Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics

Match performance related CK analysis and data mining in under-aged Hungarian football

Project Laboratory I.

Ákos Norbert Sepp
Business Informatics MSc

Advisors:
László Toka PhD and Alija Pašić PhD
Department of Telecommunications and Media Informatics

Budapest, Hungary

2019.

Contents

1. Importance of data analysis in football

 ‘Data is everywhere in the sport field. […] Gamblers regularly relied on data to know which horse or team to bet on. Coaches routinely utilized data to evaluate talent and potential for players. […] The use of data to help executives more appropriately manage their organizations can be seen in numerous examples across numerous division within any sport organization. The question often raised by those looking at data is: are we in fact examining the right issues with the correct data and making the correct correlation between the correct data points?’ (Fried and Mumcu, 2017.)

According to Fried and Mumcu, data analysis became a very important factor in modern sport in the last few years. Football, as one of the most popular sports, faces extra challenges in this field. Football managers are extremely under pressure by fans and club owners who are eager to see instant and continuous results. On the other hand, players, as the most valuable assets of the game and the football business machine, are more and more focused on their health status and factors that can improve their performance.

It is clear that all sides are unbelievably motivated to boost their chances to succeed and looking for all kinds of ways to reach their goal, which has been supported by investors in infrastructural ways such as building new stadiums, training accessories or regeneration centers. Fortunately, in the last decade players and managers found a new source of support from scientists and data analysts.

In 2019, Leeds United FC manager Marcelo Bielsa revealed his method of preparation for football matches during the scandalous event called ‘Leeds Spygate’ (The Guardian, 2019). In the presentation he showed how deeply data analysis is being involved in the process, including watching more than 200 hours gameplay videos and possible team formations regarding the previous 50 matches of the opponent. His visionary method and presentation simultaneously impressed and confused the English audition, which implies that there are still plenty of ways to develop analytical methods in the best football leagues.

But why is sports analytics so important for a small country like Hungary? In recent years, Hungarian football started to develop step-by-step, due to governmental financial supports in football academies and infrastructure. Although the circumstances nowadays are far better than used to be to reach the expected results in international competitions, it is clear that there are still several fields to improve such as data analysis. As a first step, trainers recently started to measure and record specific player performance data for further analysis.

In this research I focused on implementing a data analysis and data mining process called Creatine-Kinase (CK) level analysis. CK is an enzyme which can be evinced from the blood serum and is strongly connected to muscular damage.

In the following pages I am presenting the background, the modeling process and the results of my research. At last I present the deployment of my work, a ready-to-work predictive CK calculator. The subjects of my analysis were under-aged Hungarian football players, whose names had been anonymized for this study.

2. CK related sport analytics

2.1. About Creatine Kinase (CK)

Measuring Creatine Kinase serum level is a well-known and popular measurement factor in sport analytics, due to its connection to the body’s regeneration status.

According to Epstein (1995), ‘one of the most valid and reliable methods for assessing muscular damage is to check for increases in blood serum levels of creatine kinase (CK), the primary enzyme regulating anaerobic metabolism, because a high percentage of the body’s CK is present in skeletal muscle tissue.’

Mougios (2007) suggested, that ‘the serum CK concentration serves as an index of both overexertion and adaptation of the muscular system to repeated bouts of exercise. As such, CK is one of the top choices of athletes and coaches when requesting a biochemical profile.’

In my research, young football players’ CK level were measured several times, along with their physical performance, which led us to the opportunity of creating a model which can take into consideration the muscular damage as well.

2.2. Catapult system

Catapult is complex movement tracker and recorder system, specifically developed to analyze, compare and evaluate sporting movements for professional athletes. As a result of the automatic data gathering provided by the Catapult system, key performance indicators became objectively measurable and calculable.

The Catapult system records and immediately calculates more than a 1,000 different attributes simultaneously with built-in sensors like accelerometer, gyroscope or magnetometer, and pre-categorizes them on-the-run. Table 1 helps to understand more deeply the huge scale of data measured by Catapult (attributes with italic style are not automatically provided).

Internal load		**External load**	
cardiovascular	metabolic	locomotorical	mechanical
pulse	*blood serum*	GPS	IMA
pulse data	*lactate*	covered distance	acceleration, deceleration
time spent in target zone	*CK*	speed	fast changes of direction
rest HRV	*IGG*	time spent is speed zones	jumps

Table 1 - KPIs measured by Catapult system (Source: http://www.cardioc.eu/catapult-arendszerfelepitese/)

3. Data mining in general

3.1. CRISP-DM method

Data mining is a creative process where different facilities and knowledge is needed to succeed. For a long time there was no unified framework for data mining projects, success mostly based on the researcher's way of work, which implied that results were hardly repeatable.

The goal of CRISP-DM (Cross Industry Standard Process for Data Mining) project was to eliminate these factors through a model which is independent from industry and technology, and can be the framework for all data mining projects, as Wirth and Hipp (2000) summarizes it. CRISP-DM describes a data mining life cycle, which contains the main steps for the project (Figure 1).

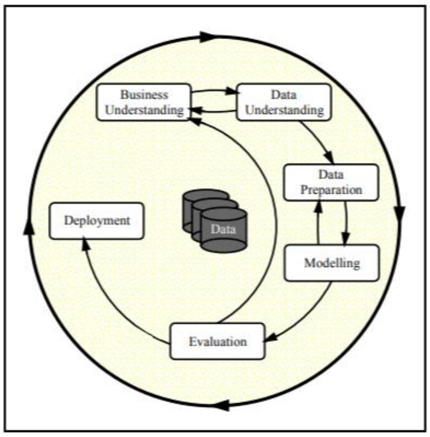


Figure 1 – CRISP-DM life cycle (Source: Wirth and Hipp, 2000, pg. 33)

3.2. Using CRISP-DM in the study

My research was based on CRISP-DM method, with the following steps:

1. Understanding the environment this project focuses on and all data came from
2. Examining and preparing the raw data
3. Presenting the modelling process on the cleaned dataset
4. Presentation of results, along with the evaluation
5. Presentation of implementation and deployment
6. Discussion of further possibilities of improving the model and the analytics.

4. Relationship between player performance and physio result

Fortunately, CK reference numbers for active male football players were calculated by Mougios (2007). Based on that research, it is possible to compare our dataset's intervals for further understanding (Figure 2 and 3).

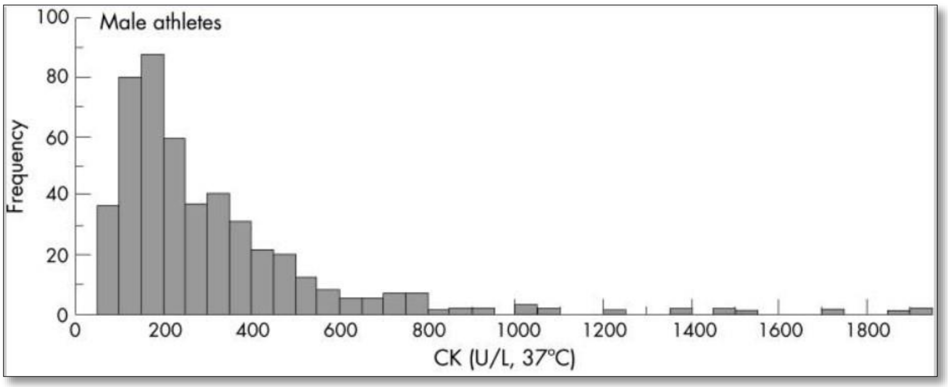


Figure 2 – Reference distribution for CK numbers in male athletes (Source: Mougios, 2000)

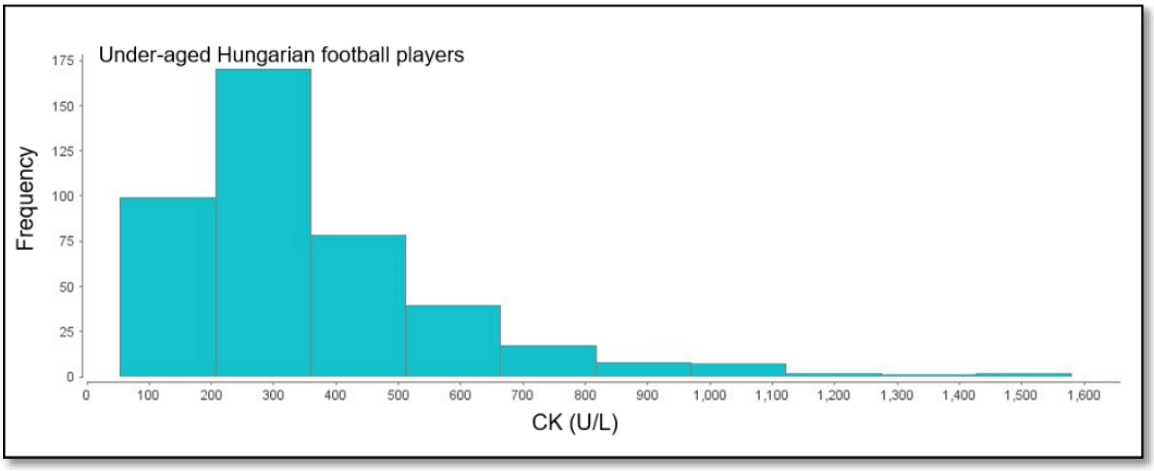


Figure 3 – CK distribution in the examined dataset (Source: own source)

As we can see, the distributions are extremely similar. Based on that fact, we can similarly use Mougios’s other conclusion, which says “the values used in this analysis should be considered as the cumulative effect of recent training sessions in conjunction with the repeated-bout effect”.

As well as he states that although gender plays a huge role in CK levels, “age did not seem to affect the reference interval for CK in athletes considerably”. That means, we can use other CK-based researches’ results as well, while they have no differences to our under-aged datasets.

Ehlers in 2002 revealed, that CK peak level elevations usually occur from 8 to 48 hours after the training session, with a peek of approximately 18 hours. To normalize this value, the body needs 2-3 days. It is almost the same what Coelho et al. say (2010). They concluded that peak points are between 12 and 20 hours after the game, returning to normal within 60-65h.

To all things consider, we can say that it is important

1) to aggregate all the physical training result (Mougios)
2) to take in consideration the last CK level as well (Mougios)
3) to analyze the past 12-20 hours to effectively measure all the effects behind the peak point result (Ehlers and Coelho et al.).

5. Examining relationship between CK and match performance

5.1. What is the main question of the study?

In this study I focused on the prediction of future CK levels based on measured physical activity provided by the Catapult system. It was important to not only create a predictive model but to deploy it in a usable format as it was needed by real-life under-aged coached to work with.

Likewise it was another important part to examine and calculate regeneration patterns in the players result, and to build as personalized models as possible.

Given the above the main questions of study is the following:

1. *How can we predict young players' future CK level based on their physical performance, and is personalization possible?*
2. *How can this prediction help the players to reach their optimal performance?*

5.2. What was the approach?

With the provided physical data and the CK levels from the Catapult system and the blood tests, I firstly put together a cleansed, aggregated dataset with all the available result. Than to reach personalization, with an analytical and data mining software called RapidMiner, I created regeneration clusters, than built predictive models for each cluster.

