190003657

Data Intensive Systems
CS5052

Practical 1

24/03/2023

Apache Spark

University of St Andrews

Contents

1 Introduction

1.1 Overview

The goal of this practical was to gain experience working with Apache Spark in order to analyse a large dataset. This was done in python using the PySpark library to operate on the UK Government's dataset on pupil absences between 2006 and 2018.

In this practical, a console application was created that implemented the core and intermediate features defined in parts 1 and 2 of the specification. For part 3, a web app was created to provide a better interface, allowing visualisation and interaction of the dataset with features like plots and charts.

1.2 Design

A tree structure is used to restrict the path of features that a user chooses to view to just what is available for each part. This simplifies implementation by allowing dictionaries to be created for features of each part, thereby allowing easy addition to a part when new features are being implemented. An if statement can then be used to determine what is displayed based on the selected feature.

Numbers have thousand separators added in order to create an easier viewing experience.

All spark queries were moved to a file named `sparkcmds.py`, this helps increase readability by isolating the majority of logic to this file, leaving the files for the console and web app to focus on displaying the resultant data.

1.2.1 Console

The inquirer library was used in this practical in order to eliminate user error. This library allows the user to select options on a CLI through a combination of arrow keys, the enter key, and the space bar. Through this, no validation is required in order to check if a user has selected a real local authority or year for example.

1.2.2 Web App

Streamlit was use for the web app for similar reasons to inquirer. The benefit of streamlit over the console based inquirer is that it interacts like a website would without the need for any HTML/CSS and JavaScript, simply using python.

1.3 Implementation

Preprocessing is done before the start of the program by gathering common data like the local authority names in England and the time periods in the data set.

Examples of functions that do preprocessing include `getLas(data)` which retrieves all distinct local authorities through the use of the `la_name` column, filtering down to the `Local authority` geographic level in order to speed up this query as much as possible.

1.4 Implemented Features

Part 1	**Part 2**	**Part 3**
Read the dataset using Spark	Compare two local authorities by year	Visualisation and interaction of the dataset
Store the dataset using Spark	Chart/Explore the performance of regions (2006-2018)	Analysis of relationship between school type, pupil absences, and the location of the school
Pupil enrolments per local authority (singular/list)		
Authorised absences (medical/illness)		
Unauthorised absences by year (region/local luthority)		
Top 3 reasons for authorised absences		

Table 1: Features implemented in each part as a cell value

2 Part 1: Core Features

2.1 Design

The file `p1Console.py` is used for the console application, with data being read and stored as taught in the lectures using PySpark.

2.1.1 Pupil Enrolments

To allow a user to search by local authority, even if given a list, it was decided that when only one local authority is chosen that a list of size 1 would be created to simplify implementation. This list is then iterated over to get the relevant dataframes for each local authority which are then shown.

2.1.2 Authorised Medical Absences

A sentence was used when displaying the total number of authorised absences to give context as to what authorised medical absences mean.

2.1.3 Top 3 Reasons for Authorised Absences

The columns chosen for the displayed dataframe were time period, the number one cause of authorised absences, followed by rank 2 and 3 for readability.

2.2 Implementation

2.2.1 Pupil Enrolments

The dataframe created from the csv file and selected local authority when given to the `getLAEnrlmnts(data, la)` command selects only enrolments and time period as the user can still see the local authority they have selected. It is then filtered to only rows with a local authority matching that given. The geographic level is also filtered down to local authority as the geographic level can be school whilst still having a matching local authority name, doubling the actual number of enrolments. Finally, school type was set to total as we want total number of pupil enrolments in the region.

2.2.2 Authorised Medical Absences

The given dataframe has the authorised appointments and illnesses columns selected as these are used to calculate the total number of pupils who were given authorised absences due to appointments or illness. These columns were filtered so that the time period matches 2017-2018 as that is what the spec has specified, in addition to the school type asked for by the user. The geographic level is then set to national to narrow down the number of rows and increase efficiency for when these rows are reduced using the add operator to get total number of authorised absences.

2.2.3 Unauthorised Absences

The data is filtered down to the time period and geographic level specified by the user, and the school type specified to be total as specified.

