International Journal of Database Theory and Application
Vol.9, No.12 (2016), pp.267-276
http://dx.doi.org/10.14257/ijdta.2016.9.12.24

Pig Vs. Hive Use Case Analysis

Danielle Kendal[1], Oded Koren[2] and Nir Perel[3,*]

Department of Industrial Engineering and Management,
Shenkar – Engineering, Design, Art.
12 Anne Frank St., Ramat-Gan, Israel.
[1]daniellekendal@gmail.com, [2]odedkoren@shenkar.ac.il, [3]perelnir@gmail.com

Abstract

Corporations are changing their practices to data-driven big data initiatives, as big data analytics has provided companies with the ability to grow their businesses and increase competition. As the importance of data analytics grew, so accordingly did the size of the data to analyze, thus demanding a more powerful data platform. This paper shows a case study of two High Level Query Languages that are constructed on top of Hadoop MapReduce; Pig and Hive. By creating a query in each query language, both resulting in an identical output, and by running each query 30 times on 2 different sized files (120 runs total), this comparison provides a statistically significant conclusion.

Keywords: *Big Data; Performance; Hadoop; Pig; Hive*

1. Introduction

Considering the growing mass of data, there is a prediction that in the next few years the amount of valuable data will increase significantly, and provide even more actionable information [11]. Many organizations have now come to realize that acquiring the appropriate technology with which to analyse their big data is a key to the discovery of trends, patterns, correlations and many different insights that could potentially affect future decisions and company strategy.

With organizations using more data every day, selecting the right big data develop platform is vital. As there are many different open source platforms and tools to choose from, the process of choosing the right one requires the consideration of many factors, such as, implementation time and difficulty, cost, employee learning curve, hardware availability, process time, language complexity and more. These make the task of choosing between the platforms quite challenging. The variety of tools, functionalities (as part of them described at [10]) and the complexity of how to use them/combine them together, demand additional effort, skills and demands from the organization that may want to use a big data platform for implementation.

Apache Pig[1] is a high-level platform that was first developed by Yahoo, and by 2007 it was transferred to Apache Software Foundation (In the rest of the paper we will refer to Apache Pig as Pig). The language used on Pig platform, named ‘Pig Latin’, is a data flow language, i.e. it is capable of connecting tools together. Pig can process complex structures of data without the requirement of a structured data set [1]. Pig’s infrastructure has a compiler which turns Pig Latin into MapReduce programs and is designed to work in batch processing. In a recent study on Pig's performance, it was determined that the higher the level of decentralization the faster the processing time [3].

The Apache Hive TM[2] was first developed by Facebook in 2007 (In the rest of the paper we will refer to Apache Hive TM as Hive). Facebook implemented a more familiar

[*] Corresponding Author

[1] https://pig.apache.org/

[2] APACHE HIVE TM available at: http://hive.apache.org/

ISSN: 2005-4270 IJDTA

concept such as relations, columns and a subset of SQL to Hadoop’s unstructured environment [8].

Hive is an open source platform using a language similar to SQL to create queries named HiveQL, also known as an ETL tool [4]. These queries are compiled into jobs executed on Hadoop [9]. HiveQL can also be extended into user defined functions, table functions and aggregates.

There are several publications comparing the performance of Pig and Hive. In [7], the authors compared between Pig, Hive and JAQL. They found that Hive outperformed in terms of running times and the length of each query. In [6] the author used the TPC-H benchmark to compare between Pig and the Oracle SQL Engine. In this research Pig outperformed. During Yahoo evaluation of Hive, they came to a conclusion that Pig is more suitable for the ETL process and Hive is more beneficial when integrated with BI tools for analysis purposes [2].

In a slightly different research it was found in [5] that the relational database management system performs better than Pig joins procedures.

2. Research Use Case

The goal of this research is to compare the performance of Pig vs. Hive in terms of running times on a variety of data sets' sizes. This research includes two queries, one written for Hive and one for Pig; running both queries outputs the same result. During the research, each query ran 30 times for each file size. There are two different file sizes (1GB file containing 29,653,834 rows, and 2GB file containing 59,299,979 rows), resulting in a total of 60 runs per query and a total of 120 runs for the entire research (as described in Table 1). The whole research was conducted on a single node environment.