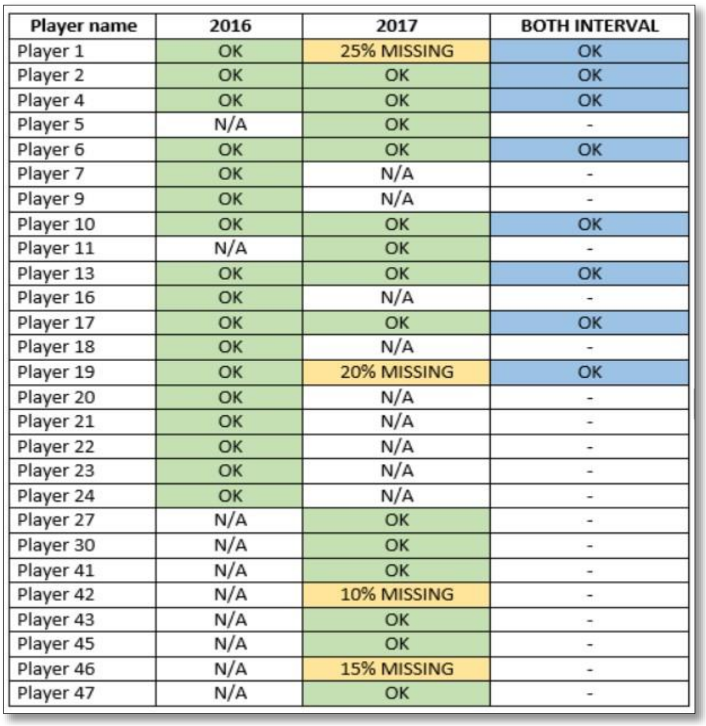
After the model was created, I manually examined what kind of relation have the CK levels with the players training load. Based on that I declared their optimal CK interval for each player for the best performance.

Lastly, I wrote in JavaScript, HTML and CSS a modeling webpage to deploy the results.

6. CK and performance data preparation

6.1. The raw datasets

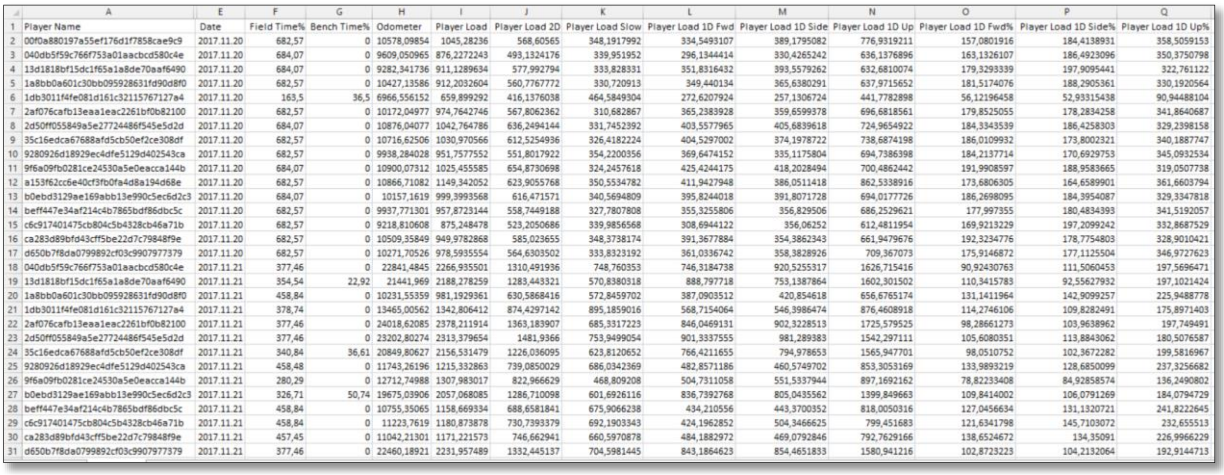
Firstly, it is important to define what kind of and how many data we have. In Figure 4 we can see the CK data availability per player in the examined time window. In Figure 5, I show an exemplary piece of the performance table to help to understand the player performance deeper.



Player name	2016	2017	BOTH INTERVAL
Player 1	OK	25% MISSING	OK
Player 2	OK	OK	OK
Player 4	OK	OK	OK
Player 5	N/A	OK	-
Player 6	OK	OK	OK
Player 7	OK	N/A	-
Player 9	OK	N/A	-
Player 10	OK	OK	OK
Player 11	N/A	OK	-
Player 13	OK	OK	OK
Player 16	OK	N/A	-
Player 17	OK	OK	OK
Player 18	OK	N/A	-
Player 19	OK	20% MISSING	OK
Player 20	OK	N/A	-
Player 21	OK	N/A	-
Player 22	OK	N/A	-
Player 23	OK	N/A	-
Player 24	OK	N/A	-
Player 27	N/A	OK	-
Player 30	N/A	OK	-
Player 41	N/A	OK	-
Player 42	N/A	10% MISSING	-
Player 43	N/A	OK	-
Player 45	N/A	OK	-
Player 46	N/A	15% MISSING	-
Player 47	N/A	OK	-

Figure 4 – CK data availability per player in the examined time window (Source: own source)

As it can be seen, there are missing values in the dataset. I marked with 'OK' label when more than 90% of the CK values were available for the player. 'N/A' means there were minimal amount of values or none at all for the examined period. If all data were available for a player, they got an 'OK' label in the '*Both interval'* column as well.



	A	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Player Name	Date	Field Time%	Bench Time%	Odometer	Player Load	Player Load 2D	Player Load Slow	Player Load 1D Fwd	Player Load 1D Side	Player Load 1D Up	Player Load 1D Fwd%	Player Load 1D Side%	Player Load 1D Up%
2	00f0a880197a55ef176d1f7858cae9c9	2017.11.20	682,57	0	10578,09854	1045,28236	568,60565	348,1917992	334,5493107	389,1795082	776,9319211	157,0801916	184,4138931	358,5059153
3	040db5f59c766f753a01aacbcd580c4e	2017.11.20	684,07	0	9609,050965	876,2272243	493,1324176	339,951952	296,1344414	330,4265242	636,1376896	163,1326107	186,4923096	350,3750798
4	13d1818bf15dc1f65a1a8de70aaf6490	2017.11.20	684,07	0	9282,341736	911,1289634	577,992794	333,828331	351,8316432	393,5579262	632,6810074	179,3293339	197,9095441	322,761122
5	1a8bb0a601c30bb095928631fd90d8f0	2017.11.20	682,57	0	10427,13586	912,2032604	560,7767772	330,720913	349,440134	365,6380291	637,9715652	181,5174076	188,2905361	330,1920564
6	1db3011f4fe081d161c32115767127a4	2017.11.20	163,5	36,5	6966,556152	659,899292	416,1376038	464,5849304	272,6207924	257,1306724	441,7782898	56,12196458	52,93315438	90,94488104
7	2af076cafb13eaa1eac2261bf0b82100	2017.11.20	682,57	0	10172,04977	974,7642746	567,8062362	310,682867	365,2383928	359,6599378	696,6818561	179,8525055	178,2834258	341,8640687
8	2d50ff055849a5e27724486f545e5d2d	2017.11.20	684,07	0	10876,04077	1042,764786	636,2494144	331,7452392	403,5577965	405,6839618	724,9654922	184,3343539	186,4258303	329,2398158
9	35c16edca67688afd5cb50ef2ce308df	2017.11.20	682,57	0	10716,62506	1030,970566	612,5254936	326,4182224	404,5297002	374,1978722	738,6874198	186,0109932	173,8002321	340,1887747
10	9280926d18929ec4dfe5129d402543ca	2017.11.20	682,57	0	9938,284028	951,7577552	551,8017922	354,2200356	369,6474152	335,1175804	694,7386398	184,2137714	170,6929753	345,0932534
11	9f6a09fb0281ce24530a5e0eacca144b	2017.11.20	684,07	0	10900,07312	1025,455585	654,8730698	324,2457618	425,4244175	418,2028494	700,4862442	191,9908597	188,9583665	319,0507738
12	a153f62cc6e40cf3fb0fa4d8a194d68e	2017.11.20	682,57	0	10866,71082	1149,342052	623,9055768	350,5534782	411,9427948	386,0511418	862,5338916	173,6806305	164,6589901	361,6603794
13	b0ebd3129ae169abb13e990c5ec6d2c3	2017.11.20	684,07	0	10157,1619	999,3993568	616,471571	340,5694809	395,8244018	391,8071728	694,0177726	186,2698095	184,3954087	329,3347818
14	beff447e34af214c4b7865bdf86dbc5c	2017.11.20	682,57	0	9937,771301	957,8723144	558,7449188	327,7807808	355,3255806	356,829506	686,2529621	177,997355	180,4834393	341,5192057
15	c6c917401475cb804c5b4328cb46a71b	2017.11.20	682,57	0	9218,810608	875,248478	523,2050686	339,9856568	308,6944122	356,06252	612,4811954	169,9213229	197,2099242	332,8687529
16	ca283d89bfd43cff5be22d7c79848f9e	2017.11.20	682,57	0	10509,35849	949,9782868	585,023655	348,3738174	391,3677884	354,3862343	661,9479676	192,3234776	178,7754803	328,9010421
17	d650b7f8da0799892cf03c9907977379	2017.11.20	682,57	0	10271,70526	978,5935554	564,6303502	333,8323192	361,0336742	358,3828926	709,367073	175,9146872	177,1125504	346,9727623
18	040db5f59c766f753a01aacbcd580c4e	2017.11.21	377,46	0	22841,4845	2266,935501	1310,491936	748,760353	746,3184738	920,5255317	1626,715416	90,92430763	111,5060453	197,5696471
19	13d1818bf15dc1f65a1a8de70aaf6490	2017.11.21	354,54	22,92	21441,969	2188,278259	1283,443321	570,8380318	888,797718	753,1387864	1602,301502	110,3415783	92,55627932	197,1021424
20	1a8bb0a601c30bb095928631fd90d8f0	2017.11.21	458,84	0	10231,55359	981,1929361	630,5868416	572,8459702	387,0903512	420,854618	656,6765174	131,1411964	142,9099257	225,9488778
21	1db3011f4fe081d161c32115767127a4	2017.11.21	378,74	0	13465,00562	1342,806412	874,4297142	895,1859016	568,7154064	546,3986474	876,4608918	114,2746106	109,8282491	175,8971403
22	2af076cafb13eaa1eac2261bf0b82100	2017.11.21	377,46	0	24018,62085	2378,211914	1363,183907	685,3317223	846,0469131	902,3228513	1725,579525	98,28661273	103,9638962	197,749491
23	2d50ff055849a5e27724486f545e5d2d	2017.11.21	377,46	0	23202,80274	2313,379654	1481,9366	753,9499054	901,3337555	981,289383	1542,297111	105,6080351	113,8843062	180,5076587
24	35c16edca67688afd5cb50ef2ce308df	2017.11.21	340,84	36,61	20849,80627	2156,531479	1226,036095	623,8120652	766,4211655	794,978653	1565,947701	98,0510752	102,3672282	199,5816967
25	9280926d18929ec4dfe5129d402543ca	2017.11.21	458,48	0	11743,26196	1215,332863	739,0850029	686,0342369	482,8571186	460,5749702	853,3053169	133,9893219	128,6850099	237,3256682
26	9f6a09fb0281ce24530a5e0eacca144b	2017.11.21	280,29	0	12712,74988	1307,983017	822,966629	468,809208	504,7311058	551,5337944	897,1692162	78,82233408	84,92858574	136,2490802
27	b0ebd3129ae169abb13e990c5ec6d2c3	2017.11.21	326,71	50,74	19675,03906	2057,068085	1286,710098	601,6926116	836,7392768	805,0435562	1399,849663	109,8414002	106,0791269	184,0794729
28	beff447e34af214c4b7865bdf86dbc5c	2017.11.21	458,84	0	10755,35065	1158,669334	688,6581841	675,9066238	434,210556	443,3700352	818,0050316	127,0456634	131,1320721	241,8222645
29	c6c917401475cb804c5b4328cb46a71b	2017.11.21	458,84	0	11223,7619	1180,873878	730,7393379	692,1903343	424,1962852	504,3466625	799,451683	121,6341798	145,7103072	232,655513
30	ca283d89bfd43cff5be22d7c79848f9e	2017.11.21	457,45	0	11042,21301	1171,221573	746,662941	660,5970878	484,1882972	469,0792846	792,7629166	138,6524672	134,35091	226,9966229
31	d650b7f8da0799892cf03c9907977379	2017.11.21	377,46	0	22460,18921	2231,957489	1332,445137	704,5981445	843,1864623	854,4651833	1580,941216	102,8723223	104,2132064	192,9144713

Figure 5 – Example of performance data (Source: own source)

Figure 6 only shows a few rows and columns of the performance dataset, however the original data table has 6587 rows and 756 columns (similarly referred as attributes).

