Mobile network data

authors: *Dominika Rzepka (271301), Mateusz Guściora (228884)*

subject: *Modelling and Analysis of Web-based Systems [Lab. Tuesday 15:15]*

Raport Structure:

1. General information, Project pipeline, Program structure
2. First look on data
3. Loading data (loading and merge)
4. Cleaning data (handling duplicates, missings and outliers,)
5. Analysis of data (tables, charts, description etc…)
6. Scaling data
7. Predictions, evaluation
8. Experiments
9. Conclusions

General information

Dataset:

The collection of the data (six csv files) were grouped to two types: downloads and uploads. Feature concerning the name of the mobile network provider from file name was added to each loaded data frame file.

Then, the new dataset was created merging above two datasets named: merged. Feature concerning the name of the dataset origin - download/upload

Environment:

Python 3.9, libraries: pandas, numpy, sklearn, matplotlib, seaborn lib.

Visual studio code, excel and csv extensions

Excel, Orange tool

Project pipeline

First look on the data → Loading datasets into program and merging into new → Preprocessing - cleaning part → Analyzing remaining data → Preprocessing - scaling/standardizing/normalizing → Prediction (building a model and evaluating) → Experiments → Formalize Conclusions.

Program structure

Program written in Python is divided into modules that are run through the 'main.py' file. This file runs consecutively files put into the src folder, that is 'loading', 'cleaning', 'analyzing', 'prediction'. Each module is a functionality running after the previous one. Data derived from functionalities that can benefit in analysis and the output of this analysis (charts, tables, minimum values, maximum values, averages, medians..) is saved in the 'evaluation' folder. Program structure is represented below.

- Program - "mobile data loading-cleaning-analyzing"
    - Campaign3
    - main.py
    - src folder

        - __init__.py
        - ex2_loading.py
        - ex2_cleaning.py
        - ex2_analyzing.py
        - ex2_prediction
    - analysis folder
        - charts, tables, minimum values, maximum values, averages, medians…
    - evaluation folder
        - models, evaluations
    - readMe, requirements, .gitignore

First look on data:

a) All of these datasets are for capturing data related to downloading and uploading data from/to the network for different providers of mobile network services as o2, vodafone and telekom.

b) Downloads set features

- chipsettime: a timestamp of when the measurement was taken, in Unix time format (number of seconds since January 1, 1970)
- cellid: a unique identifier for the cellular base station that the device was connected
- mcs0: modulation and coding scheme 0
- mcs1: modulation and coding scheme 1
- mcsindex: index of modulation and coding scheme used
- tbs0: transport block size on primary carrier
- tbs1: transport block size on secondary carrier
- mimo: multiple-input, multiple-output (MIMO) technology used
- rnti: radio network temporary identifier used to identify the device in the cellular network
- **throughput:** measured throughput in megabits per second
- rb0: number of resource blocks allocated on primary carrier
- rb1: number of resource blocks allocated to another carrier
- **tp_cleaned**: measured throughput in megabits per second, with some data cleaning applied
- scc: secondary cell configuration used
- caindex: carrier aggregation index
- gpstime: timestamp of when the measurement was taken, in GPS time format (number of seconds since January 6, 1980)
- longitude: longitude coordinate of the device's location during the measurement
- latitude: latitude coordinate of the device's location during the measurement
- speed: speed of the device, in meters per second
- rsrq: reference signal received quality in decibels
- rsrp: reference signal received power in decibels
- rssi: received signal strength indicator in decibels
- earfcn: E-UTRA Absolute Radio Frequency Channel Number
- cqi: channel quality indicator, used to measure the quality of the radio link

c) Uploads set features:

- qualitytimestamp: a Unix timestamp representing the time at which the measurement was taken
- cellid: identifier for the cell tower providing the connection
- mcsindex: an index indicating the modulation and coding scheme used for transmission
- tbs: transport block size, a measure of the amount of data transmitted in a given time period
- rbs: resource block size, a measure of the bandwidth used for transmission
- **tp_cleaned**: the throughput, or amount of data successfully transmitted, after some filtering or processing
- gpstime: a Unix timestamp representing the time at which the measurement was taken, in GPS time
- longitude: the longitude of the device or tower providing the connection
- latitude: the latitude of the device or tower providing the connection
- speed: the speed of the device or tower providing the connection, in meters per second
- rsrq: reference signal received quality, a measure of the quality of the received signal
- rsrp: reference signal received power, a measure of the strength of the received signal
- rssi: received signal strength indicator, another measure of signal strength
- earfcn: the frequency band used for transmission, in E-UTRA Absolute Radio Frequency Channel Number
- cqi: channel quality indicator, a measure of the quality of the channel used for transmission.

Upload datasets have similar features and therefore data but with some differences. Has fewer features describing data. Features ‘ throughput is measured in Mbps in the upload dataset, while tp_cleaned is measured in bits/s in the download dataset.
In both datasets features that are similar are:
{cellid, gpstime, longitude, latitude, speed, rsrq, rsrp, rssi, earfcn, cqi}

d) Takeaway:
Features that are included in each dataset are different because they are capturing different aspects of the network performance. Uploading data to the network and downloading data from the network. For example, features related to modulation scheme, transport block size, and throughput may be more relevant to data uploaded to the network, while features related to MIMO transmission, resource allocation, and carrier aggregation may be more relevant to data downloaded from the network.

e) Observations:
After the first look at the data, some dependencies were noticed.
There is a correlation between the first column in download (chipsettime) and upload (qualitytimestamp). The data are closed to each other, which means it could be the same and can be merged in next steps. There is also a correlation between ‘mcsindex’ and ‘mcs0’. Each index has its word representation. For index 1 ‘mcs0’ equals ‘QPSK’, for 2 it is ‘16QAM’, and for index 3 ‘msc0’ is ‘64QAM’. In most cases ‘msc1’ has the same correlation, but there are cases when it is not the same, so it can't be taken for sure.

Another correlation was noticed between ‘throughput’ and ‘tp_cleaned’. In the downloads dataset, they are almost the same. The difference is marked by ‘caindex’ equals 0 (the same) and 1 (different). The equation for calculating ‘tp_cleaned’ in this case is (tbs0 + tbs1) / 1000 which also gives throughput in most of the cases. With the upload it is tbs / 1000 which leads to another conclusion, that tbs = tbs0 + tbs1.

The last correlation was with the columns: ‘rb0’ and ‘rb1’. Both are exactly the same. Which means that one of them can be removed in next steps.

Loading data and merging data:

In ‘Download’ data, information about mobile network provider was added as a feature (taken from file name) and loaded data frames were concatenated into one data frame for downloads. The same operation was applied to ‘Upload’ data files.

Creation of new dataset was performed in the loading. Based on observations, the merged dataset was created. For this step, some changes must be made:

- all data that were in both sets was taken, which means columns: {cellid, mcsindex, tp_cleaned, gpstime, longitude, latitude, speed, rsrq, rsrp ,rssi, earfcn, cqi}
- the information about upload and download data was added to correctly detects previous datasets
- values of ‘tbs0’ and ‘tbs1’ from download were added together to create ‘tbs’ column and added to data frames
- ‘rb0’ from download dataset was renamed to ‘rbs’ to match ‘rbs’ from upload dataset, then added to data frames
- ‘mcs’ was created for upload data based on the conclusions above. Then ‘mcs0’ was renamed to ‘mcs’ to match datasets and both were added to data frames
- ‘qualitytimestamp’ was renamed to ‘chipsettime’ and both were added to data frames
- both new data frames were merged together to create new dataset for next steps

Cleaning data:

- Duplicates:

The number of duplicates was checked and after consideration, it was decided to drop all duplicates. Number of duplicates for ‘download’ data is *3224* instances, which is *1,26%* of all the data. Number of duplicates for ‘upload’ data is *166* instances, which is *0,06%* of all the data. Additionally, the number of duplicates for merged data was provided which is *3390* instances, that equals *0,62%* of all the data. Based on small amount of duplicates, it was deleted from data frames.