Table 1. Research Runs

	PIG – total runs	HIVE - total runs	Total
1GB Dataset	30	30	60
2GB Dataset	30	30	60
Total	60	60	120

The query chosen for the research was created to simulate a common need in the BI world as the query’s purpose is to aggregate customer orders. The query ran on three files that simulate random order entry data. The query’s objective is to unite the files, and produce a summary table for every month of every year for each customer. The customers' data represents the number of items and the total purchase amount for each item ID. Table 2 is an example of three order entry rows from the files prepared.

Table 2. Input Sample

Customer Id	Item Id	Number of Items	Date	Price per Item	Purchase Amount
111	1	2	5/11/2016	18	36
111	1	5	5/24/2016	18	90
222	69	4	5/26/2016	26	104
333	34	1	3/04/2016	150	150

The main purpose of the query is to create a summary table by using these actions:
(a) Union of the three files;
(b) Creation of a column with the year and the month in this format: YYYY-MM;
(c) Grouping by year-month column, the customer id and the item id column.

(d) Storing the output.

Table 3 presents the output of the above procedures (values for example only).

Table 3. Output Sample

Year-Month	Customer Id	Item Id	Number of Items	Purchase Amount
2016-05	111	1	7	126
2016-06	222	69	4	104
2016-05	333	34	1	150

The results in Table 3 indicate a summary of purchases for each client showing the number of different items purchased each month, and the total purchase amount.

Note: While conducting this research a date format issue raised. The Hive platform is unable to apply date functions in the following format: YYYY-MM-DD and so it needed to be altered to a format that includes a timestamp like so: YYYY-MM-DD HH:MM:SS. The files for Pig and Hive had the **same number of rows** but had different sizes due to the alteration in the date format. The 1GB file for Pig was transformed into a 1.2GB file (in this article, we will use the term "1 GB" for both 1 and 1.2GB) with the **same** number of rows, and the 2GB file transformed into a 2.5GB file also with the **same** number of rows (in this article, we will use the term "2 GB" for both 2 and 2.5GB).

With Hive, the query is easily scripted using an inner query to union the files and an external query to create the column and group by the key fields. The hive query used in this research is as follows:

```
create table Hive1 as
select   YearMonth ,CustomerID,ItemID,SUM(NumberOfItems) as
NumberOfItems,SUM(TotalPurchaseAmount) as TotalPurchaseAmount
from (
select concat(year(cast(date as date))+'-'+month(cast(date as date))) as
YearMonth,CustomerID,ItemID,NumberOfItems,TotalPurchaseAmount from file1
union ALL
select concat(year(cast(date as date))+'-'+month(cast(date as date))) as
YearMonth,CustomerID,ItemID,NumberOfItems,TotalPurchaseAmount from file2
union ALL
select concat(year(cast(date as date))+'-'+month(cast(date as date))) as
YearMonth,CustomerID,ItemID,NumberOfItems,TotalPurchaseAmount from file3
  )k
  group by YearMonth,CustomerID,ItemID
```

However, in Pig Latin, a more complex query is needed, using more functions to group the data. The process can be described as follows:

(a) Union of the 3 loaded files;
(b) Creation of an additional column with the year-month format;
(c) Grouping by the key and flatten by summarizing the numbers.

The final stage for each query is storing the new created table as a new file. The Pig query used in this research is the following:

```
FullFile= LOAD '/user/cloudera/{File1Pig,File2Pig,File3Pig}' USING PigStorage(',') as
(
CustomerID:int,ItemID:int,NumberOfItems:int,Date:Datetime,PricePerItem:int,TotalPurchaseAmount:int);
New= Foreach FullFile generate * CONCAT((Chararray)GetYear(Date),CONCAT(+'-'+(Chararray)GetMonth(Date))) as YearMonth:Chararray;
New2= Group New By(YearMonth,CustomerID,ItemID);
New3= foreach New2 Generate FLATTEN (group) as (YearMonth,CustomerID,ItemID),
SUM(New.NumberOfItems) as NumberOfItems,
SUM(New.TotalPurchaseAmount) as TotalPurchaseAmount;
Store New3 Into '/user/cloudera/FinalPig';
```

3. Results and Analysis

In this section we present the results of the research as well as the analysis conducted in order to compare between the performances of Pig vs. Hive. The results contain a total of 120 running times collected manually after each run using the timestamps in the log. We analysed the results by using a Two-way Anova, in order to study the influence of both file size and platform on the running time, as well as the interaction between them. Figure 1 presents the running times results for the 1GB file size, and Table 4 presents statistical metrics for the 1GB running time results.