6.2. Data cleansing and transformation

Before I could join the two dataset together, I had to sort out missing values, and those events which had no matching CK value. According to Mougios (as I mentioned in section 3) it is important to aggregate all the physical load that the player did during their performance, so it was another task to summarize all the different attributes where it was possible to do so.

The most important data preparation step was to pair the relevant CK value with the performance. As I discussed in section 3, CK elevation needs time. In Figure 6 we can see the exact time delay in the 2016 dataset.

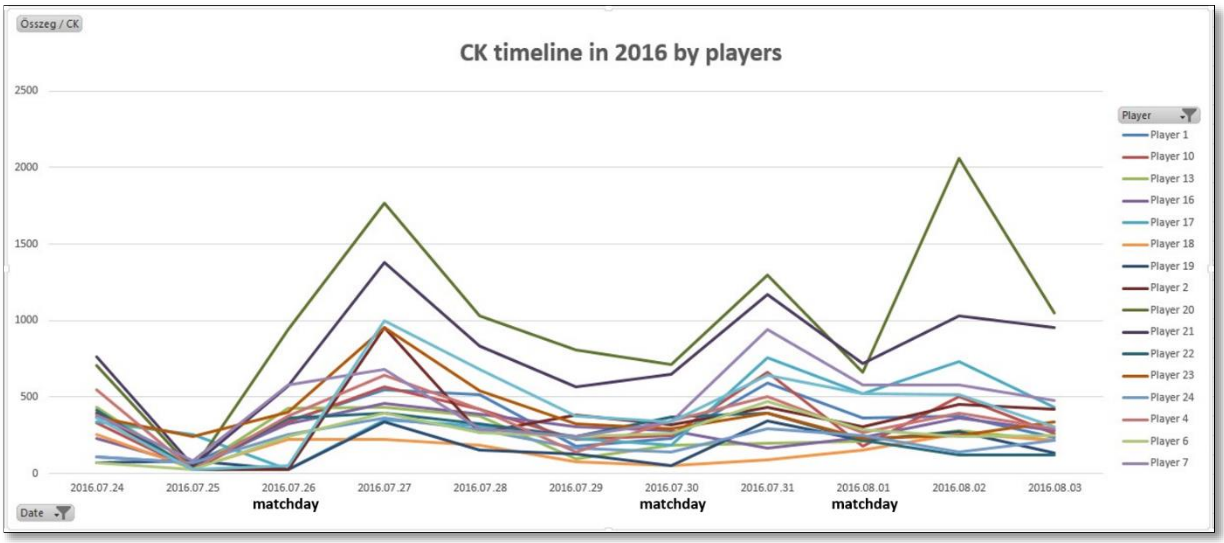


Figure 6 – Time delay in CK elevation (Source: own source)

As we can see, CK level elevated on the next day, which is a key fact for the analysis. Due to this information now we know that all performance data must be paired with the post-match day's CK level as an output, and the match day's CK level as an input (in dataset it is called *CK-1* attribute). The input CK is extremely important not only as we discussed in section 3, but because the player load results in a relative CK value difference, and we need a baseline to correctly predict a new value.

6.3. Barriers of personalization

Although one of the main goals was to create a personalized predictive model, it became clear that due to the lack of unique, player-based data points, it is impossible to create a reliable yet precise model for each player.

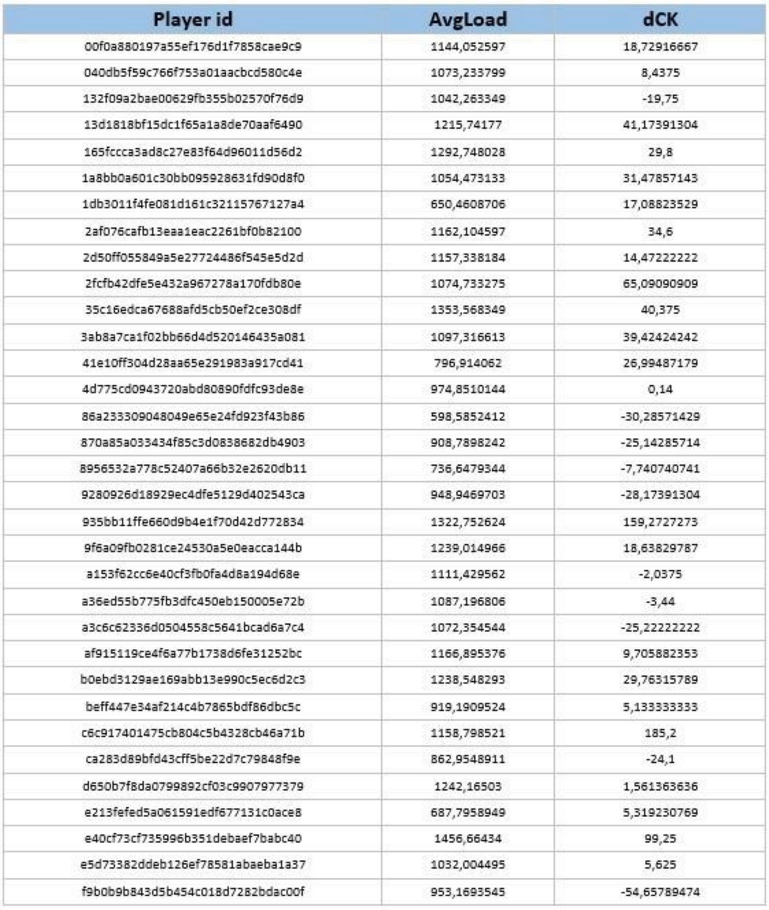
To get around the problem of this problem, I decided to use clustering as a method of personalization. With clustering we can not only get more accurate models for each players than an aggregated model but also can sustain the reliability of the calculations due to the wider datasets.

6.4. Clustering

Another important task was the clustering process. To simplify the large attribute set (756 different columns) I selected the most representative attribute, the *Total Player Load*. I calculated an average value for all the players (*AvgLoad)*, and compared them to the CK values.

CK values needed another transformation to make them useful for the analysis. Because I was looking for regeneration clusters, I had to differentiate the players based on their CK normalization behavior. To make it a numerical value I calculated the difference between post- and pre-match CK values (*dCK*), than averaged these values to all the players.

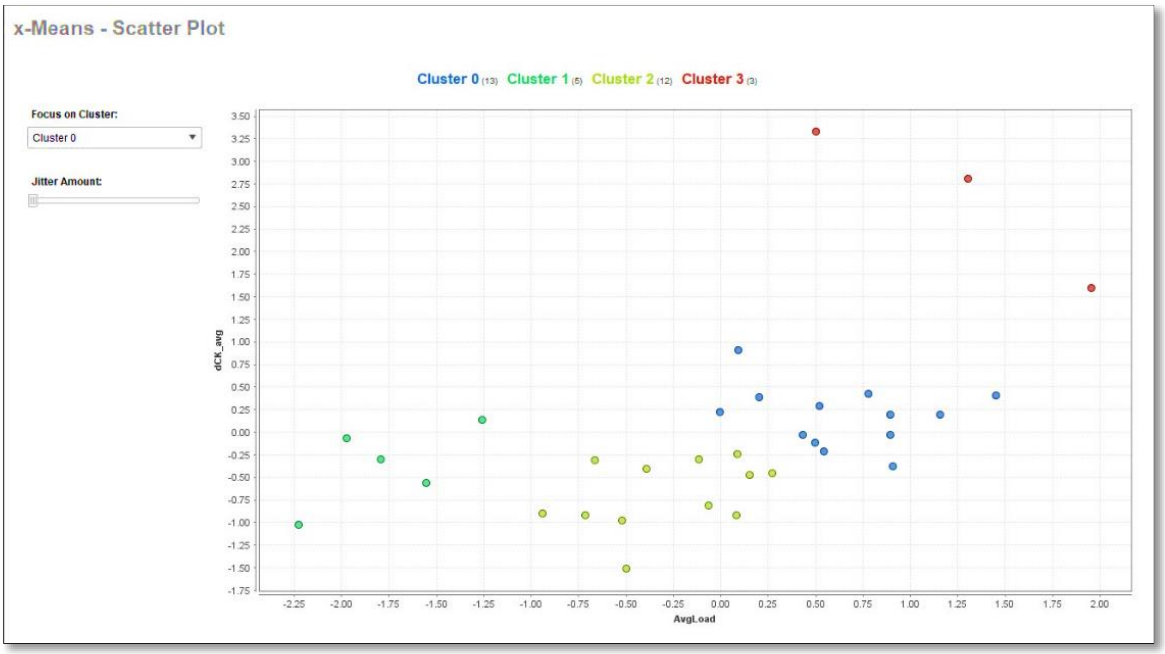
Finally, I joined the two data table to create the base for the clustering (Table 2).



Player id	AvgLoad	dCK
00f0a880197a55ef176d1f7858cae9c9	1144,052597	18,72916667
040db5f59c766f753a01aacbcd580c4e	1073,233799	8,4375
132f09a2bae00629fb355b02570f76d9	1042,263349	-19,75
13d1818bf15dc1f65a1a8de70aaf6490	1215,74177	41,17391304
165fccca3ad8c27e83f64d96011d56d2	1292,748028	29,8
1a8bb0a601c30bb095928631fd90d8f0	1054,473133	31,47857143
1db3011f4fe081d161c32115767127a4	650,4608706	17,08823529
2af076cafb13eaa1eac2261bf0b82100	1162,104597	34,6
2d50ff055849a5e27724486f545e5d2d	1157,338184	14,47222222
2fcfb42dfe5e432a967278a170fdb80e	1074,733275	65,09090909
35c16edca67688afd5cb50ef2ce308df	1353,568349	40,375
3ab8a7ca1f02bb66d4d520146435a081	1097,316613	39,42424242
41e10ff304d28aa65e291983a917cd41	796,914062	26,99487179
4d775cd0943720abd80890fdfc93de8e	974,8510144	0,14
86a233309048049e65e24fd923f43b86	598,5852412	-30,28571429
870a85a033434f85c3d0838682db4903	908,7898242	-25,14285714
8956532a778c52407a66b32e2620db11	736,6479344	-7,740740741
9280926d18929ec4dfe5129d402543ca	948,9469703	-28,17391304
935bb11ffe660d9b4e1f70d42d772834	1322,752624	159,2727273
9f6a09fb0281ce24530a5e0eacca144b	1239,014966	18,63829787
a153f62cc6e40cf3fb0fa4d8a194d68e	1111,429562	-2,0375
a36ed55b775fb3dfc450eb150005e72b	1087,196806	-3,44
a3c6c62336d0504558c5641bcad6a7c4	1072,354544	-25,22222222
af915119ce4f6a77b1738d6fe31252bc	1166,895376	9,705882353
b0ebd3129ae169abb13e990c5ec6d2c3	1238,548293	29,76315789
beff447e34af214c4b7865bdf86dbc5c	919,1909524	5,133333333
c6c917401475cb804c5b4328cb46a71b	1158,798521	185,2
ca283d89bfd43cff5be22d7c79848f9e	862,9548911	-24,1
d650b7f8da0799892cf03c9907977379	1242,16503	1,561363636
e213fefed5a061591edf677131c0ace8	687,7958949	5,319230769
e40cf73cf735996b351debaef7babc40	1456,66434	99,25
e5d73382ddeb126ef78581abaeba1a37	1032,004495	5,625
f9b0b9b843d5b454c018d7282bdac00f	953,1693545	-54,65789474

Table 2 – Dataset for clustering (Source: own source)

In RapidMiner software I executed an optimized x-Means algorithm to determine the ideal number of clusters, the result are shown in Figure 7. X-Means algorithm, as Pelleg and Moore presented in 2000, is a modified k-Means clustering algorithm, which according to Abbas (2008) has the best performance quality compared to other clustering algorithms.