- Missing data:

For all the datasets described above, existence of missing data was checked. Due to lack of missing data in both data frames, no actions were taken.

- Outliers:

For detecting outliers, the Z score was provided. The percentage was small, between, almost 0% (upload) to 1% (download), so no action was taken. All the outliers will be normalized in next steps.

Analysing data

Analysis with visualization were conducted on data frames created in section ‘loading data’ and ‘cleaning data’. New data sets are ‘downloads’, ‘uploads’ and ‘merged’. Analysis was visualized and based on heat maps to investigate correlation, histograms to investigate distribution and on boxplots to investigate outliers and mean. In this section it is discussed first about ‘download’ and ‘upload’ results of analysis and then about ‘merge’ results. Next analysis of basic statistical calculations was conducted. At the end of the section analysis of network providers was conducted on ‘merge’ dataset.

- Correlation analysis for ‘upload’ and ‘ download’ dataset

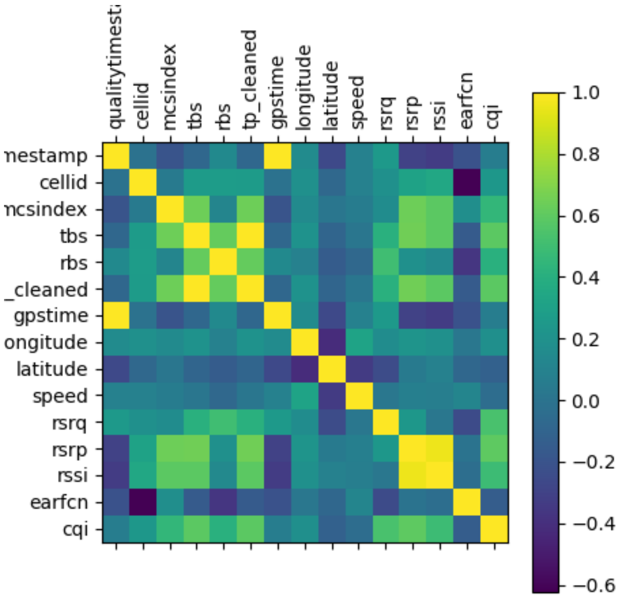


Fig.1 heatmap for upload data

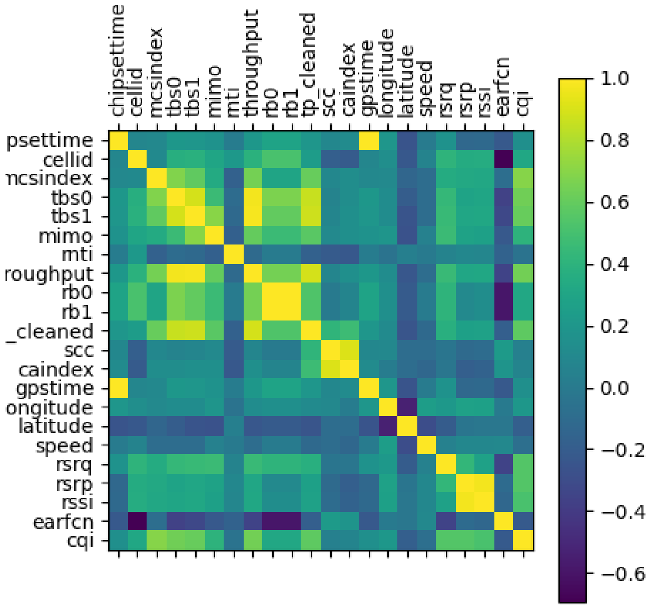


Fig.2 heatmap for download data

Conclusions from section ‘first look’ were proven in the heatmap. It is worth mentioning that the data provided above is not cleaned yet. ‘rb0’ and ‘rb1’ are the same, and ‘tbs0’ and tbs1‘ are in correlation with ‘tp_cleaned’ and ‘throughput’.

There is also a correlation between ‘earfcn’ and ‘rb0’, ‘rb1’, ‘tbs0’ and ‘tbs1’.

The ‘chipsettime’ and ‘gpstime’ are similar, as we can see on the chart below. The same with ‘qualitytimestamp’, so the conclusion that they are the same is proven.