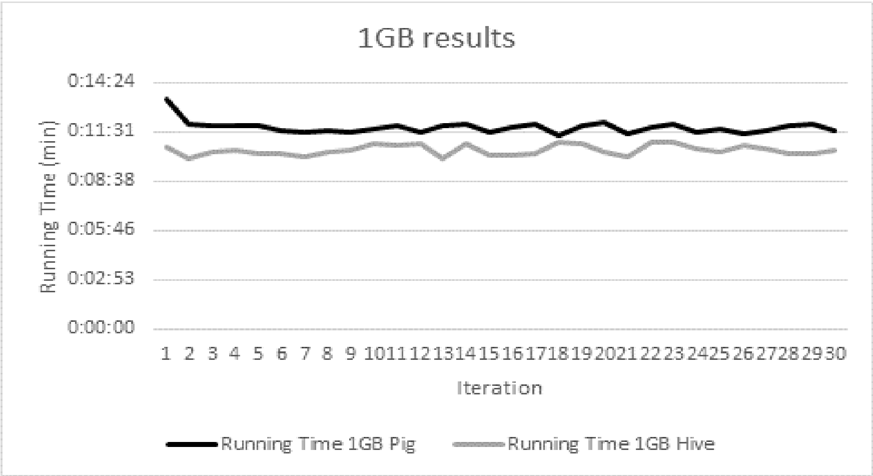

Figure 1. Running Times Results for 1GB File Size

Table 4. Group Statistics for the 1GB File Size

	type	N	Mean	Std. Deviation	Std. Error Mean
Running time	Pig, 1GB	30	11.79950	.382402	.069817
	Hive, 1GB	30	10.46663	.305627	.055800

Table 4 shows that the mean running time in Pig is 11.7 minutes, slower than Hive with a mean of 10.4 minutes. In addition, the above table provides an indication of the data’s variation by presenting the standard deviation, which in this case clearly shows that the Hive running time results are closer together. This is also shown in Table 5, which presents the results of two-independent samples T-test. It is clearly seen from the output that Hive is (statistically significantly) faster than Pig.

Table 5. Results of Two-Independent Samples T-test for the 1GB File Size

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Running time	Equal variances assumed	0.014	0.906	14.91	58	0.00	1.33	0.089	1.15	1.51
	Equal variances not assumed			14.91	55.31	0.00	1.33	0.089	1.15	1.51

To provide a clearer presentation of the data collected, and the distribution of values, Figure 2 presents the 1GB Pig running times on the left, and the 1GB running times for Hive on the right.