Focus on Cluster: Cluster 0

Jitter Amount:

Figure 7– Clustering the players by dCK and average load (Source: own source)

As Figure 7 shows, there were 4 different cluster based on the selected two attributes. (It is important to note that in Figure 7 the two attributes had been regularized by the software.)

The main differences (Figure 8) are the following:

- cluster 0 contains 13 players with a little bit higher *AvgLoad* with an average *dCK* value
- cluster 1 contains 5 players with the lowest *AvgLoad* with a little bit lower than average *dCK* value
- cluster 2 contains 12 players with a little lower than average *AvgLoad* and the lowest *dCK* value
- cluster 3 contains 3 players with the highest *AvgLoad* and *dCK* value

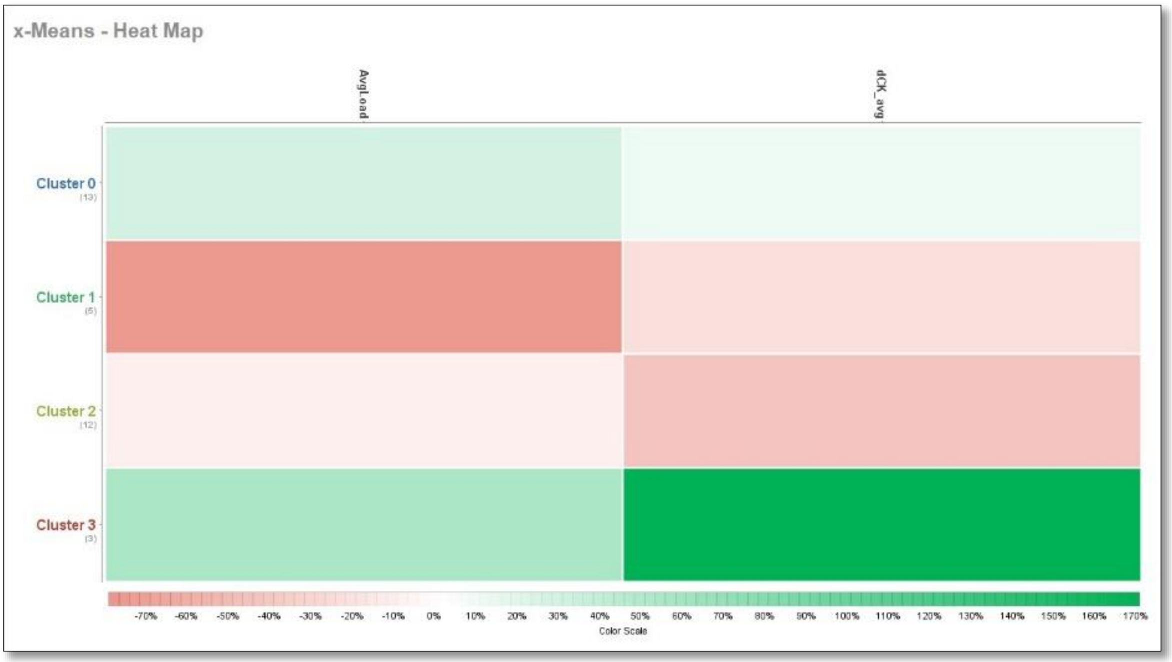


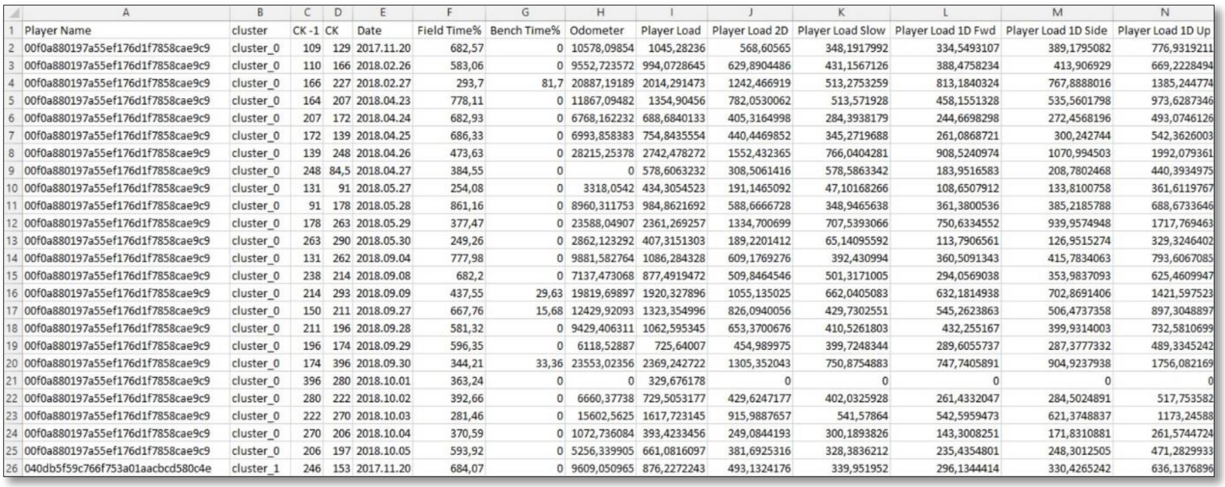
Figure 8 – Heat map of the four clusters (Source: own source)

6.5. The final dataset for data mining

To sum up, for the final dataset I made the following preparation:

- cleaned the two datasets by removing N/A and missing values
- summarized the performance data attributes where it made sense (e.g. average values were left out)
- joined the two datasets by the *CK* values
- paired the *CK* values with their pre-event *CK* values (*CK-1*)
- clustered the players by their body's average CK elevation (*dCK*) reaction to the averaged *Total Player Load (AvgLoad)*
- merged the previously joined datasets with the cluster table

The final dataset (Figure 9) contained 866 rows and 435 columns.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Player Name	cluster	CK -1	CK	Date	Field Time%	Bench Time%	Odometer	Player Load	Player Load 2D	Player Load Slow	Player Load 1D Fwd	Player Load 1D Side	Player Load 1D Up
2	00f0a880197a55ef176d1f7858cae9c9	cluster_0	109	129	2017.11.20	682,57	0	10578,09854	1045,28236	568,60565	348,1917992	334,5493107	389,1795082	776,9319211
3	00f0a880197a55ef176d1f7858cae9c9	cluster_0	110	166	2018.02.26	583,06	0	9552,723572	994,0728645	629,8904486	431,1567126	388,4758234	413,906929	669,2228494
4	00f0a880197a55ef176d1f7858cae9c9	cluster_0	166	227	2018.02.27	293,7	81,7	20887,19189	2014,291473	1242,466919	513,2753259	813,1840324	767,8888016	1385,244774
5	00f0a880197a55ef176d1f7858cae9c9	cluster_0	164	207	2018.04.23	778,11	0	11867,09482	1354,90456	782,0530062	513,571928	458,1551328	535,5601798	973,6287346
6	00f0a880197a55ef176d1f7858cae9c9	cluster_0	207	172	2018.04.24	682,93	0	6768,162232	688,6840133	405,3164998	284,3938179	244,6698298	272,4568196	493,0746126
7	00f0a880197a55ef176d1f7858cae9c9	cluster_0	172	139	2018.04.25	686,33	0	6993,858383	754,8435554	440,4469852	345,2719688	261,0868721	300,242744	542,3626003
8	00f0a880197a55ef176d1f7858cae9c9	cluster_0	139	248	2018.04.26	473,63	0	28215,25378	2742,478272	1552,432365	766,0404281	908,5240974	1070,994503	1992,079361
9	00f0a880197a55ef176d1f7858cae9c9	cluster_0	248	84,5	2018.04.27	384,55	0	0	578,6063232	308,5061416	578,5863342	183,9516583	208,7802468	440,3934975
10	00f0a880197a55ef176d1f7858cae9c9	cluster_0	131	91	2018.05.27	254,08	0	3318,0542	434,3054523	191,1465092	47,10168266	108,6507912	133,8100758	361,6119767
11	00f0a880197a55ef176d1f7858cae9c9	cluster_0	91	178	2018.05.28	861,16	0	8960,311753	984,8621692	588,6666728	348,9465638	361,3800536	385,2185788	688,6733646
12	00f0a880197a55ef176d1f7858cae9c9	cluster_0	178	263	2018.05.29	377,47	0	23588,04907	2361,269257	1334,700699	707,5393066	750,6334552	939,9574948	1717,769463
13	00f0a880197a55ef176d1f7858cae9c9	cluster_0	263	290	2018.05.30	249,26	0	2862,123292	407,3151303	189,2201412	65,14095592	113,7906561	126,9515274	329,3246402
14	00f0a880197a55ef176d1f7858cae9c9	cluster_0	131	262	2018.09.04	777,98	0	9881,582764	1086,284328	609,1769276	392,430994	360,5091343	415,7834063	793,6067085
15	00f0a880197a55ef176d1f7858cae9c9	cluster_0	238	214	2018.09.08	682,2	0	7137,473068	877,4919472	509,8464546	501,3171005	294,0569038	353,9837093	625,4609947
16	00f0a880197a55ef176d1f7858cae9c9	cluster_0	214	293	2018.09.09	437,55	29,63	19819,69897	1920,327896	1055,135025	662,0405083	632,1814938	702,8691406	1421,597523
17	00f0a880197a55ef176d1f7858cae9c9	cluster_0	150	211	2018.09.27	667,76	15,68	12429,92093	1323,354996	826,0940056	429,7302551	545,2623863	506,4737358	897,3048897
18	00f0a880197a55ef176d1f7858cae9c9	cluster_0	211	196	2018.09.28	581,32	0	9429,406311	1062,595345	653,3700676	410,5261803	432,255167	399,9314003	732,5810699
19	00f0a880197a55ef176d1f7858cae9c9	cluster_0	196	174	2018.09.29	596,35	0	6118,52887	725,64007	454,989975	399,7248344	289,6055737	287,3777332	489,3345242
20	00f0a880197a55ef176d1f7858cae9c9	cluster_0	174	396	2018.09.30	344,21	33,36	23553,02356	2369,242722	1305,352043	750,8754883	747,7405891	904,9237938	1756,082169
21	00f0a880197a55ef176d1f7858cae9c9	cluster_0	396	280	2018.10.01	363,24	0	0	329,676178	0	0	0	0	0
22	00f0a880197a55ef176d1f7858cae9c9	cluster_0	280	222	2018.10.02	392,66	0	6660,37738	729,5053177	429,6247177	402,0325928	261,4332047	284,5024891	517,753582
23	00f0a880197a55ef176d1f7858cae9c9	cluster_0	222	270	2018.10.03	281,46	0	15602,5625	1617,723145	915,9887657	541,57864	542,5959473	621,3748837	1173,24588
24	00f0a880197a55ef176d1f7858cae9c9	cluster_0	270	206	2018.10.04	370,59	0	1072,736084	393,4233456	249,0844193	300,1893826	143,3008251	171,8310881	261,5744724
25	00f0a880197a55ef176d1f7858cae9c9	cluster_0	206	197	2018.10.05	593,92	0	5256,339905	661,0816097	381,6925316	328,3836212	235,4354801	248,3012505	471,2829933
26	040db5f59c766f753a01aacbcd580c4e	cluster_1	246	153	2017.11.20	684,07	0	9609,050965	876,2272243	493,1324176	339,951952	296,1344414	330,4265242	636,1376896