- Distribution analysis for ‘upload’ and ‘ download’ dataset

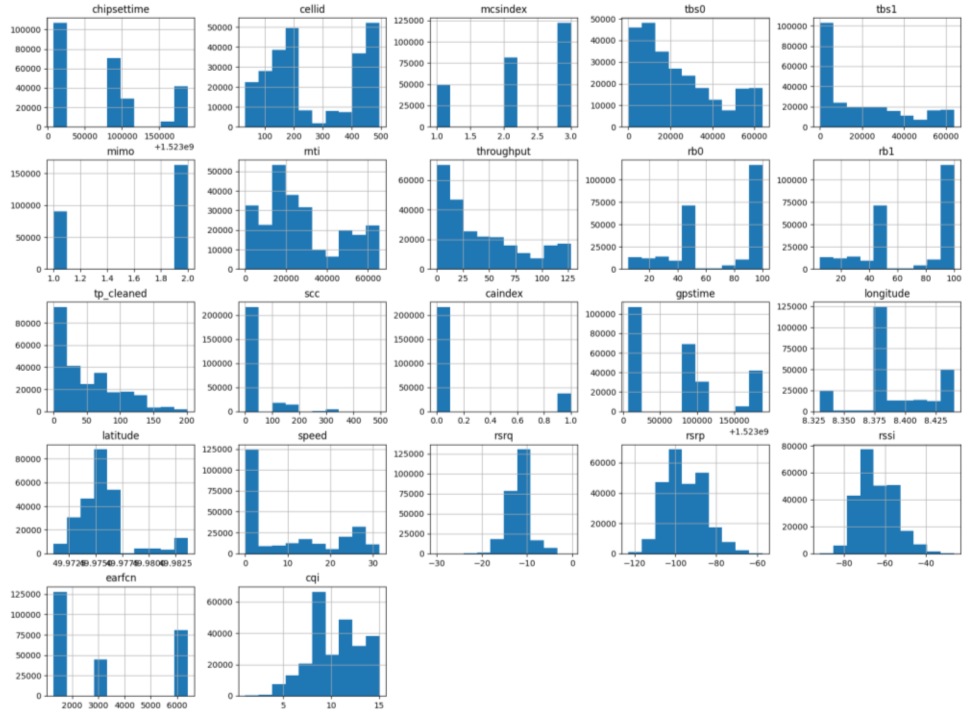


Fig.3 histogram for download data

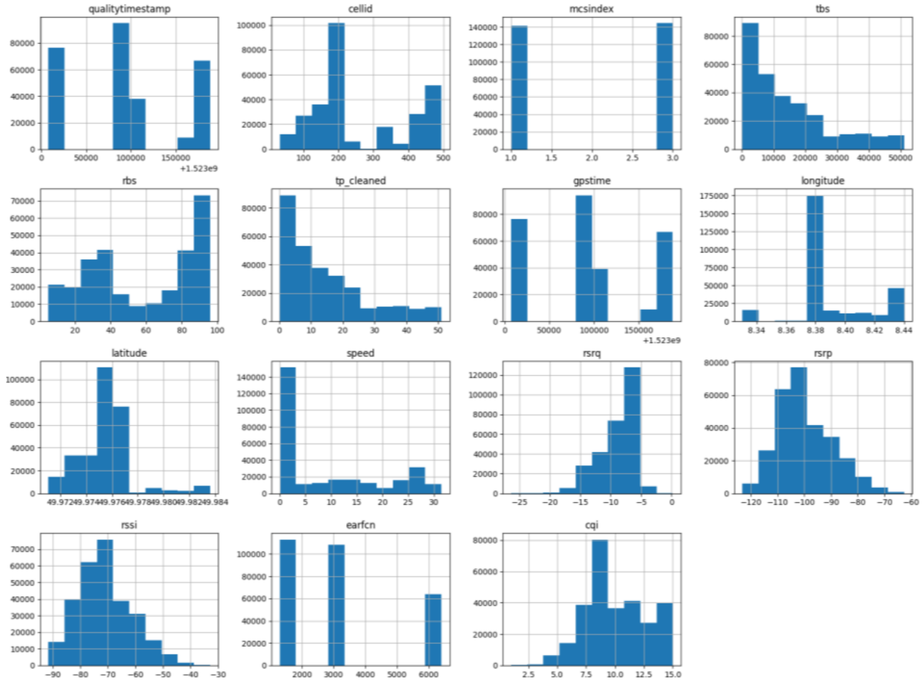


Fig.4 histogram for upload data

More conclusions can be provided through histograms. The difference between ‘throughput’ and ‘tp_cleaned’ is visible in this map. The ‘caindex’ in most cases is 0. This is the difference between two mentioned values. It is less than ¼ of the data.‘rb0’ and ‘rb1’ are the same as it is seen on histogram. Therefore, deletion of one of those was a good decision. Data in both histograms oscillates in the same range, or similar. Both (download and upload) are taken from the same time(‘gpstime’) and places(‘longitude’, latitude), but different amounts of rows are provided.

- outliers and mean analysis for ‘upload’ and ‘ download’ dataset

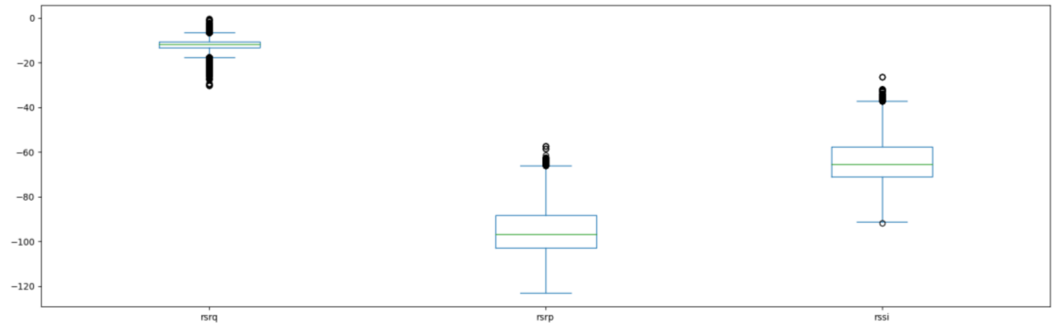


Fig.5 boxplot of download data ['rsrq', 'rsrp', 'rssi']

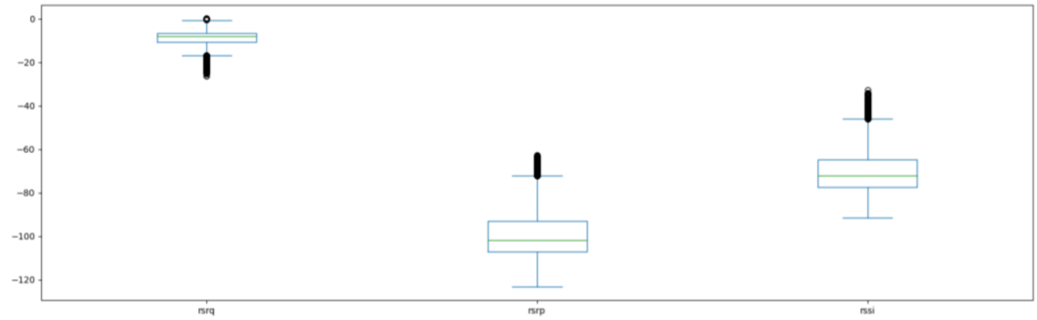


Fig.6 boxplot of upload data ['rsrq', 'rsrp', 'rssi']

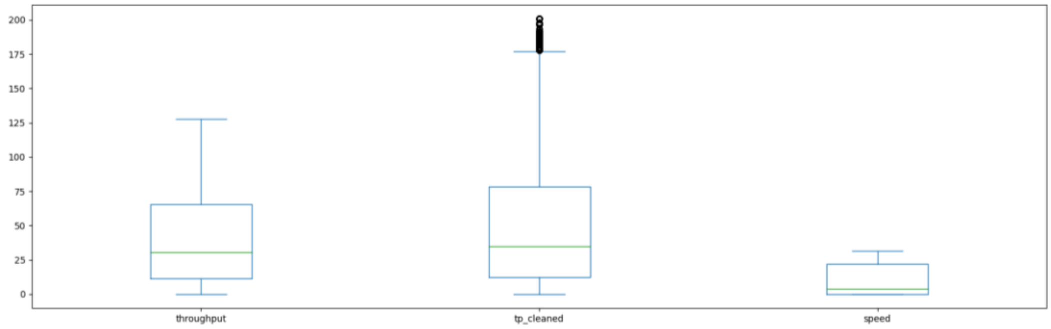


Fig.7 boxplot of download data ['throughput', 'tp_cleaned', 'speed']

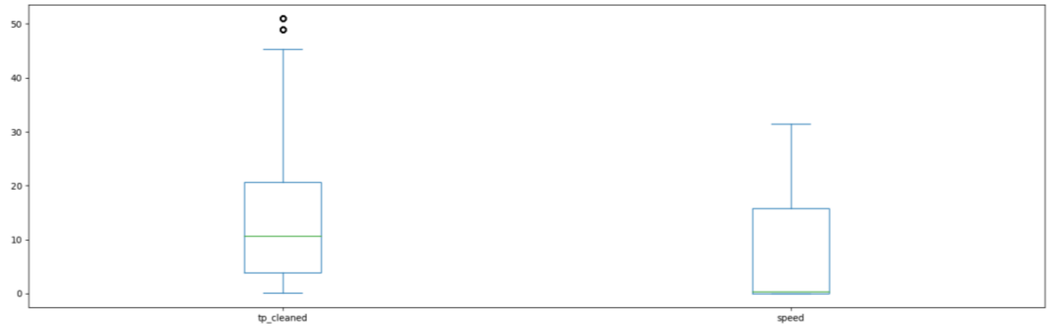


Fig.8 boxplot of ‘upload’ data [‘tp_cleaned’, ‘speed’]

Something interesting can be observed in three columns ('rsrq', 'rsrp', 'rssi'). All of them are similar but have different amounts of outliers. The range of these two charts are the same, but in histograms (Fig.3 and Fig.4) above there is no such correlation as seen above.