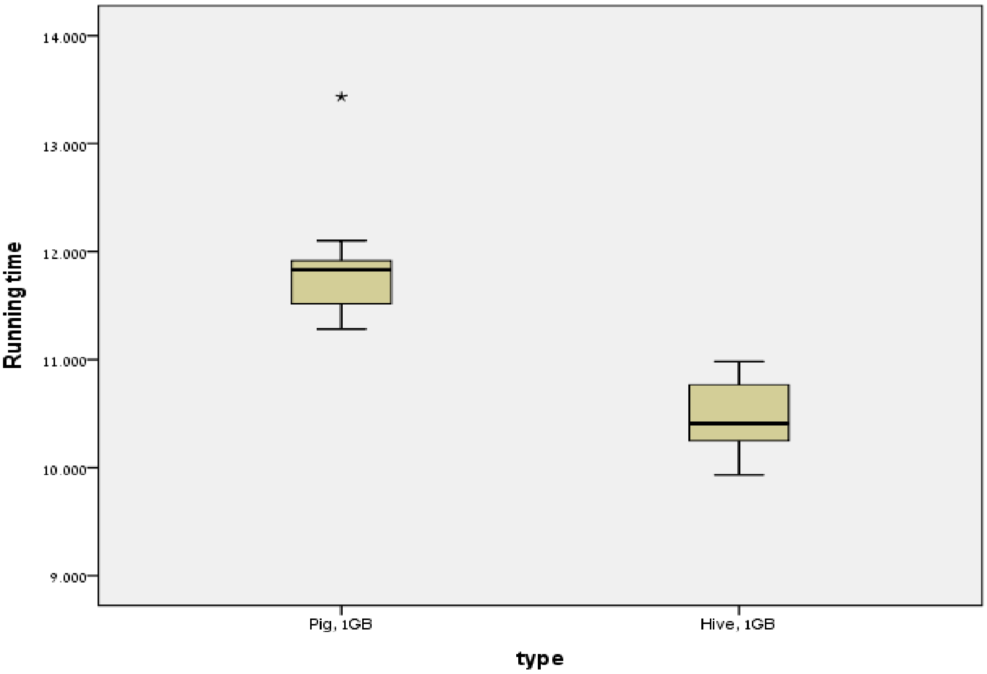

Figure 2. Box Plot for 1GB Running Times

Figure 3 presents the running times results for the 1GB file size, where Table 6 presents statistical metrics for the 2GB running time results. It shows that mean running time in Pig is 22.8 minutes, while Hive was faster with a mean of 15.59 minutes. The difference between the Pig and Hive mean results is significantly larger than the difference in results for the 1GB means.

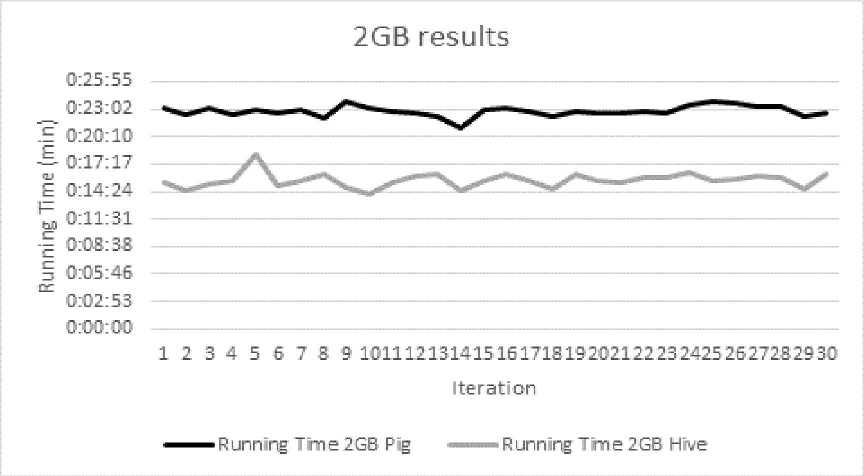

Figure 3. Running Times Results for 2GB File Size

Table 6. Group Statistics for the 2GB File Size

	type	N	Mean	Std. Deviation	Std. Error Mean
Running time	Pig, 2GB	30	22.83227	.564224	.103013
	Hive, 2GB	30	15.50277	.609189	.111222

Another significant difference from the 1GB results is that in this case the smaller standard deviation belongs to Pig. Again, a two independent samples T-test indicates that Hive is significantly faster than Pig, see Table 7. Figure 2 presents the 2GB Pig running times on the left and the 2GB running times for hive on the right. The circle represents an abnormal result (outlier). It is clear from figure 4 that once again all the Hive results were lower on the chart than the Pig results, which means that also in this case Hive is faster than Pig.

Table 7. Results of Two-Independent Samples T-test for the 2GB File Size

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Running time	Equal variances assumed	0.308	0.581	48.35	58	0.00	7.33	0.153	7.03	7.63
	Equal variances not assumed			48.35	57.66	0.00	7.33	0.153	7.03	7.63