2.2.4 Top 3 Reasons for Authorised Absences

The selected columns were all the reasons for an authorised absence and time period so that we can determine what absence types are most common for each year. To get the top 3 reasons, we collect so that we can iterate through and find the top 3 reasons. We then find the top 3 reasons for each year by pushing each element onto a max priority queue for each year, popping 3 times to get the top 3 for each year.

2.3 Difficulties Encountered

The main difficulty encountered in this section was getting to grips with the data and learning the spark functions and methods necessary to implement the required features. The other difficulty was trying to determine the top 3 reasons for authorised absences due to attempts to do so using just PySpark which did not seem to be possible, leading to a pure algorithmic approach to the problem.

3 Part 2: Intermediate Features

3.1 Design

3.1.1 Comparison of Local Authorities

The user is asked for which two local authorities they would like to compare in addition to the time period they would like to compare at. The comparison was chosen to be done by time period as contextual information not including the time period would be lost otherwise. This way a user can also see how differences change over time for the two local authorities.

Percentages are commonly used in this section as it allows comparison of regions with differing enrolments, sessions etc.

The next column chosen was enrolments and the number of persistent absentees. With this information a user would be able to learn the size of the student population in a local authority and the percentage of which were persistent absentees.

After this the percentages of each school type was found for each local authority. This can help a user understand why some statistics are the way they are. For example a local authority with a large number of state funded primary schools could have a higher rate of absences due to illness as children may have yet to fully develop immunity to common illnesses.

Then the number of overall sessions is shown for each local authority with which the following percentages are calculated using.

The percentage of normal vs absence sessions is next. Here, the dataframe has columns `normal_percent`, `sess_overall_exc_pa`, and `overall_pa_10_exact_percent`. These columns detail the percent of normal sessions, the percent of absence sessions excluding persistent absentees, and then the percent of sessions of persistent absentees. With this you can determine how big of an issue absence and persistent absentees is.

After this is a percentage breakdown of all the reasons for authorised absences. This can help provide insight into how local authorities might differ when brought into the context of their distribution of schools. For example two local authorities with similar school distributions, but one with a higher rate of absence to study can indicate a potentially more academic local authority.

Finally percentages of unauthorised absences is used to break down the reasons for unauthorised absences. An example of why this is a useful statistic to compare is that if one local authority has a higher rate of no reason yet then it could indicate an area of improvement for policies there.

These found percentages are then put into tables for easy viewing. The rows being the local authorities and the columns the percentages. All other statistics are displayed with the local authority followed by the value.

3.1.2 Performance of Regions in England (2006-2018)

To chart the performance of regions in England between 2006 and 2018, the plotly library was used to create the required figures. A line chart was used to compare the regional absence rates over the given time period, providing a visual tool that can help understand the average change in absence rate between the regions over time, along with identifying any outliers. The national average was also added as a point of comparison as a baseline.

After this, the absence rates in the previous chart are used to rank the regions from best to worst, which is then plotted onto a heat map. With this it's easy to identify changes in absence rate of a region in comparison to its peers by its change (or lack there of) in the rankings.

The rankings from the heat map are then used to find the average rank of each region, from which they are ranked again to show their average ranking over time in a table, allowing us to identify which regions on average are doing the best in terms of absence rate, and the worst.

3.2 Implementation

3.2.1 Comparison of Local Authorities

The function `getCmpData(data, cols, sumMap, la, year, groupBy)` is used to retrieve data used for comparison for a local authority. The columns given are those selected from the provided dataframe, and the year and local authority are used in conjunction with setting the geographic level to local authority when filtering. School type is set to not be total as we are wanting to compare school types in one of our comparisons. These are then grouped by the specified column `groupBy`, and then aggregated on the specified columns the sum operator, taken from `sumMap`.

The resulting dataframes are then put into a dictionary. For the sections that use percentages, the `getPercentages(data, numer, denom)` function will create new columns with `_percent` appended onto the end of the column name you are wanting to find the percentage from, defined by `numer`, `denom` denotes the column to use as the denominator in the percentage calculation.

