons on the maps and the statistics are represented separately.
Statistics of a nine-year period (from 2010-11 2018-19) would be represented.	This was changed to a period of five years (from 2010-11 to 2014-15).
Proposed map (See Figure 1)	Final Map (See Figure 1)

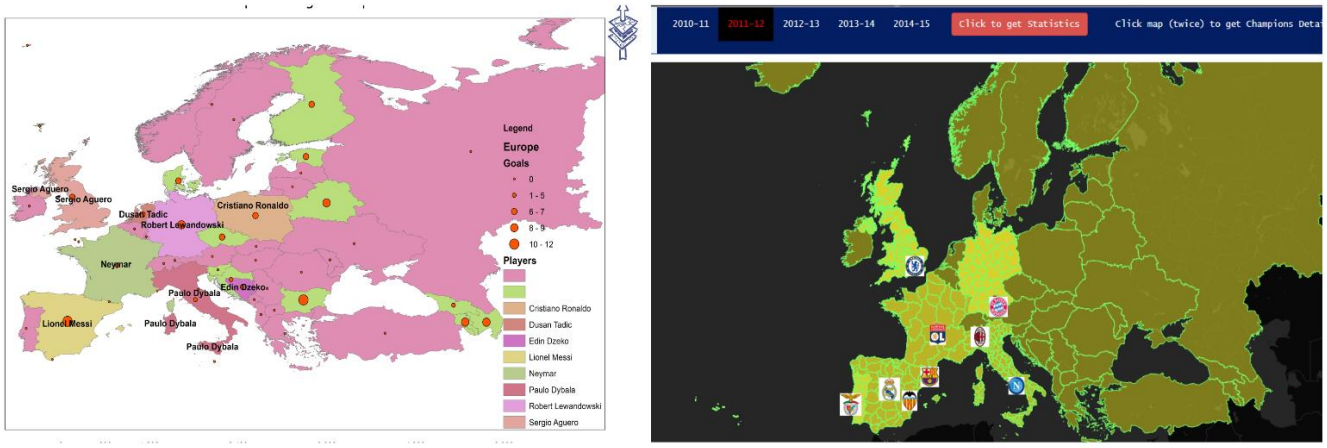

Figure 1: Proposed and Final Map Comparison

Steps Followed for Project Implementation:

As the changes made to the project have been discussed, here we will discuss how those changes were brought about and how the map/s were created systematically.

- The first step was to obtain the datasets that were to be represented on the maps. For this, a shapefile of Europe was downloaded (see References). Along with a shapefile of Europe, separate shapefiles of Spain, UK, Germany, France, Italy and Portugal were also downloaded. These countries were downloaded separately because the Europe shapefile did not contain data for the cities of each country. The country shapefiles or layers were

then merged using Arc Map with the Europe shapefile to create a new layer with complete city and district dataset for major European countries.

- The next step was to add the statistical data to the shapefile layer. The statistical data included all major clubs participating in the UCL, top 5-10 goal scorers, top 5-10 assisters, number of goals scored and number of assists provided. The clubs were not selected based on season ranking, rather they were selected on the basis of the player/s representing the club in the respective season in terms of goals or assists. For Example FC Barcelona would be represented only if a player from the club was in the top goals or assists statistics.
- After the datasets for each season were stored in the layer, it was then converted from a shapefile to a GeoJSON layer using QGIS.
- The next step was to create a web map using OpenLayers. As the data was to be represented for different seasons, different base maps were selected. For 2010-11 a generic OSM base map, For 2011-12, a carto dark base map was selected. For 2013, a stamen water color base map was selected and for 2014 and 2015, an ArcGIS World topographic map was selected.
- After the selection of the base maps was complete, the next step was to create a styling function for the GeoJSON layer and add it to the map. The styles created were of two types: the first style was to create distinguishing colors for the features in the layers. The clubs representing each city were given different colors. A base color fort the rest of the GeoJSON layer was selected as well. The second type of style was to create icons on the maps. Each icon would represent the participating clubs for that season. Clicking anywhere on the layer or any icon would open a popup representing the season's champions with a small summary.
- The next step was to add the season's statistics to the map. For this, pie charts and co-relation charts were created using Google Charts. The statistics for top scorers, top assisters etc. were added and different charts were created.
- The final step was to create a GUI for the web maps. For this, a simple navbar was created using HTML and Bootstrap. The navbar allows the user to switch between maps. The

height, width and placement styling for the navbar and the web maps was done using CSS.

Results:

The product of the above-discussed methodology and its' implementation is:

- Five dynamic web maps with complete statistical representations for each of the five seasons from 2010-11 to 2014-15.
- A user interface that allows some basic interaction with the maps such as viewing statistics and map switching.
- Complete statistics of all five seasons of the UCL in the form of pie charts, donut charts, bar charts and co-relational charts.