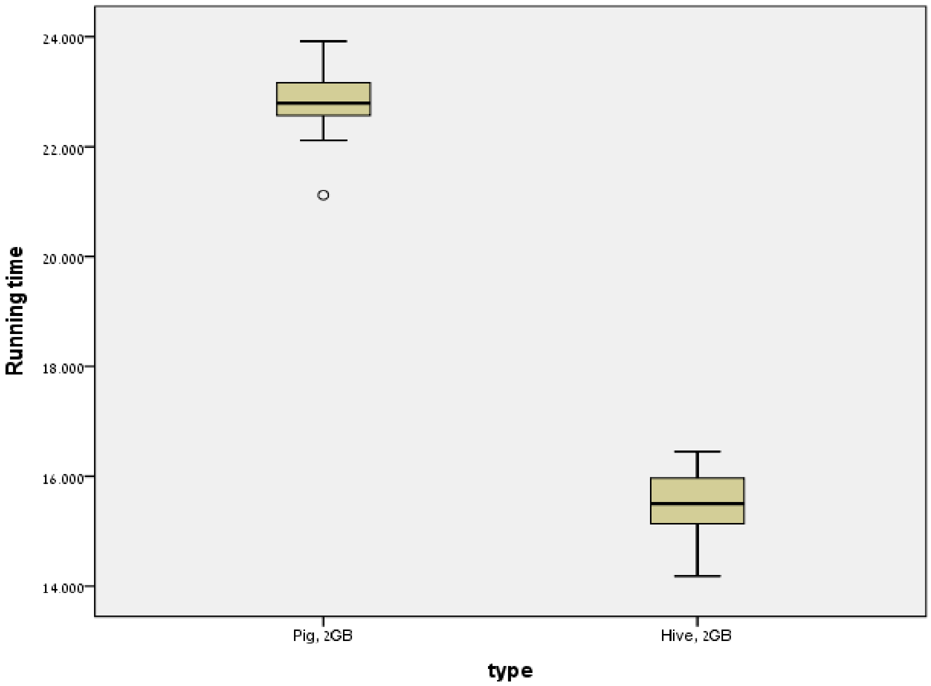

Figure 4. Box Plot for 2GB Running Times

We also conducted a Two-way ANOVA test to examine the influence of the independent variables, the file size and the platform. The dependent variable for the test is the running time. The results are summarized in Table 8, from which we conclude the following: The platform (software) affects the running time. More specifically, Hive outperforms Pig, with faster running times. Also, when considering the interaction between the platform and the size of the file, it also affects the running time. Figure 5 illustrates this interaction. For file size of 1GB, the mean running time difference between Pig and Hive is smaller than the one in file size of 2GB.

Table 8. Results of Two-Way Anova

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	2769.043[a]	3	923.014	3973.801	.000
Intercept	27543.761	1	27543.761	118582.572	.000
GB	1936.572	1	1936.572	8337.411	.000
Software	562.774	1	562.774	2422.881	.000
GB * Software	269.697	1	269.697	1161.111	.000
Error	26.944	116	.232		
Total	30339.748	120			
Corrected Total	2795.987	119			

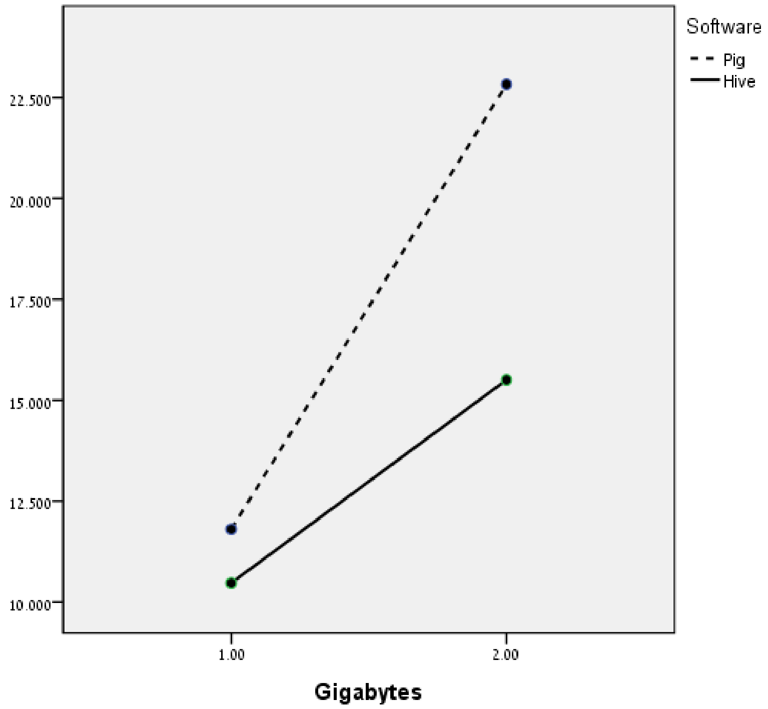

Figure 5. Interaction Plot – Estimated Marginal Means of Running Time

4. Conclusions

In this use case, we compared the performance of Pig and Hive by creating a query in each high-level query language that produces the same output. In addition, we created six files, three 1GB files and three 2GB files. Each query ran 60 times, 30 times on each file size, resulting in 120 running time values which were analysed.