Similar analysis was conducted for ‘merged’ dataset and the conclusion was as one would expect from merging two datasets ‘download’ and ‘upload’.

- Correlation analysis for merged dataset

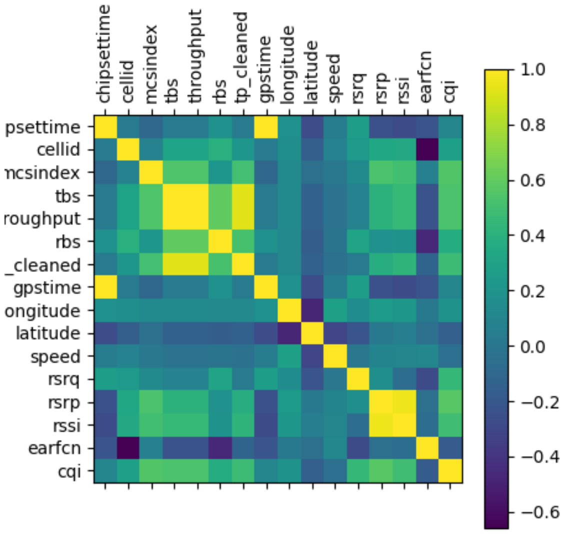


Fig.9 heatmap for merged data

High correlation can be observed in attributes like chipsettime and gpstime, cellid and earfcn, rsrp and rssi, tp_cleaned and tbs, tp_cleaned and throughput. This heatmap proves the assumptions presented before, which is the simplest representation of what was presented before with the smallest column set.

- Distribution analysis for merged dataset

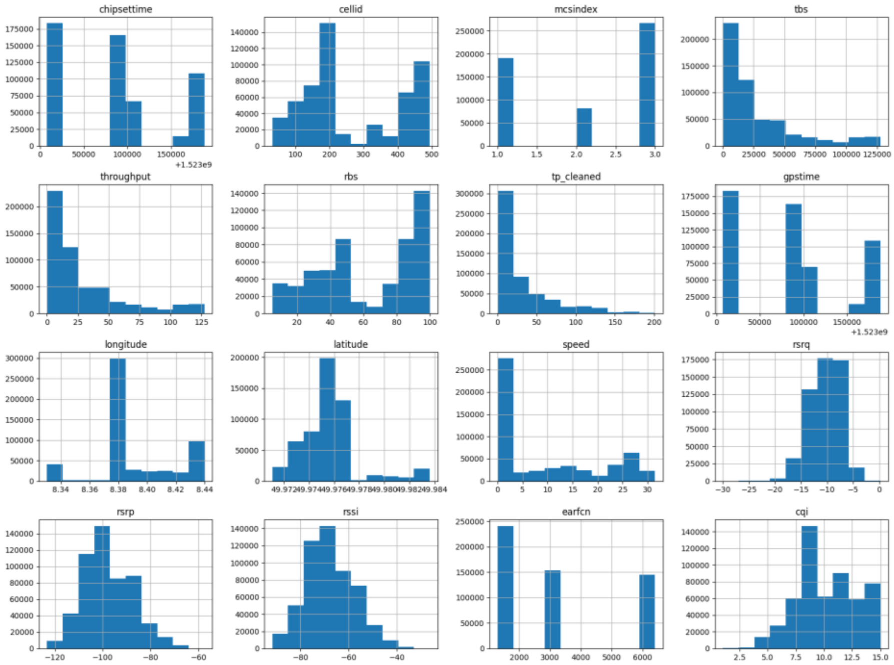


Fig.10 histogram for merged data

- outliers and mean analysis for merged dataset

Distribution of values of attributes among ‘merged’ data frame is similar to distribution for ‘downloads’ and ‘uploads’. That is indicating that feature construction was correctly done.

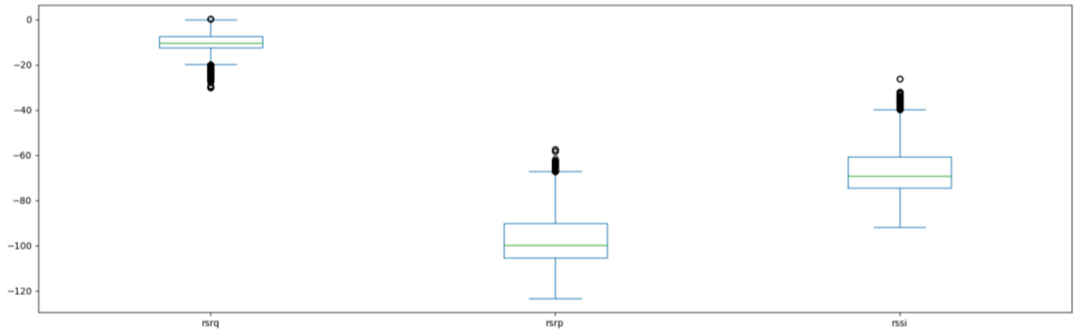


Fig.11 boxplot of merged data ['rsrq', 'rsrp', 'rssi']

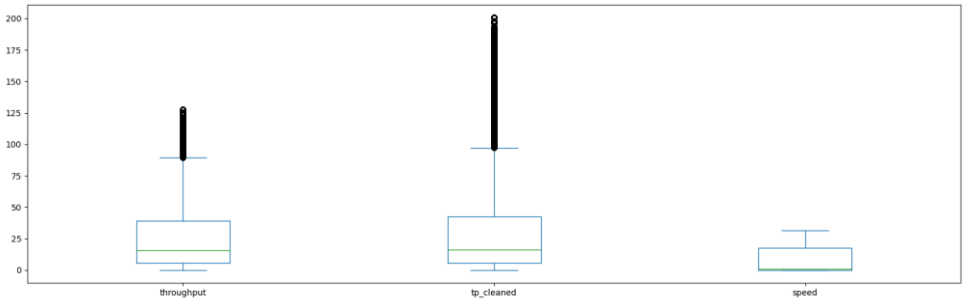
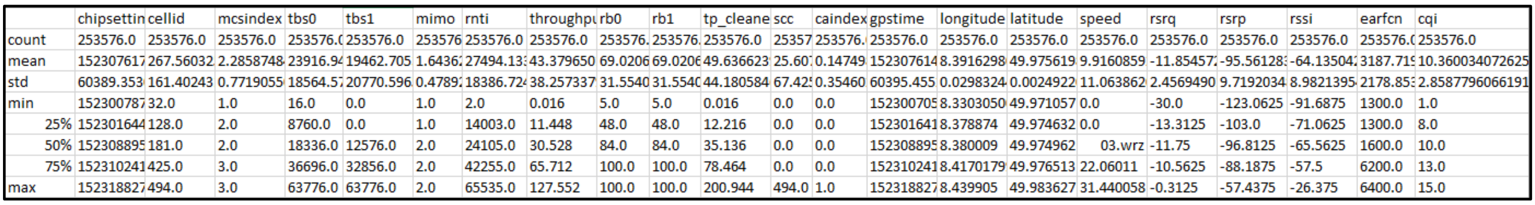


Fig.12 boxplot of merged data [‘throughput,’tp_cleaned’,’speed’’]

Two boxplots were visualized. First including attributes as 'rsrq', 'rsrp', 'rssi' and later ‘throughput,’tp_cleaned’,’speed’. As the first plot values seem to be very similar to what we acquire during analysis of ‘download’ and ‘upload’ data frames, the second plot seems to be different for tp_cleaned. The plot shows a high amount of outliers and this can be a potential bias of the merging part. As the reason for this is not known and there is no impact on prediction, this observation was ignored.