Figure 9 – Final dataset with clusters and CK values (Source: own source)

7. Building the predictive model

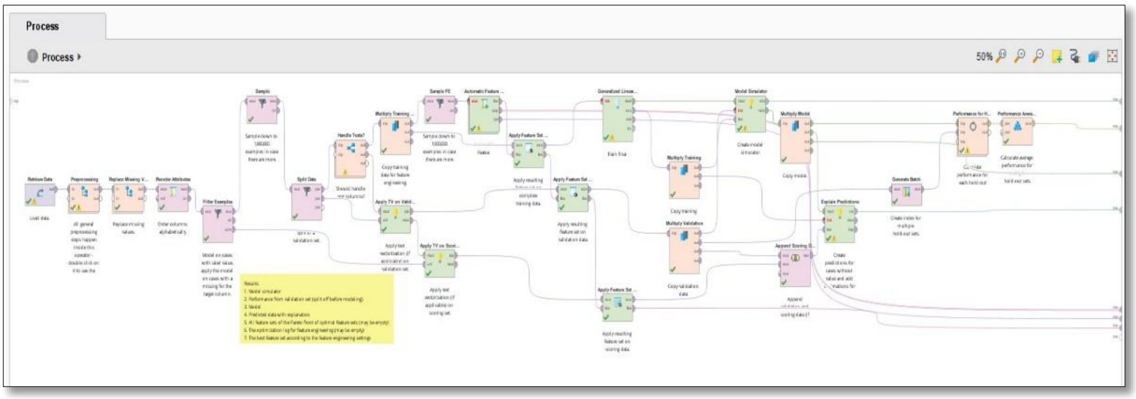
7.1. Modelling process

For the modelling process I decided to use GLM (Generalized Linear Model) as from all predictive models in RapidMiner it had the most consistent result for all the clusters' and had the highest correlation value with the second lowest absolute error value.

Another important fact was that GLM models can be implemented easier in webpages than other predictive methods like Decision Trees or Support Vector Machines.

The whole predictive model building process is shows in Figure 10, with the following steps:

- data reading and sampling
- data splitting for validation
- train data multiplying
- modeling
- automatic feature engineering
- 3-fold cross validation
- calculation simulator building
- performance calculations



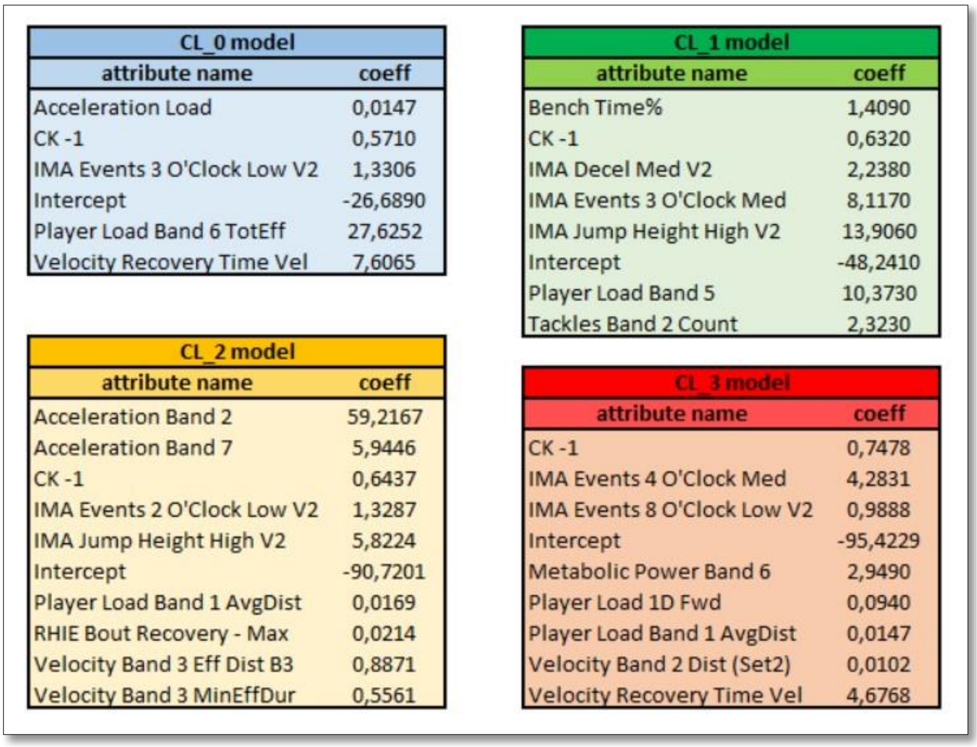
Results:
1. Model simulator
2. Performance from validation set (split off before modeling)
3. Model
4. Predicted data with explanation
5. All feature sets of the Pareto front of optimal feature sets (may be empty)
6. The optimization log for feature engineering (may be empty)
7. The best feature set according to the feature engineering settings

Figure 10 – GLM model building process in RapidMiner (Source: own source)

7.2. Presentation of the final model

With the GLM I was able to finally create the four models for all the clusters. With the built-in feature engineering function the software selected the most important ca. 40 attributes out of the 435 different columns. But for the implementation I needed to narrow down this to a reasonable amount of variables, which an ordinary user with relatively short time can use as well.

Through trials and errors I determined the minimal number of attributes with the best correlation and lowest absolute error for all the clusters (Figure 11).



CL_0 model	
attribute name	**coeff**
Acceleration Load	0,0147
CK -1	0,5710
IMA Events 3 O'Clock Low V2	1,3306
Intercept	-26,6890
Player Load Band 6 TotEff	27,6252
Velocity Recovery Time Vel	7,6065

CL_1 model	
attribute name	**coeff**
Bench Time%	1,4090
CK -1	0,6320
IMA Decel Med V2	2,2380
IMA Events 3 O'Clock Med	8,1170
IMA Jump Height High V2	13,9060
Intercept	-48,2410
Player Load Band 5	10,3730
Tackles Band 2 Count	2,3230

CL_2 model	
attribute name	**coeff**
Acceleration Band 2	59,2167
Acceleration Band 7	5,9446
CK -1	0,6437
IMA Events 2 O'Clock Low V2	1,3287
IMA Jump Height High V2	5,8224
Intercept	-90,7201
Player Load Band 1 AvgDist	0,0169
RHIE Bout Recovery - Max	0,0214
Velocity Band 3 Eff Dist B3	0,8871
Velocity Band 3 MinEffDur	0,5561

CL_3 model	
attribute name	**coeff**
CK -1	0,7478
IMA Events 4 O'Clock Med	4,2831
IMA Events 8 O'Clock Low V2	0,9888
Intercept	-95,4229
Metabolic Power Band 6	2,9490
Player Load 1D Fwd	0,0940
Player Load Band 1 AvgDist	0,0147
Velocity Band 2 Dist (Set2)	0,0102
Velocity Recovery Time Vel	4,6768

Figure 11 – Minimalized models for the four clusters (Source: own source)

In Figure 11, column '*coeff'* (stands for coefficient) shows, how each attributes play different roles in the final predicted value. The meaning of a coefficient is, that if none of the other values were changed, one unit of increment in the attribute value would increase the predicted result with the value of the coefficient.

8. Results of analysis

8.1. Evaluating results through measurement numbers

In Table 3 I summarized the key measurement numbers of the clusters to all minimized predictive models.

Model name	Correlation	Absolute error
GLM for cluster 0	0.784	89 +/- 17.30
GLM for cluster 1	0.758	118 +/- 11.95
GLM for cluster 2	0.772	121 +/- 37.67
GLM for cluster 3	0.848	95 +/- 9.14

Table 3 – Measurement numbers for GLM models

As the table above illustrates, all the models have a correlation higher than 0.75, which means a strong connection between the target value and the attributes (Hunyadi-Vita, 1991, 1992). At the same time it means our model is far from ideal connection (e.g. correlation above 0.9).

To understand absolute error more deeply, we have to examine the analyzed dataset. The result of our model is a predicted CK value, so during the comparison we need to focus on the distribution and the scale of the original CK data. As Figure 3 shows, CK values are distributed between 50 and 1600, which means an approximately 1500 interval for all values. So absolute error values in the model should be compared to this interval, which leads us to errors around 6-8 percent, which is, considering the size of our dataset is acceptable for a prediction.

8.2. Understanding the predictive model

In the light of the evaluation process it can be concluded that the presented four models are acceptable for a prediction model.

We know that the main purpose of the predictive model is a real-life train planning, therefore it only used for approximate estimation.

All things considered we can use our presented models to implement the required webpage application, as long as the user aware of the model’s error.

8.3. Answering the first question of the study

In section 4.1 I declared the main questions of the study. The first question was:

‘*How can we predict young players’ future CK level based on their physical performance, and is personalization possible?*

Examining the results, it is clear that one way to create a personalized predictive model for young players’ future CK level can be implemented by regenerations cluster-based GLM models with selected key attributes, shown in Figure 11.

9. Deployment – the CK Calculator

9.1. Goals and UX requirements of the webpage

While my research is heavily relied on data mining processes, I had another important task, stated as the second main question of the study:

How can this prediction help the players to reach their optimal performance?

Although there are limitless possibilities to answer this question, as a business informatics student I was looking for a solution that can be deployed to real-life users. All things considered I chose to create a webpage, with the following requirements:

- the webpage should be easy-to-use for users with no IT knowledge
- the webpage should help the coach’s work with visual UX functions such as colors and optimal intervals
- the webpage should predict the future CK level based on selected factors real-time
- the webpage must be able to handle different player types (clusters)
- the webpage should be a lightweight solution with minimal server-client communication
- the webpage should use the newest front-end technologies (JavaScript 1.7, JQuery3, HTML5, CSS3)
- the webpage should be secure due to data protection restrictions

Before I could create the solution, I had to solve a crucial part of the tasks first, the problem of optimal intervals.

9.2. Personalized optimal CK range

As can be seen, CK level indicates the body's reaction to physical work, however it works in the other way around as well, as current CK level can impact on next day's possible physical activity. For instance there are several players whose performance are superb under higher CK level. Furthermore the optimal starting CK value for best KPI numbers naturally varies amongst all players.

To make the solution more effective and helpful, it was crucial to determine each players' optimal CK range.

Even so it seemed like an ordinary task to perform, it turned out to be a hard and rather challenging problem.

As Figure 12 shows, CK values against *Load* are widely distributed all the way around. In Figure 13, I highlighted two different groups of data points.

The red group contains events, where the outstanding performances happened. However, the green groups contain runner-up events, several times with totally different CK ranges. Moreover, there are non-highlighted areas in the same CK range, with average or poor performance.

To make the analysis even more complex, there were players with far less data points than the presented four, and players with multiple peak performances in different ranges. After several attempts, I gave up on creating a mathematical solution, and performed the range determination manually. To be consistent, I selected the best 5-10 performances close to each other, and chose the CK range to not to be wider than 200 unit.