While creating the queries in Hive and Pig, we found that in this case the Hive query is more efficient in comparison to the Pig query, in the order and number of actions needed to get the required output.

The data analysis was done by using the distribution of the running times for each platform on each file size for and comparing the results. The summary tables and box plot charts clearly show that Hive is faster than Pig. In addition, we performed a Two-way ANOVA test to find any interaction between the file size and platform to the running time results collected. The test, as expected, revealed that both the file size and the platform affect the running time.

In this use case, we found that Hive would be a preferable platform for those large companies aggregating large files, as the running times have proved to be shorter and the query more efficient.

Additional research using a variety of multiple data nodes environments should be conducted to further investigate this use case. In addition, this use case was created to compare aggregation queries. Further analysis can be done on other common query needs of big corporations using BI tools.

References

[1] S. Dhawan and S. Rathee, "Big data analytics using Hadoop components like Pig and Hive", American International Journal of Research in Science, Technology, Engineering & Mathematics, vol. 2, **(2013)**, pp. 88-93.

[2] A. Gates, "Pig and Hive at yahoo", YAHOO developer network, http://yahoohadoop.tumblr.com/post/98256601751/pig-and-hive-at-yahoo, **(2010)**.

[3] G. Engelberg, O. Koren and N. Perel, "Big Data Performance Evaluation Analysis Using Apache Pig", International Journal of Software Engineering and Its Applications, vol. 10, no. 11, **(2016)**, pp. 429-440.

[4] P. J. Jamack, "Hive as a tool for ETL or ELT", IBM developer Works, http://www.ibm.com/developerworks/library/bd-hivetool/, **(2014)**.

[5] S. Loebman, D. Nunley Y. Kwon, B. Howe, M. Balazinska and J. P. Gardner, "Analyzing massive astrophysical datasets: Can Pig/Hadoop or a relational DBMS help", In 2009 IEEE International Conference on Cluster Computing and Workshops. IEEE, **(2009)**, pp. 1-10.

[6] R. Moussa, "TPC-H Benchmarking of Pig Latin on a Hadoop Cluster", In Communications and Information Technology (ICCIT), 2012 International Conference on. IEEE, **(2012)**, pp. 85-90.

[7] R. J. Stewart, P. W. Trinder and H.-W. Loidl, "Comparing high level MapReduce query languages", In Proceedings of the 9th international conference on advanced parallel processing technologies, ser. APPT'11. Berlin, Heidelberg: Springer-Verlag, **(2011)**, pp. 58-72.

[8] A. Thusoo, J. Sen Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff and R. Murthy, "Hive - A petabyte scale data warehouse using Hadoop", In 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010). IEEE, **(2010)**, pp. 996-1005.

[9] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff and R. Murthy, "Hive - A Warehousing Solution Over a Map-Reduce Framework", Proceedings of the VLDB Endowment, vol. 2, no. 2, **(2009)**, pp. 1626-1629.

[10] T. White, "Hadoop: The Definitive Guide", 4th edition, OReilly Media, Sebastopol, CA, **(2015)**.

[11] http://www.idc.com/promo/thirdplatform/fourpillars/bigdataanalytics;jsessionid=ACEF8D13E59C518BDEDDA867C8108C43

Authors

D. Kendal is a Data Scientist, working with BI tools analyzing data and customizing segmentation models for clients across a variety of industries. She is in the final furlong of her M.Sc. degree in Industrial Engineering at Shenkar – Engineering, Design, Art. Her main interests are data science, big data and machine learning domains.

O. Koren has a Ph.D. from Tel-Aviv University and he is a full faculty member in the Department of Industrial Engineering and Management in Shenkar - Engineering, Design, Art. Oded's research interests are in the areas of open source development domains, Big Data, AI related aspects and mobile applications permissions.

N. Perel received a Ph.D. degree in operations research from Tel-Aviv University, Israel. He is a senior faculty member in the Department of Industrial Engineering and Management in Shenkar - Engineering, Design, Art. His research interests include operations research modeling and queueing theory. Lately he is interested in AI, and open source development.