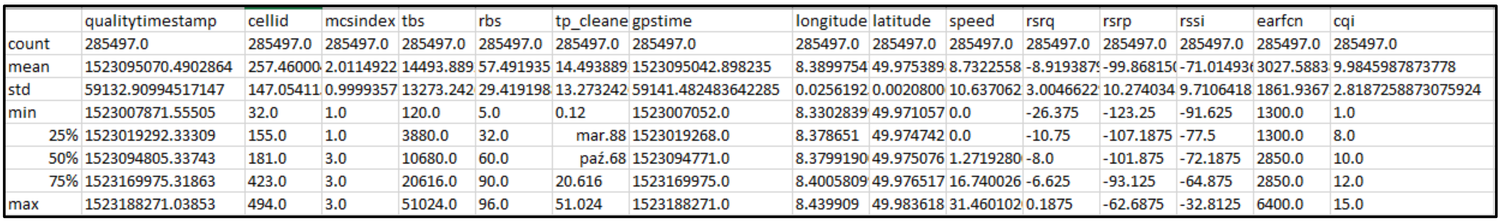
- Statistical basic calculations analysis

Further analysis included, analysis of basic statistical calculation. Below there are tables of statistics for constructed data frames: ‘download’, ‘upload’, ‘merged’. Statistics were calculated using the describe() method from the pandas library.



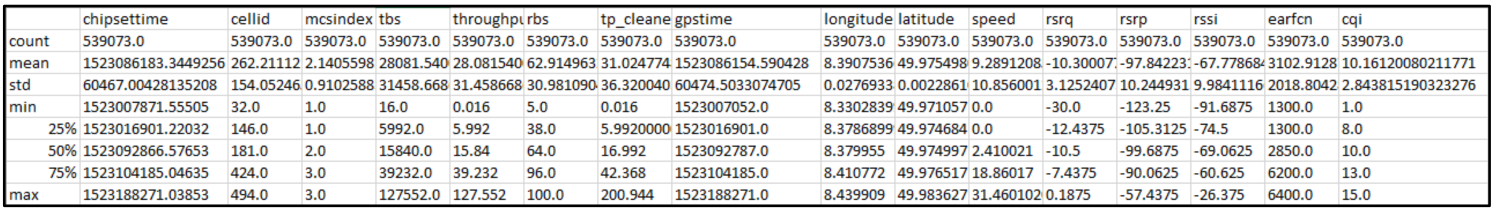
	chipsettin	cellid	mcsindex	tbs0	tbs1	mimo	rnti	throughpu	rb0	rb1	tp_cleane	scc	caindex	gpstime	longitude	latitude	speed	rsrq	rsrp	rssi	earfcn	cqi
count	253576.0	253576.0	253576.0	253576.0	253576.0	253576	253576.0	253576.0	253576.	253576.	253576.0	25357	253576.	253576.0	253576.0	253576.0	253576.0	253576.0	253576.0	253576.0	253576.0	253576.0
mean	152307617	267.56032	2.2858748	23916.94	19462.705	1.6436	27494.13	43.379650	69.0206	69.0206	49.636623	25.60	0.14749	152307614	8.3916298	49.975619	9.9160859	-11.85457	-95.56128	-64.13504	3187.71	10.360034072625
std	60389.353	161.40243	0.7719055	18564.57	20770.596	0.4789	18386.72	38.257337	31.5540	31.5540	44.180584	67.42	0.35460	60395.455	0.0298324	0.0024922	11.063862	2.4569490	9.7192034	8.9821395	2178.85	2.8587796066191
min	152300787	32.0	1.0	16.0	0.0	1.0	2.0	0.016	5.0	5.0	0.016	0.0	0.0	152300705	8.3303050	49.971057	0.0	-30.0	-123.0625	-91.6875	1300.0	1.0
25%	152301644	128.0	2.0	8760.0	0.0	1.0	14003.0	11.448	48.0	48.0	12.216	0.0	0.0	152301641	8.378874	49.974632	0.0	-13.3125	-103.0	-71.0625	1300.0	8.0
50%	152308895	181.0	2.0	18336.0	12576.0	2.0	24105.0	30.528	84.0	84.0	35.136	0.0	0.0	152308895	8.380009	49.974962	03.wrz	-11.75	-96.8125	-65.5625	1600.0	10.0
75%	152310241	425.0	3.0	36696.0	32856.0	2.0	42255.0	65.712	100.0	100.0	78.464	0.0	0.0	152310241	8.4170179	49.976513	22.06011	-10.5625	-88.1875	-57.5	6200.0	13.0
max	152318827	494.0	3.0	63776.0	63776.0	2.0	65535.0	127.552	100.0	100.0	200.944	494.0	1.0	152318827	8.439905	49.983627	31.440058	-0.3125	-57.4375	-26.375	6400.0	15.0

Table1 Statistic table for download data



	qualitytimestamp	cellid	mcsindex	tbs	rbs	tp_cleane	gpstime	longitude	latitude	speed	rsrq	rsrp	rssi	earfcn	cqi
count	285497.0	285497.0	285497.0	285497.0	285497.0	285497.0	285497.0	285497.0	285497.0	285497.0	285497.0	285497.0	285497.0	285497.0	285497.0
mean	1523095070.4902864	257.46000	2.0114922	14493.889	57.491935	14.493889	1523095042.898235	8.3899754	49.975389	8.7322558	-8.919387	-99.86815	-71.01493	3027.5883	9.9845987873778
std	59132.90994517147	147.05411	0.9999357	13273.242	29.419198	13.273242	59141.48248364285	0.0256192	0.0020800	10.637062	3.0046622	10.274034	9.7106418	1861.9367	2.8187258873075924
min	1523007871.55505	32.0	1.0	120.0	5.0	0.12	1523007052.0	8.3302839	49.971057	0.0	-26.375	-123.25	-91.625	1300.0	1.0
25%	1523019292.33309	155.0	1.0	3880.0	32.0	mar.88	1523019268.0	8.378651	49.974742	0.0	-10.75	-107.1875	-77.5	1300.0	8.0
50%	1523094805.33743	181.0	3.0	10680.0	60.0	paź.68	1523094771.0	8.3799190	49.975076	1.2719280	-8.0	-101.875	-72.1875	2850.0	10.0
75%	1523169975.31863	423.0	3.0	20616.0	90.0	20.616	1523169975.0	8.4005809	49.976517	16.740026	-6.625	-93.125	-64.875	2850.0	12.0
max	1523188271.03853	494.0	3.0	51024.0	96.0	51.024	1523188271.0	8.439909	49.983618	31.460102	0.1875	-62.6875	-32.8125	6400.0	15.0