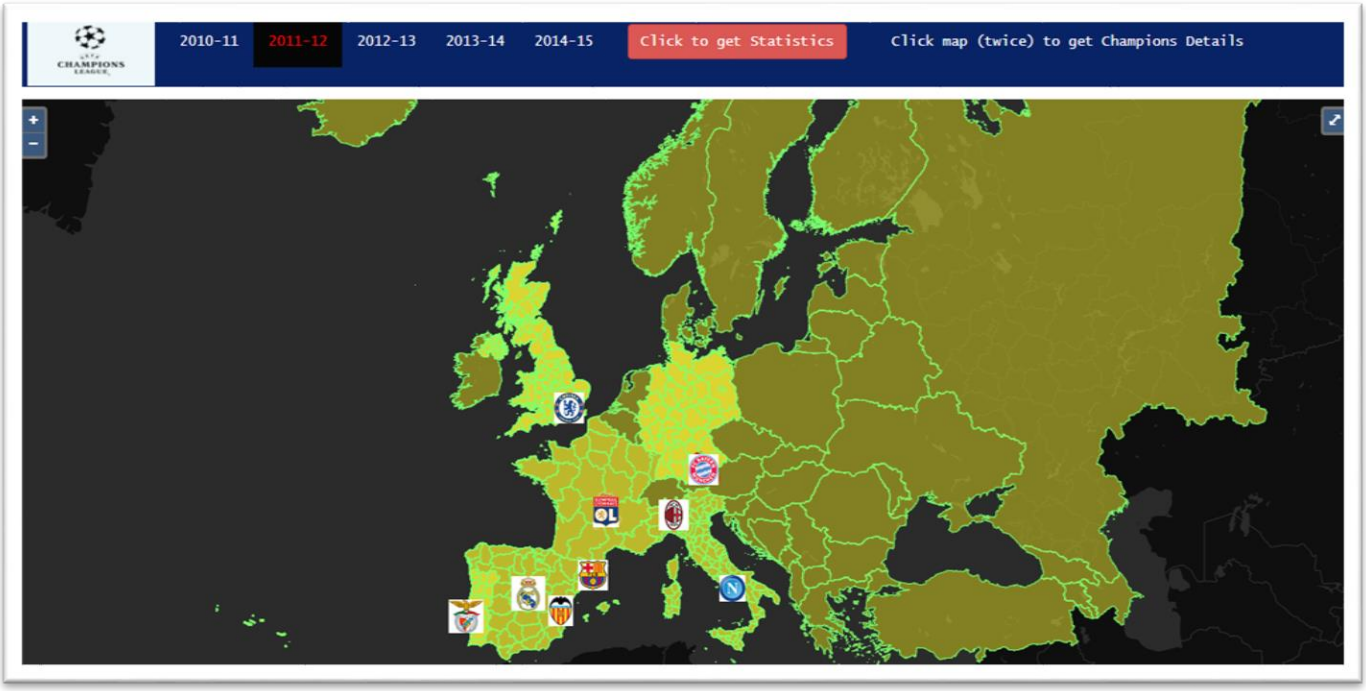

Figure 2: Web Map for 2011-12 season (example)