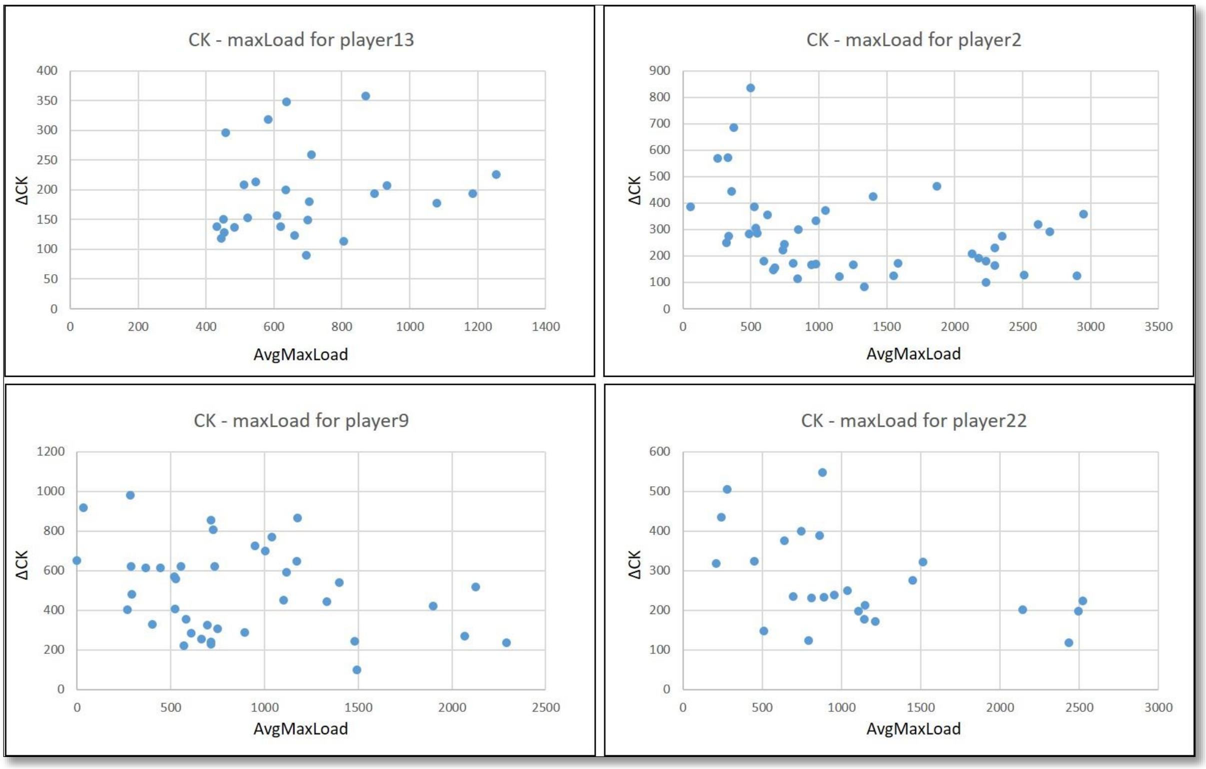


Figure 12 – Example for CK against Load value-pairs (Source: own source)

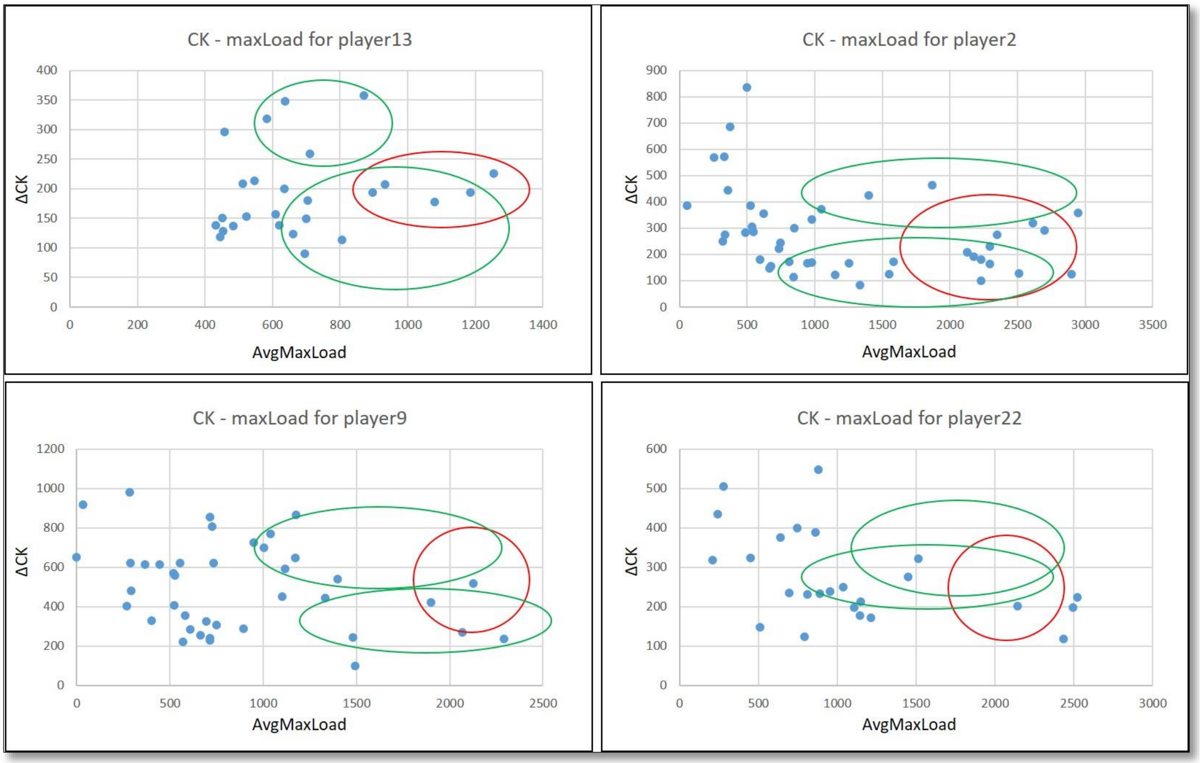


Figure 13 – Highlighted data points in CK-Load value-pairs (Source: own source)

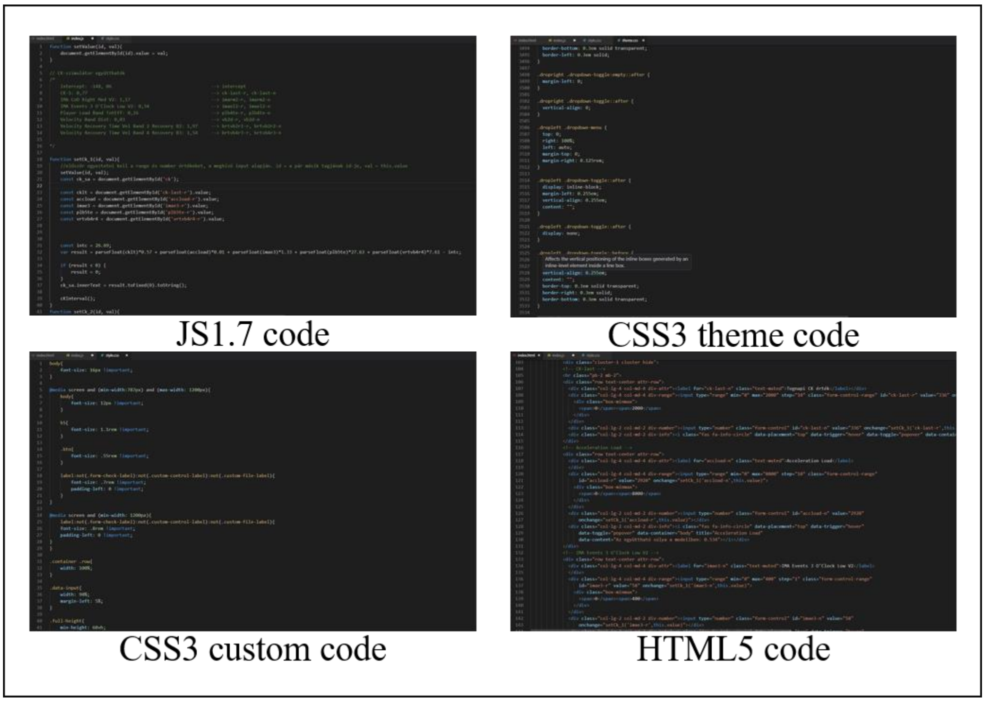
9.3. Front-end development

For the webpage I used the most recent and most used front-end technologies, as one of the requirements stated to do so, HTML5, JS1.7 and CSS3 (Figure 14).

Figure 14 – The front-end triangle (Source: https://www.toughlex.com/technologies/front-end)

The front-end triangle is a vastly evolving group of programming languages that provides a multi-platform opportunity for web development.

Firstly, the HTML gives the webpage a structure, basically its main function to create a space for all the content. Secondly, CSS is responsible for styling, its purpose to make the user-experience smoother, more enjoyable and understandable. Thirdly, JavaScript is responsible for all the behavioral part of the webpage. It contains the business logic, in our case it calculates the final prediction for the given user input, based on the GLM models. Selected chunks of my code are provided in Figure 15.


JS1.7 code

CSS3 theme code

CSS3 custom code

HTML5 code

Figure 15 – Examples of developer code for the webpage (Source: own source)

There were two other important technological parts for the development, both of them were third-party libraries. One of them was JQuery 3.3.1, a JavaScript library for event handling, animation, DOM manipulation and document traversal, which I used through an Ajax API. The other one was Bootstrap 4.0, a front-end component library, which I mostly used for building the HTML and CSS sections, through BootstrapCDN.

During my work on the webpage, I wanted to create a simple and clear website for the user, where all information are available simultaneously, and easy to handle.

Therefore I chose to develop a modular page, where all modules, responsible for different tasks, can be visually distinguished.

I could define four distinct user interactions or goals, which determined the purpose of each modules:

- the selection of a certain player (module 1)
- the input of the needed performance values (module 2)
- the results of the calculations (module 3)
- further navigational or optional user interactions e.g. resetting values (module 4)

To illustrate, how the modules communicate and react to each other real-time, I summarized the process in Figure 16.