Table2 Statistic table for upload data



	chipsettime	cellid	mcsindex	tbs	throughpu	rbs	tp_cleane	gpstime	longitude	latitude	speed	rsrq	rsrp	rssi	earfcn	cqi
count	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0	539073.0
mean	1523086183.3449256	262.21112	2.1405598	28081.540	28.081540	62.914963	31.024774	1523086154.590428	8.3907536	49.975498	9.2891208	-10.30007	-97.84223	-67.77868	3102.9128	10.16120080211771
std	60467.00428135208	154.05246	0.9102588	31458.668	31.458668	30.981090	36.320040	60474.5033074705	0.0276933	0.0022861	10.856001	3.1252407	10.244931	9.9841116	2018.8042	2.843815190323276
min	1523007871.55505	32.0	1.0	16.0	0.016	5.0	0.016	1523007052.0	8.3302839	49.971057	0.0	-30.0	-123.25	-91.6875	1300.0	1.0
25%	1523016901.22032	146.0	1.0	5992.0	5.992	38.0	5.9920000	1523016901.0	8.3786899	49.974684	0.0	-12.4375	-105.3125	-74.5	1300.0	8.0
50%	1523092866.57653	181.0	2.0	15840.0	15.84	64.0	16.992	1523092787.0	8.379955	49.974997	2.410021	-10.5	-99.6875	-69.0625	2850.0	10.0
75%	1523104185.04635	424.0	3.0	39232.0	39.232	96.0	42.368	1523104185.0	8.410772	49.976517	18.86017	-7.4375	-90.0625	-60.625	6200.0	13.0
max	1523188271.03853	494.0	3.0	127552.0	127.552	100.0	200.944	1523188271.0	8.439909	49.983627	31.460102	0.1875	-57.4375	-26.375	6400.0	15.0

Table3 Statistic table for merged data

Using those basic statistical information, the specified analysis can be given. In this case, the best thing to do is compare download and upload datasets to prove that information from the merged dataset makes sense and is merged properly.

Both datasets have columns named "qualitytimestamp" and "chipsettime," representing timestamps. The timestamp range is the same in both datasets, ranging from 1523007872 to 1523188270. This indicates that the data was collected over approximately 24 hours.

The "cellid" column in both datasets represents cell IDs. The range of cell IDs is the same, ranging from 32 to 494. The mean value is approximately 257.46, which is slightly higher than the median value of 181. This indicates that the distribution of cell IDs is slightly skewed towards higher values.

The "mcsindex" column in both datasets represents the Modulation and Coding Scheme (MCS) index values. The range of MCS index values is different between the datasets. In the download dataset, it ranges from 1 to 3, while in the upload dataset, it ranges from 1 to 3 as well. This indicates that the possible MCS index values are the same in both datasets.

The "tbs0" and "tbs1" columns in the download dataset and the "tbs" column in the upload dataset represent Transport Block Size (TBS) values. The range of TBS values is different between the datasets. In the download dataset, "tbs0" and "tbs1" range from 16 to 63 776, while in the upload dataset, "tbs" ranges from 120 to 51 024. This indicates that the TBS ranges differ between the datasets.

The "rb0" and "rb1" columns in the download dataset and the "rbs" column in the upload dataset represent Resource Block Size (RBS) values. The range of RBS values is the same in both datasets, ranging from 5 to 100. This indicates that the possible RBS values are the same in both datasets.

The "tp_cleaned" column in both datasets represents throughput values. The range of throughput values is different between the datasets. In the download dataset, it ranges from 0.016 to 200.944, with a mean value of 49.64. In the upload dataset, it ranges from 0.12 to 51.024, with a mean value of 14.50. This indicates that the throughput values differ between the datasets.

The "gpstime" column in both datasets represents GPSTime values. The timestamp range is the same for "gpstime" and "qualitytimestamp," ranging from 1523007872 to 1523188270. This indicates that the GPSTime values are consistent between the datasets.

The "longitude" and "latitude" columns in both datasets represent geographical coordinates. The range of longitude values is similar between the datasets. In the download dataset, it ranges from 8.330305 to 8.439905, while in the upload dataset, it ranges from 8.330284 to 8.439909. The range of latitude values is the same in both datasets, ranging from 49.971057 to 49.983627. This suggests that the data was collected from the same location on Earth in both datasets.

The "speed" column in both datasets represents the speed values. In the download dataset, the speed ranges from 0 to 31.440058, with a mean value of 9.916085923. In the upload dataset, the speed ranges from 0 to 31.460102, with a mean value of 8.73225583. This indicates that the speed values differ slightly between the datasets, with the download dataset showing a slightly higher mean speed.

The "rsrq," "rsrp," and "rssi" columns in both datasets represent signal strength values. The range of "rsrq" values is similar in both datasets, ranging from -30 to -0.3125. The range of "rsrp" values is also similar, ranging from -123.0625 to -57.4375. The range of "rssi" values is similar as well, ranging from -91.6875 to -26.375. This indicates that the signal strength values are comparable between the datasets.

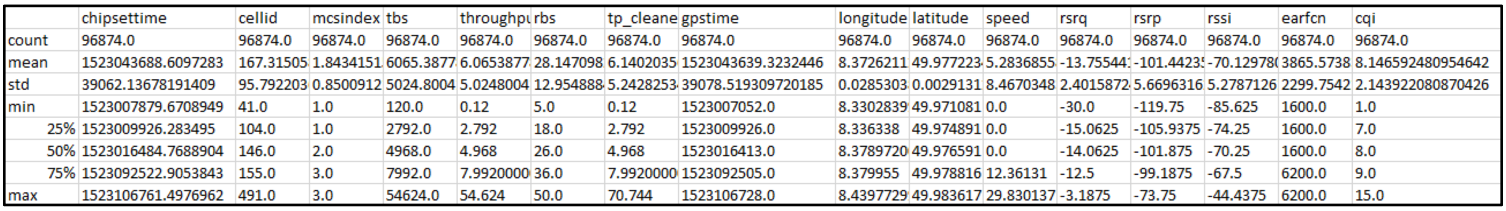
The "earfcn" column in both datasets represents the E-UTRA Absolute Radio Frequency Channel Number. The range of "earfcn" values is similar between the datasets, ranging from 1300 to 6400. This suggests that the frequency channel numbers used for communication are consistent between the datasets.

The "cqi" column in both datasets represents the Channel Quality Indicator values. The range of "cqi" values is similar, ranging from 1 to 15. This indicates that the possible CQI values are the same in both datasets.

To summarize, while there are some differences in the ranges and mean values of certain columns between the download and upload datasets, several columns show similar patterns. The "cellid," "gpstime," "longitude," and "latitude" columns indicate that the data was collected from the same location and time. Additionally, the "mcsindex," "rb0," "rb1," "rsrq," "rsrp," "rssi," "earfcn," and "cqi" columns exhibit comparable ranges and values, suggesting consistent characteristics across both datasets. However, there are variations in the "tbs0," "tbs1," "tbs," "tp_cleaned," and "speed" columns, indicating differences in the specific measurements between the download and upload datasets.

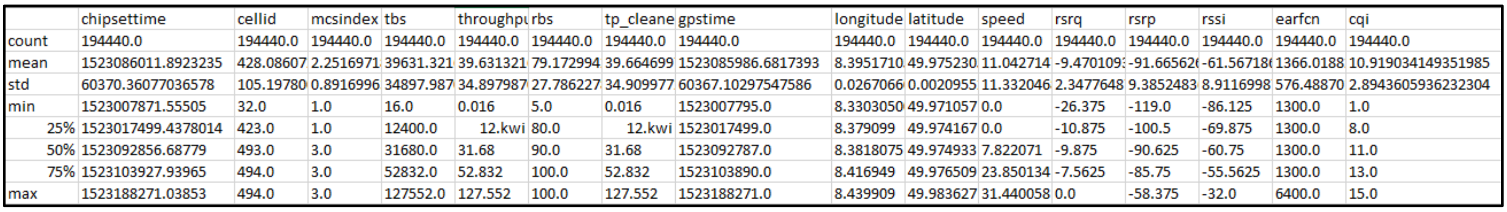
- Operator analysis in ‘merged’ dataset.