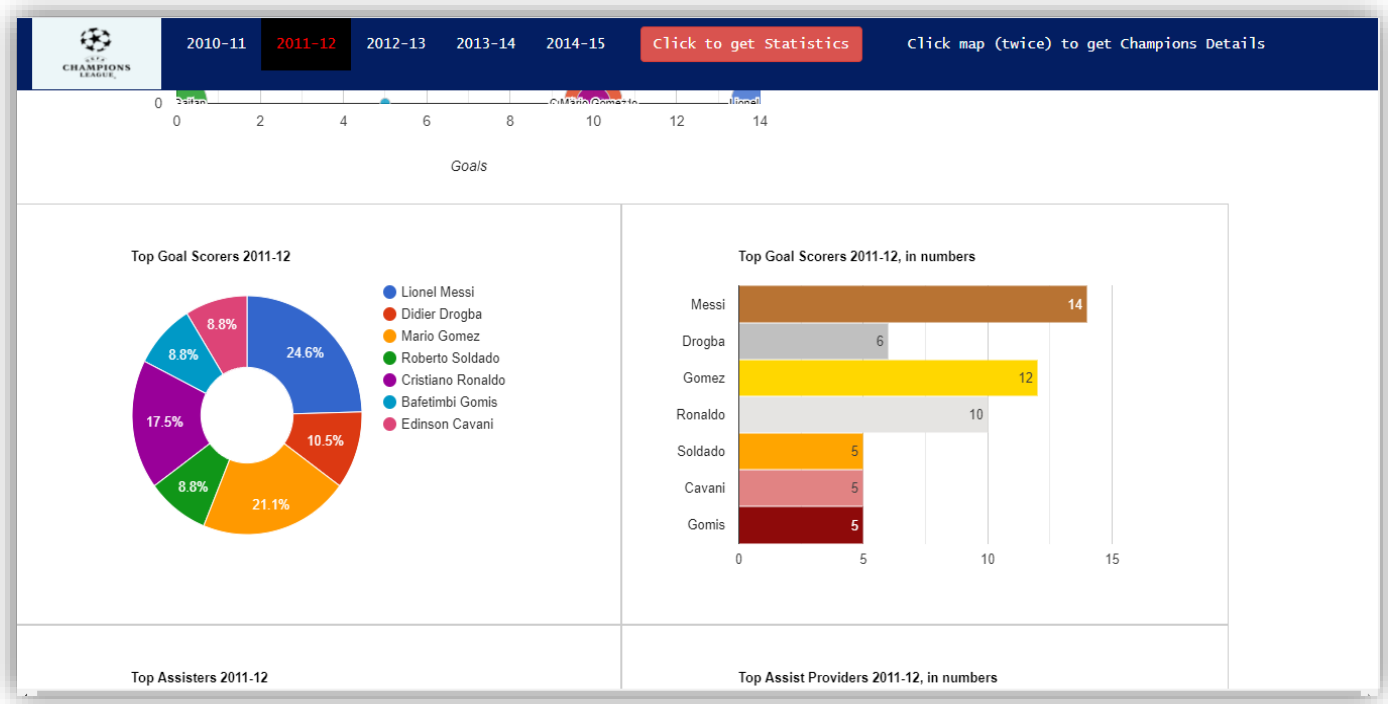

Figure 3: Statistics for 2011-12 season

Problems and Recommendations:

A number of problems and issues were encountered which resulted in many changes that were made to the project as discussed previously. Some of the problems encountered and their subsequent workarounds are discussed here.

1. One of the most significant problem was that of the mapping library OpenLayers itself. In OpenLayers 6 markers, popups and icons are deprecated. This means that a user can manually create icons or graphics on the maps, but the library does not support functionalities like marker hover or popup for GeoJSON. In fact, for GeoJSON data, *hover on features to display information* can be performed easily. This functionality is suitable only for choropleths. Therefore, if a user creates multiple markers with different popups, only one popup will appear for every marker.

2. Another major issue with OpenLayers 6 is that it is suitable for production purposes only. This means that for a user who wants to create a small time project, the functionalities are very limited. OpenLayers encourages users to learn NodeJS for proper functionalities and usage. Therefore, to use OpenLayers properly, a user cannot simply attach the CSS and JS files in the html page and start working. A user has to learn NodeJS first. This is obviously a time consuming endeavor for someone who is not familiar with NodeJS and wants to develop a small project. This effectively means that all the documentation of OpenLayers 6 (which is for NodeJS users only) is useless for small time developers.
3. Initially, dot proportional maps were proposed as the main form of data representation for the project. However, there were two major issues. Firstly, the number of maps needed to be created would be too large. A single map would represent one variable, say Top Goal Scorer for one season. That would mean that three maps per season would need to be created i.e. top goal scorer, top assister and top football clubs. That would mean 27 maps for 9 years. Even with the revised number of years brought to five the maps would number 15, still too many. Too many maps of the same kind would become extremely boring and repetitive. The second issue was that OpenLayers documentation for dot proportional maps can be found for OpenLayers 2 or below. OpenLayers 2 is of course deprecated and is out of use and out of favor by all developers.

All these problems do have their workarounds, but to solve them properly requires a long period. A user must first learn and master NodeJS. Then the user would have to learn the thematic web mapping techniques for OpenLayers from a teacher because the OpenLayers' documentation does not cover this topic apart from choropleths. The user would also have to learn how to use the OpenLayers mapping extensions (for charts etc.) which are used for making thematic web maps, especially dot proportional or statistical map like bar charts maps etc.

In short, OpenLayers may have many interesting and useful functionalities, but they are not suitable for beginners or users with little experience. As a user who has worked with both OpenLayers and Leaflet, I would easily recommend Leaflet, as it is very user friendly, has ample and easy to understand documentation and does not actively require NodeJS to function.

References:

Download Shapefile of Europe Free

Download Shapefiles of Countries Free

UEFA Champions League Season Statistics

Create dynamic charts with Google Charts API