3.2.2 Performance of Regions in England (2006-2018)

The function `getAnalysis(data)` is used to retrieve the line chart, heat map, and average rankings of each region. The results from which each statistic is based off of is found by selecting the time period, region name, and overall absence rate from the original dataset. This data is filtered to the geographic level `Regional` and school type `Total` so that only the relevant data is retrieved. The national average absence rate is then selected and unioned with the regions to get the resulting chart.

To create the heat map, three steps are necessary. The results dataframe is collected so that it may be iterated over, then an intermediary matrix is created to transform the collected results into an adjacency matrix using a dictionary that is the correct format for the heat map function, before finally changing the intermediary matrix into a 2d array for plotting.

The intermediary matrix is necessary as the rows and columns of the original collected dataframe has each row contain the time period, region name, and absence rate as columns. When what was necessary is for the rows to be the region names and the columns the time period.

Finally, for the average rankings table the heat map is simply iterated over to get the average ranking for each region, from which a ranking could be determined after sorting. A spark dataframe is then created using the newly found rankings for easy viewing.

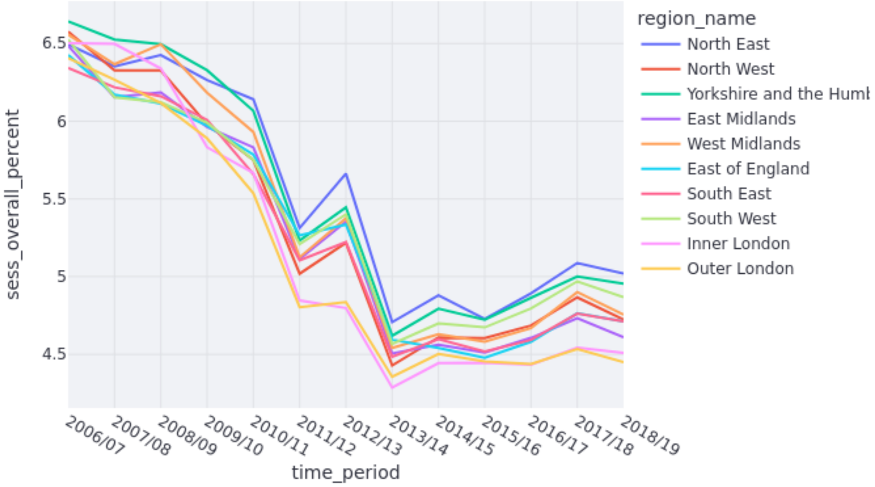

Figure 1: Absence Rate Over Time for English Regions

3.3 Analysis of Performance of Regions in England from 2006-2018

Figures 1, 2, and table 2 are the results from running the program. From figure 1, we can see that over time attendance has trended upwards for all regions between 2006 and 2018 on average. However, spikes in absences can be found like in the time period 2012/13, with attendance peaking in 2013/14. After this period though, it can be seen that absences have been slowly increasing over time on average.

In figure 2, we can see view changes in ranking of a region with respect to their peers over time. Using this comparison, it can be found that regions such as the north east have actually gotten worse over time when compared to its peers. In comparison, the North West has gotten better, peaking at

Rank	**Region Name**
1	Outer London
2	Inner London
3	South East
4	East Midlands
5	East of England
6	North West
7	South West
8	West Midlands
9	North East
10	Yorkshire and the Humber