Analysis of network providers - on ‘merged’(download and upload) data frame. Below there are tables of statistics of operators: o2, telekom and vodafone. Statistics were calculated using the describe() method from the pandas library.



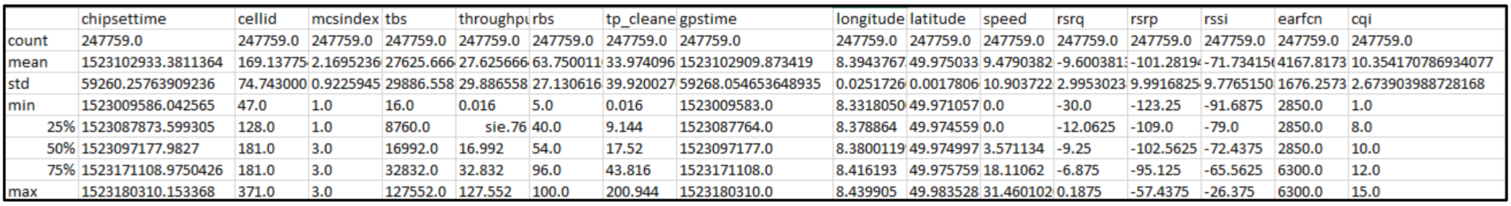
	chipsettime	cellid	mcsindex	tbs	throughpu	rbs	tp_cleane	gpstime	longitude	latitude	speed	rsrq	rsrp	rssi	earfcn	cqi
count	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0	96874.0
mean	1523043688.6097283	167.31505	1.8434151	6065.3877	6.0653877	28.147098	6.1402035	1523043639.3232446	8.3726211	49.977223	5.2836855	-13.75544	-101.4423	-70.12978	3865.5738	8.146592480954642
std	39062.13678191409	95.792203	0.8500912	5024.8004	5.0248004	12.954888	5.2428253	39078.519309720185	0.0285303	0.0029131	8.4670348	2.4015872	5.6696316	5.2787126	2299.7542	2.143922080870426
min	1523007879.6708949	41.0	1.0	120.0	0.12	5.0	0.12	1523007052.0	8.3302839	49.971081	0.0	-30.0	-119.75	-85.625	1600.0	1.0
25%	1523009926.283495	104.0	1.0	2792.0	2.792	18.0	2.792	1523009926.0	8.336338	49.974891	0.0	-15.0625	-105.9375	-74.25	1600.0	7.0
50%	1523016484.7688904	146.0	2.0	4968.0	4.968	26.0	4.968	1523016413.0	8.3789720	49.976591	0.0	-14.0625	-101.875	-70.25	1600.0	8.0
75%	1523092522.9053843	155.0	3.0	7992.0	7.9920000	36.0	7.9920000	1523092505.0	8.379955	49.978816	12.36131	-12.5	-99.1875	-67.5	6200.0	9.0
max	1523106761.4976962	491.0	3.0	54624.0	54.624	50.0	70.744	1523106728.0	8.4397729	49.983617	29.830137	-3.1875	-73.75	-44.4375	6200.0	15.0

Table o2-merged-stats



	chipsettime	cellid	mcsindex	tbs	throughpu	rbs	tp_cleane	gpstime	longitude	latitude	speed	rsrq	rsrp	rssi	earfcn	cqi
count	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0	194440.0
mean	1523086011.8923235	428.08607	2.2516971	39631.321	39.631321	79.172994	39.664699	1523085986.6817393	8.3951710	49.975230	11.042714	-9.470109	-91.66562	-61.56718	1366.0188	10.919034149351985
std	60370.36077036578	105.19780	0.8916996	34897.987	34.897987	27.786227	34.909977	60367.10297547586	0.0267066	0.0020955	11.332046	2.3477648	9.3852483	8.9116998	576.48870	2.894360593623230
min	1523007871.55505	32.0	1.0	16.0	0.016	5.0	0.016	1523007795.0	8.3303050	49.971057	0.0	-26.375	-119.0	-86.125	1300.0	1.0
25%	1523017499.4378014	423.0	1.0	12400.0	12.kwi	80.0	12.kwi	1523017499.0	8.379099	49.974167	0.0	-10.875	-100.5	-69.875	1300.0	8.0
50%	1523092856.68779	493.0	3.0	31680.0	31.68	90.0	31.68	1523092787.0	8.3818075	49.974933	7.822071	-9.875	-90.625	-60.75	1300.0	11.0
75%	1523103927.93965	494.0	3.0	52832.0	52.832	100.0	52.832	1523103890.0	8.416949	49.976509	23.850134	-7.5625	-85.75	-55.5625	1300.0	13.0
max	1523188271.03853	494.0	3.0	127552.0	127.552	100.0	127.552	1523188271.0	8.439909	49.983627	31.440058	0.0	-58.375	-32.0	6400.0	15.0

Table telekom-merged-stats



	chipsettime	cellid	mcsindex	tbs	throughpu	rbs	tp_cleane	gpstime	longitude	latitude	speed	rsrq	rsrp	rssi	earfcn	cqi
count	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0	247759.0
mean	1523102933.3811364	169.13775	2.1695236	27625.666	27.625666	63.750011	33.974096	1523102909.873419	8.3943767	49.975033	9.4790382	-9.600381	-101.2819	-71.73415	4167.8173	10.354170786934077
std	59260.25763909236	74.743000	0.9225945	29886.558	29.886558	27.130616	39.920027	59268.054653648935	0.0251726	0.0017806	10.903722	2.9953023	9.9916825	9.7765150	1676.2573	2.673903988728168
min	1523009586.042565	47.0	1.0	16.0	0.016	5.0	0.016	1523009583.0	8.3318050	49.971057	0.0	-30.0	-123.25	-91.6875	2850.0	1.0
25%	1523087873.599305	128.0	1.0	8760.0	sie.76	40.0	9.144	1523087764.0	8.378864	49.974559	0.0	-12.0625	-109.0	-79.0	2850.0	8.0
50%	1523097177.9827	181.0	3.0	16992.0	16.992	54.0	17.52	1523097177.0	8.3800119	49.974997	3.571134	-9.25	-102.5625	-72.4375	2850.0	10.0
75%	1523171108.9750426	181.0	3.0	32832.0	32.832	96.0	43.816	1523171108.0	8.416193	49.975759	18.11062	-6.875	-95.125	-65.5625	6300.0	12.0
max	1523180310.153368	371.0	3.0	127552.0	127.552	100.0	200.944	1523180310.0	8.439905	49.983528	31.460102	0.1875	-57.4375	-26.375	6300.0	15.0

Table vodafone - merged - stats

Data in each operator are very similar. "longitude" and "latitude" are exactly the same, as well as most maximal values, except “rsrq” and minimal values, except “rsrp” and “rssi”. This information once again confirms that all the data was taken from the same place, time and with similar conditions. It can be noted that each operator has a different number of data: o2 - 96 874, telkom - 194 440 and vodafone - 247 759.

Visualizations of Analysis of network providers - on ‘merged’(download and upload) data frame. This analysis was conducted on already cleaned data, this means some rows may have been deleted - that has duplicates, missing data etc… Visualization of counted measurements by operators and selected statistics by operator are shown below.