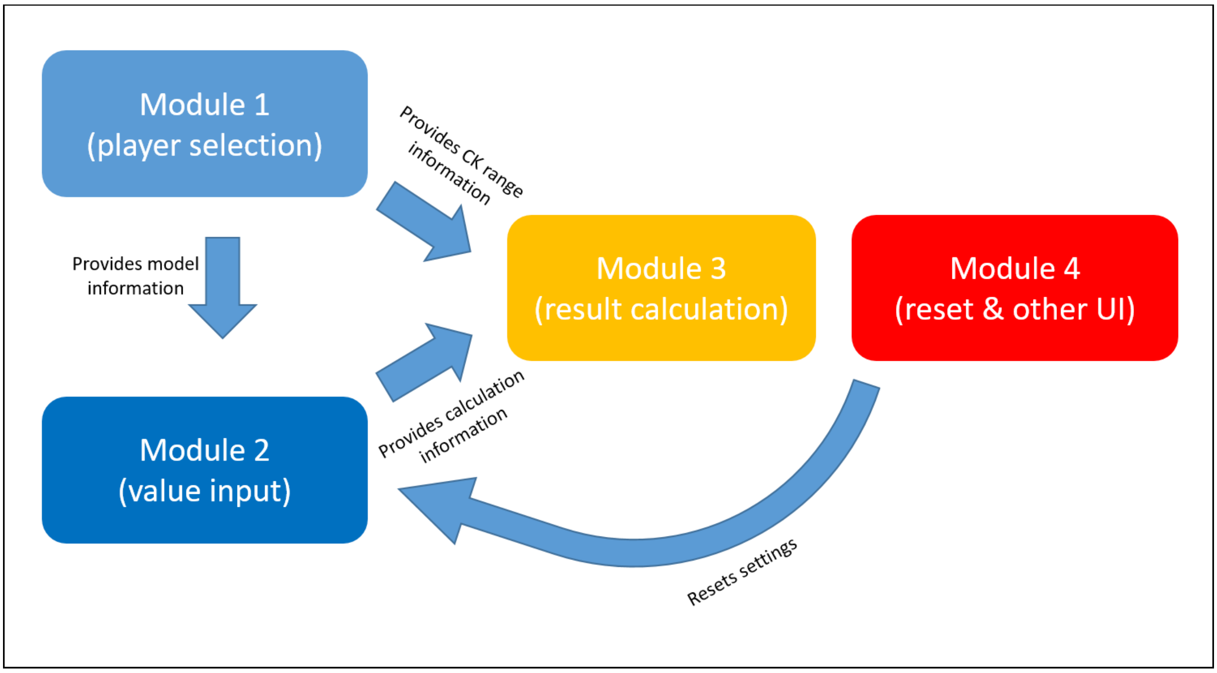


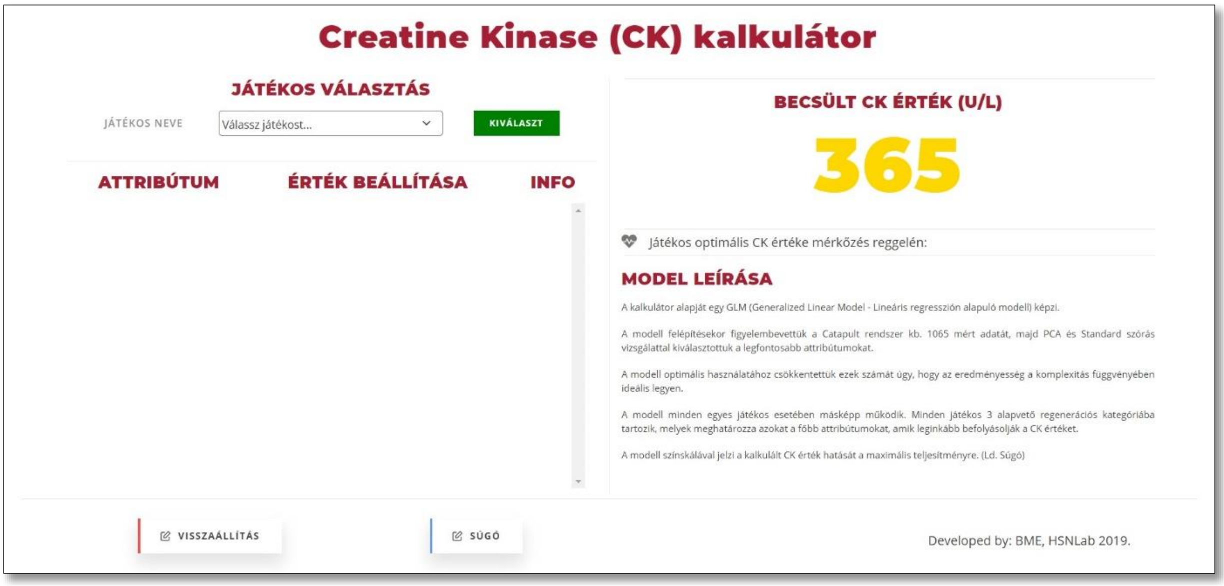
Figure 16 – Relationship between modules (Source: own source)

9.4. Final webpage

In Figure 17 I present the final structure of the webpage.

Correspondingly to section 8.3, all the different parts of the UI (user interface) are equivalent to one of the modules in Figure 16, with the following pairing:

- UI part labelled as *'Játékos választás'* represents Module 1 (player selection)
- UI part with three labels (*'Attribútum', 'Érték beállítása', 'Info')* represents Module 2 (model value input)
- UI part labelled as *'Becsült CK érték'* and *'Játékos optimális CK értéke mérkőzés elején'* represents module 3 (results of calculations)
- UI part with buttons and label *'Model leírása'* represents module 4 (resetting and other user interactions)



Creatine Kinase (CK) kalkulátor

JÁTÉKOS VÁLASZTÁS

JÁTÉKOS NEVE: Válassz játékost...

KIVÁLASZT

ATTRIBÚTUM | ÉRTÉK BEÁLLÍTÁSA | INFO

BECSÜLT CK ÉRTÉK (U/L)

365

Játékos optimális CK értéke mérkőzés reggelén:

MODEL LEÍRÁSA

A kalkulátor alapját egy GLM (Generalized Linear Model - Lineáris regresszión alapuló modell) képzi.

A modell felépítésekor figyelembevettük a Catapult rendszer kb. 1065 mért adatát, majd PCA és Standard szórás vizsgálattal kiválasztottuk a legfontosabb attribútumokat.

A modell optimális használatához csökkentettük ezek számát úgy, hogy az eredményesség a komplexitás függvényében ideális legyen.

A modell minden egyes játékos esetében másképp működik. Minden játékos 3 alapvető regenerációs kategóriába tartozik, melyek meghatározza azokat a főbb attribútumokat, amik leginkább befolyásolják a CK értéket.

A modell színskálával jelzi a kalkulált CK érték hatását a maximális teljesítményre. (Ld. Súgó)

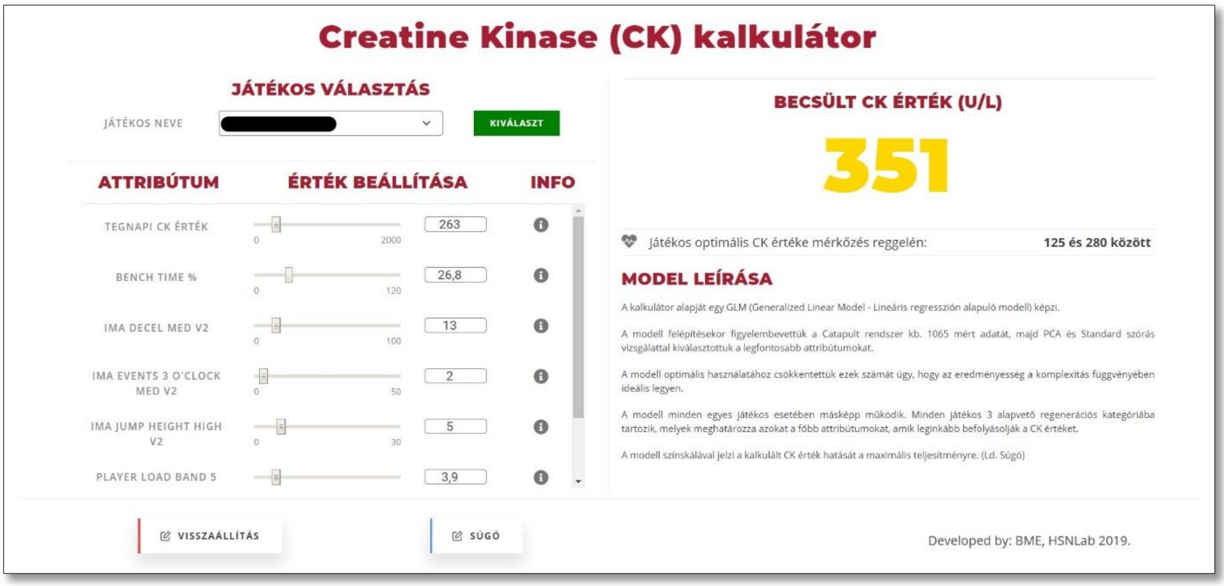
VISSZAÁLLÍTÁS

SÚGÓ

Developed by: BME, HSNLab 2019.

Figure 17 – The final landing page (Source: own source)

In Figure 18 we can see how the webpage works during a normal process. An anonymized player had been selected for calculation in module 1, therefore the cluster related values appeared in module 2. The user can set up easily the required values with the slider or numerical input fields. In module 3, the predicted value reacts to all user changes, showing not only the result but the optimal CK interval as well. Nonetheless it helps the user with color schemes (green – optimal, yellow – acceptable, red – inadequate).



Creatine Kinase (CK) kalkulátor

JÁTÉKOS VÁLASZTÁS

JÁTÉKOS NEVE [redacted]

ATTRIBÚTUM	ÉRTÉK BEÁLLÍTÁSA			INFO
TEGNAPI CK ÉRTÉK	0	2000	263	ℹ
BENCH TIME %	0	120	26,8	ℹ
IMA DECEL MED V2	0	100	13	ℹ
IMA EVENTS 3 O'CLOCK MED V2	0	50	2	ℹ
IMA JUMP HEIGHT HIGH V2	0	30	5	ℹ
PLAYER LOAD BAND 5			3,9	ℹ

BECSÜLT CK ÉRTÉK (U/L)

351

Játékos optimális CK értéke mérkőzés reggelén: **125 és 280 között**

MODEL LEÍRÁSA

A kalkulátor alapját egy GLM (Generalized Linear Model - Lineáris regresszión alapuló modell) képzi.

A modell felépítésekor figyelembevettük a Catapult rendszer kb. 1065 mért adatát, majd PCA és Standard szórás vizsgálattal kiválasztottuk a legfontosabb attribútumokat.

A modell optimális használatához csökkentettük ezek számát úgy, hogy az eredményesség a komplexitás függvényében ideális legyen.

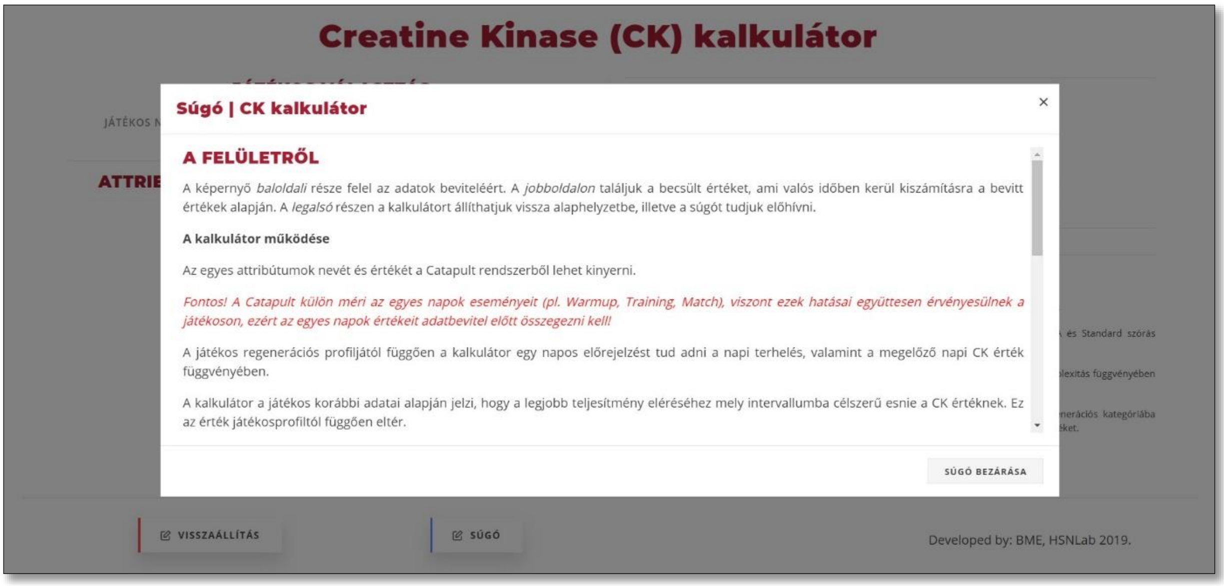
A modell minden egyes játékos esetében másképp működik. Minden játékos 3 alapvető regenerációs kategóriába tartozik, melyek meghatározza azokat a főbb attribútumokat, amik leginkább befolyásolják a CK értéket.