Table 2: Ranking of Average Absence Rate Per Region from 2006-2018

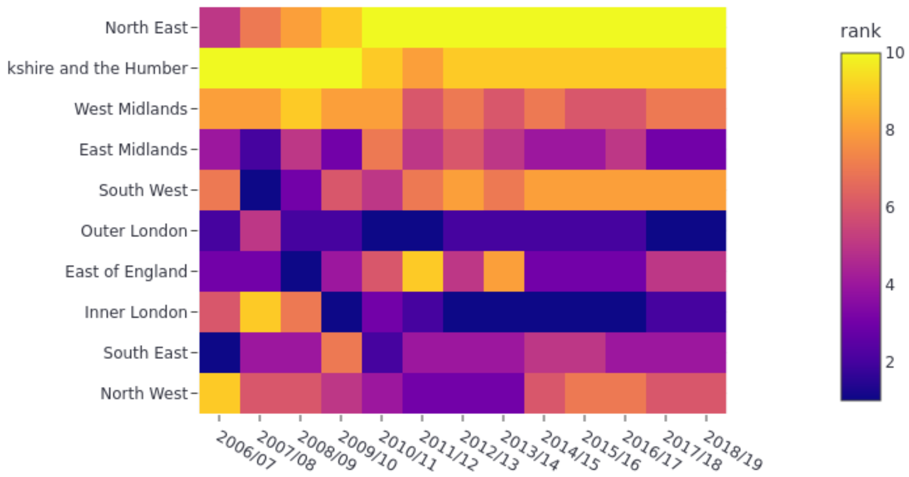

Figure 2: Ranking of Regions by Absence Rate from Best to Worst

rank 3 in the time period 2011 - 2014, beating out previously well performing regions like the South East, which ranked 1st in 2006 - 2007. Inner London perhaps has the greatest change in ranking, going from as low as rank 9 in 2007 - 2008, to holding a 5 year streak at rank 1 between 2012 and 2017, holding rank 2 when the data ends.

Table 2 shows us that regions around London in the South Eastern parts of England predominantly have the best average ranking, with northern regions having the worst by far.

3.4 Difficulties Encountered

The main difficulty in part 2 was determining what data to select in order to provide the best comparisons between local authorities and regions. Followed by what would be the best way to chart/explore the data when comparing regions to get the best analysis possible.

4 Part 3: Advanced Features

4.1 Design

4.1.1 Visualisation and Interaction

A web app was created using `streamlit` in order to make visualisation and interaction possible. Previously in the console program the dataframes were just displayed in a table format. With this, the tables can be displayed as plots and other charts when applicable as well. This allows an easier comparison of data like in the intermediate feature of comparing local authorities, where the percentages can be plotted as sections of a bar chart. This provides an easy visual way of comparing two local authorities by just looking at the difference in size between various regions.

The addition of buttons and other common aspects of websites makes navigation to target features simpler as well, as you can simply select from a drop down box whenever, where previously you had to select a feature before being able to select another one.

Plots can be interacted with by simply clicking and dragging to zoom into features, and toggling items in the legend in order to enable and disable that section of the plot in addition to other features such as maximising the plot to full screen for better viewing.

4.1.2 Links Between School Type, Pupil Absences and Location of School

For this feature, school type, pupil absences, and location of school were plotted on a heat map in order to enable easy contrast and compare of data points. A slider is used to allow the user to change the time period to allow differences over time to be visible on the same plot. Pupil absence rate is used so that schools types can be compared effectively.

4.2 Implementation

4.2.1 Links Between School Type, Pupil Absences and Location of School

The heat map is found is created as before in part 2 when comparing regions in England. The legend for the heat map plot is fixed as otherwise the min and max colours would be the min and max for each plot, thereby not allowing the plots to be compared between years.

4.3 Analysis of Relationship Between School Type, Pupil Absences and Location of School

Student absences in the starting period of 2006-2007 varies a decent amount between regions, especially in special schools, becoming less variable in primary schools and the least variable in secondary schools.

As time goes on, it can be found that absence rates become more consistent between regions, culminating in 2018-2019 being almost solid in colour for each type of school.

The most distinct difference that can be seen in this data is that school type matters most in terms of absence rate, as primary schools always have very low absence rates, hovering around 4-5%, with high schools usually marking higher by 1-4%, and finally special schools which usually have absence rates around 10-12%

High schools can be found to have the greatest change in absence rate, decreasing from around 8% absence rate to around 5%.

Finally, it can be seen that average absence rate decreases everywhere for all regions and school types over time.

4.4 Difficulties Encountered

The main difficulty in this section was learning how to use the streamlit library to create the web app.

5 Summary

From this practical I was able to get a good understanding of how to use Apache Spark by operating on the provided dataset and optimising queries in order to provide a good user experience.

The importance of understanding your dataset was also brought to life by this practical, as mistakes such as not filtering the geographic region correctly can lead to results being doubled due to counting more than once, or including the total when it is not necessary.

Determining what data/columns was useful or not is another point which had to be learnt, as what data you choose impacts the analysis possible significantly.

To conclude, this practical provided great insight into how one should approach analysing a large data set using apache spark.

6 Final Word Count: 2710