A modell színskálával jelzi a kalkulált CK érték hatását a maximális teljesítményre. (Ld. Súgó)

Developed by: BME, HSNLab 2019.

Figure 18 – Webpage in work (Source: own source)

Focusing on the usability, the website has a built-in '*Help'* section, describing the use cases and containing user instructions for all the modules (Figure 19).



Creatine Kinase (CK) kalkulátor

Súgó | CK kalkulátor

A FELÜLETRŐL

A képernyő *baloldali* része felel az adatok beviteléért. A *jobboldalon* találjuk a becsült értéket, ami valós időben kerül kiszámításra a bevitt értékek alapján. A *legalsó* részen a kalkulátort állíthatjuk vissza alaphelyzetbe, illetve a súgót tudjuk előhívni.

A kalkulátor működése

Az egyes attribútumok nevét és értékét a Catapult rendszerből lehet kinyerni.

Fontos! A Catapult külön méri az egyes napok eseményeit (pl. Warmup, Training, Match), viszont ezek hatásai együttesen érvényesülnek a játékoson, ezért az egyes napok értékeit adatbevitel előtt összegezni kell!

A játékos regenerációs profiljától függően a kalkulátor egy napos előrejelzést tud adni a napi terhelés, valamint a megelőző napi CK érték függvényében.

A kalkulátor a játékos korábbi adatai alapján jelzi, hogy a legjobb teljesítmény eléréséhez mely intervallumba célszerű esnie a CK értéknek. Ez az érték játékosprofiltól függően eltér.

Developed by: BME, HSNLab 2019.

Figure 19 – Built-in help section in the webpage (Source: own source)

9.5. Evaluating the deployment

In spite of the fact, that feedbacks from users and coworkers found the website useful and practical, it is always an indispensable step to review and evaluate the final result at the end of the process. Therefore I compared the final website to the original requirements (declared in section 8.1) in Table 4.

Requirement	Solution
the webpage should be easy-to-use for users with no IT knowledge	Help section with thorough descriptions, modular structure, easy-to-understand labels
the webpage should help the coach's work with visual UX functions such as colors and optimal intervals	Color schemes and calculated intervals
the webpage should predict the future CK level based on selected factors real-time	Real-time prediction with cluster-based attributes
the webpage must be able to handle different player types (clusters)	JS code and website structure handles clusters, easy player selection
the webpage should be a lightweight solution with minimal server-client communication	almost zero server-client communication, resetting solved without need of page reload
the webpage should use the newest front-end technologies (JavaScript 1.7, JQuery3, HTML5, CSS3)	Used the most recent technologies, supplemented with JQuery and Bootstrap
the webpage should be secure due to data protection restrictions	all sensitive data is hidden between a strongly protected university server with secure sign-in process

Table 4 – Evaluating website through the original requirements (Source: own source)

As described, the webpage meets the most important given criteria, therefore we can accept the solution for deployment.

10. Further possibilities in the research

10.1. Ways to improve the model

In section 7.1 I concluded that the built-up models in light of the selected measurement numbers and the size of the given datasets are acceptable. Having said that, there are a number of factors and possibilities for improvement.

Firstly, a crucial problem with CK values, that human body reaction to physical activity varies among all people (e.g. in magnitude, minimum and maximum levels, range, etc.), even in athletes from the same age and sport. To resolve the problem of personalization, way more data points are needed from every players. With these, it would be possible to find a better way for finer predictions that clusters.

Secondly, the problem with the meaning of CK elevation itself. In section 2.1 I presented how CK serum level elevation is a response from the body to muscular damage. According to Baird et al. (2012), there are other factors that can undermine the validity of the meaning of CK value, most importantly sport injuries. To resolve this problem, a profound description of all players' medical history targeting injuries specifically, and registered physical contacts during trainings and matches could also improve significantly the results.

10.2. Ways to improve the webpage

In the same way, there are plenty of options to develop the website further. Although it meets the most important criteria for every-day functioning as well as it is already in use, yet it is clear that improvement can be done both in technological and UX (user experience) ways.

Speaking of technology, it is important to note that the current website only uses front-end technologies, there is no back-end behind it. It means all the business logic and the data are 'hard-coded' in, it cannot actually react to any kind of new input, whether it is data, model, cluster or any other development. All players and all their calculated optimal CK range, along with the clusters are pre-determined, meaning it is impossible to dynamically change these settings on-the-run.

However, if back-end solutions were provided, all these matters could be solved with options like administrator settings, dynamic databases and real-time data-fetching e.g. through an API. Moreover, it would allow the user to not only work in real-time, but also save, import or export calculations and results about certain players. It is similarly the field of UX improvements, which implicates that most of the improvements are both technology and UX related.

All things considered, the website already works and completes the current user demands, however there are several ways to make the solution more complex and useful in the future.

11. Summary

In this research I focused on implementing a data analysis and data mining process called CK level analysis. I followed the CRISP-DM method throughout the whole study, with the following steps:

Business understanding

I observed the background of the data, to understand the circumstances and the environment for the analysis. I presented the meaning of Creatine-Kinase, the most-preferred measurement method for muscular damage, and the Catapult system which provided the match and training related performance data.

Data understanding

I examined the given data's structure and stability to define a possible data format for the analysis and to understand the real meaning behind the numbers. One of the most important steps was the realization of time shifts between recorded performance and the corresponding CK level elevation.

Data preparation

I needed to perform all the data preparations to sort out meaningless attributes, missing values and wrongly exported data formats. To create the desired final dataset, after the cleansing I had to join the different data sources. Thanks to the recognized time shift this process provided valid data rows that described real-life events.

I had to deal with the demand of personalization, which was a quite difficult task to perform, due to the small size of data points per player. I managed to create a work-around by using a clustering algorithm called x-Means, which created four, one by one homogeneous groups, based on players' CK elevation reaction to training load.

Modeling and Evaluating

I built the predictive models for each clusters via RapidMiner, using Generalized Linear Model method, which was one of the most accurate and easily implemented models, provided by the analytical software. After the creation of the models, I evaluated the result through absolute error and correlation, two popular and well-known measurement number. Here

Deployment

Due to user demand, I implemented the models in a form of a website, called '*CK calculator*', which enables the user to select a certain player and, based on the player's regeneration cluster, input the values for the most important attributes to accurately predict the player's future CK level. The website also provides the optimal CK range for best performance for each player to make the service useful for training planning. The site contains a help guide and operates with color schemes for better user experience.

Further possibilities

I observed the possibilities of further improvements. I found that both in the modelling process and the implementing phase can be improved with further work, however the website already meets the most important criteria for real-life usability.

Answering the questions of the research

The two main questions of this research were:

1. *How can we predict young players' future CK level based on their physical performance, and is personalization possible?*
2. *How can this prediction help the players to reach their optimal performance?*

Given the above study, I concluded the following answers:

1. One way to predict future CK level is to create a predictive model. To simultaneously achieve personalization, creating regeneration clusters for the players, and building a GLM-based model is a possible solution for the task.
2. Prediction alone provides an insight of possible training plans for coaches. With a CK calculator that not only predicts future CK level, but also implicates the optimal CK range for each player, training planning becomes an easier and more insightful process for coaches.

12. Acknowledgements

I would like thank to Alija Pašić PhD, László Toka PhD, Réka Vass PhD and Péter Revisnyei for their various help during my Project Laboratory work.

13. References and figures

13.1. References

Abbas, A. O. (2008). *Compararison Between Data Clustering Algorithms*. The International Arab Journal of Information Technology, 2008. Vol 5, No 3. pp. 320-325.

Baird, M. F. – Graham, S. M. – Baker, S. J., Bickerstaff, G. F. (2012). *Creatine-Kinase- and Exercise-Related Muscle Damage Implications for Muscle Performance and Recovery*. Journal of Nutrition and Metabolism, Vol. 2012, pp. 13.

Borresen J. Lambert MI (2009). *The quantification of training load the training response and the effect on performance*. Sports Med 2009; 39: pp.779–795.

Cardioc.eu. (2019). *A Catapult rendszer felépítése – Cardio Consulting Hungary Kft. [online] Available at: http://www.cardioc.eu/catapult-arendszerfelepitese/ [Downloaded 4 May 2019]*

Coelho. D. B. – Morandi. R. F. – de Melo. M. A. A.. Silami-Garcia. E. (2010). *Creatine kinase kinetics in professional soccer players during a competitive season.* Rev Bras Cineantropom Desempenho Hum 2011. 13(3): pp. 189-94.

Ehlers. G. G. – Ball. E. T. – Liston. L. (2002). *Creatine Kinase Levels are Elevated During 2-A-Day Practices in Collegiate Football Players.* Journal of Athletic Training. 2002 Apr-Jun. 37: pp. 151-156.

Epstein. Y. (1995). *Clinical significance of serum creatine phosphokinase activity levels following exercise.* Israeli Journal of Medical Science. 1995. pp.698-699.

Fried. G. – Mumcu. C (2017). *Sport Analytics. A data-driven approach to sport business and management.* Routledge. New York. 2017.

Hunyadi László – Vita László (1991). Statisztika I. Aula Kiadó, Budapest, 1991.

Hunyadi László – Vita László (1992). Statisztika II. Aula Kiadó, Budapest, 1992.

Jain, A. K. (2010). Data Clustering: *50 years Beyond K-means. Pattern Recognition Letters,* 2010., Vol 31, Issue 8, pp. 651-666.

Kohavi, R. (1995). *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. In: International Joint Conference on Artificial Intelligence, 1995, August. Vol. 14. No.2, pp.1137-1145.

Liu, Y. – Li, Z. – Xiong, H. – Gao, X., Wu, J. (2010). *Understanding of Internal Clustering Validation Measures*. IEEE Conference on Data Mining 2010., pp. 911-916.

Madhulatha, T. S. (2012*). An Overview on Clustering Methods*. IOSR Journal of Engineering. April 2012, Vol. 2(4), pp. 719-725.

Milligan, W. G. – Cooper, M. C. (1987). *Methodology Review: Clustering Methods. Applied Psychological Measuremen*t 1987, pp. 329-354.

Mougios. V. (2007). *Reference intervals for serum creatine kinase in athletes.* British Journal of Sports Medicine. 2007 Oct. 41: pp.647-678.

Naranjo. J. – De la Cruz. B. – Sarabla. E. – De Hoyo. M. – Dominguez-Cobo. S. (2015). *Heart Rate Variability: a Follow-up in Elite Soccer Players Throughout the Season*. Georg Thieme Verlag KG. Stuttgart. 2015.

Pelleg, D., – Moore, A. W. (2000). *X-means: Extending K-means with Efficient Estimation of the Number of Clusters.* Proceedings of the 17th International Conference on Machine Learning, 2000, pp.727-734.

The Guardian (2019). *Marcelo Bielsa admits Leeds have spied on every opponent this season. https://www.theguardian.com/football/2019/jan/16/marcelo-bielsa-leeds-spied-every-opponent [Downloaded: 9 May 2019]*

Wirth. R. – Hipp. J. (2000). *CRISP-DM: Towards a Standard Process Model for Data Mining. In:* Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining. April. 2000. pp29-39.

Witten, I. H. – Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kauffman, San Francisco, CA.

13.2. List of figures and tables

Figures

Tables