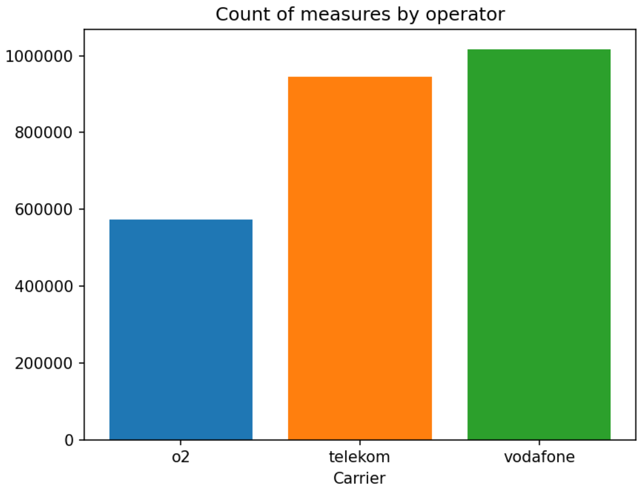


Fig Operator count measurements

Above plot shows counted all measurements (rows) for each operator for merged data frames. Smallest set of measurements was conducted for ‘o2’ operator and the biggest for ‘vodafone’.

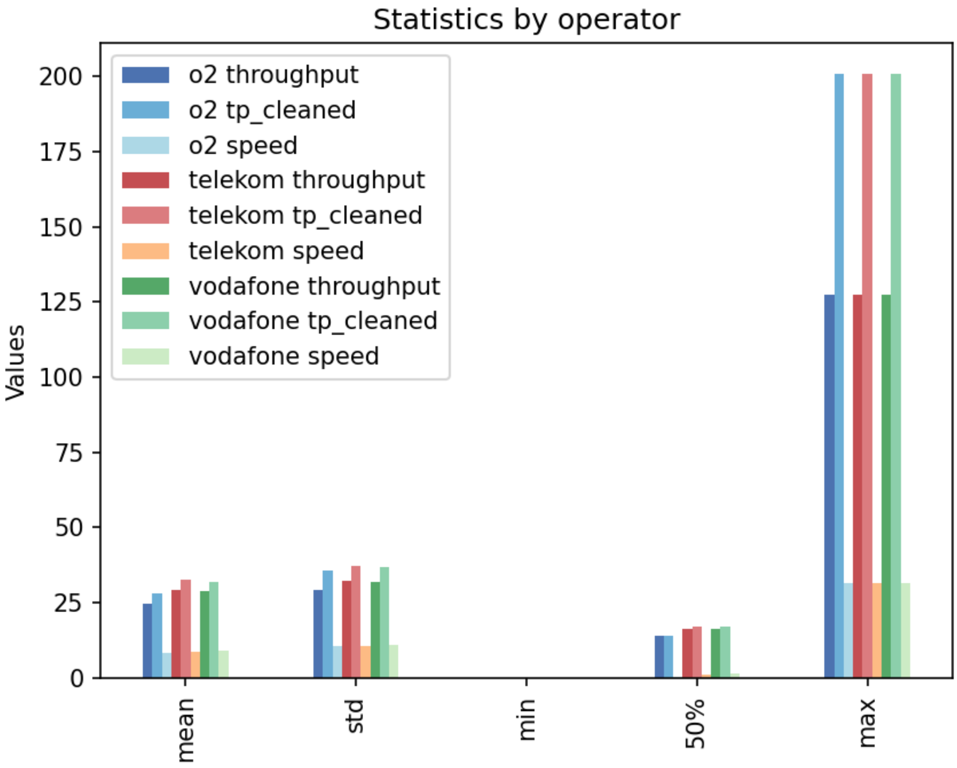


Fig selected statistics by operator

Above plot shows selected statistics for each operator for the ‘merged’ data frame. Values for min are not visible because these are values from 0-0.2. Each value for each operator is very similar. It means that basic statistical values are similar for all operators and source datasets are balanced. 50 percentile is a median of values. Std is standard deviation.

Scaling data:

Next step and the last step of the preprocessing phase was to scale numeric data to provide more readable data into machine learning algorithms. Two scaling techniques were proposed: standardization using the function 'StandardScaler' from the 'sklearn' library. and normalization - 'MinMaxScaler' from the same library.

The purpose is that, later the model will improve its capability to discover patterns and results that potentially have more impact on the predicted value. In this case, the chosen features for this function were: almost all the numeric data was normalized to exclude 'chipsettime' from download and merged and 'qualitytimestamp' from upload data. Text data was not taken into consideration. It is up to the user to select which technique wants to use on data. First, a normalization technique was used.

Predicting data and Evaluating results

After the preprocessing phase it is possible to make predictions on this data. Choosing a model and building it depends on our chosen Target - Predicting value. It can be a regression problem or classification problem. It was established to do (separately) prediction of attributes (targets):

- ‘throughput’ (regression problem) or ‘tp_cleaned’ for uploads dataset
- ‘mobiProv_name’ (classification problem)- it is name of mobile network provider {o2, vodafone, telekom} and it is constructed feature (feature construction)
- ‘speed‘ (regression problem)

Moreover, the above prediction was built on three datasets created in the preprocessed phase - ‘download’, ‘upload’ and ‘merged’.

The Prediction module has two functions. For the regression problem and for classification problem. These functions build models, train data and then calculate evaluation metrics and create plots. For regression and classification different models, different evaluation metrics and different plots were used.

Machine learning models used in this module were: decision tree, gradient boosting and random forest both on regression and classification problem and linear regression on regression problem.

Different attributes acting as predictors were chosen depending on the experiment. These attributes were either

- all attributes or
- for 10 attributes experiment - 'mcsindex','longitude', 'latitude', speed','rsrq','rsrp','rssi','earfcn','cqi','mobiProv_name' or
- for 5 attributes experiment - 'speed','rsrq','rsrp','rssi','earfcn'.

Prediction and evaluation was done in python using supporting libraries. Last experiment was done on preprocessed datasets in the Orange tool.

Evaluation metrics for the regression problem were: mean squared error, root mean square error and R2 (r-squared). The evaluation of the model's performance was visually presented using a scatter plot of the predicted values versus the actual values. Evaluation metrics for the classification problem were: accuracy, precision, recall, F1 score. The evaluation of the model's performance was visually presented using both a confusion matrix and an ROC curve. The results are presented and described in the ‘Experiments’ section.

Experiments

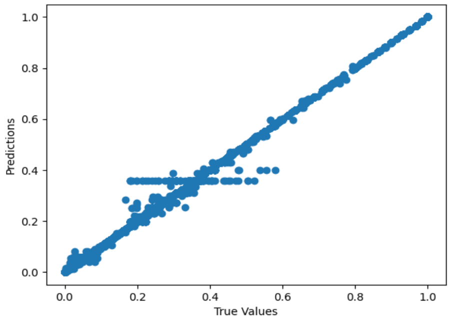
Experiments were conducted for different datasets, for different algorithms, for different attributes acting as predictors and for different targets.

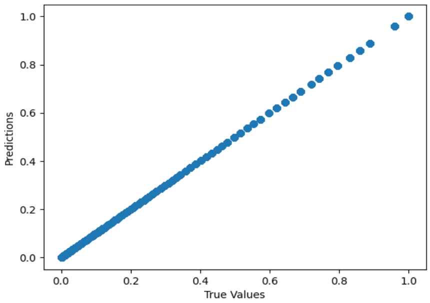
Overall model performance was very good, for some models it was perfect. Although overfitting of the model is possible, a more probable option is that models were recognizing patterns and relating very well to many dependencies within attributes and were able to classify and make regression almost perfectly. Similar results across different models and and on testing data would confirm this option. Preprocessing phase was very detailed and the output data was very friendly for machine learning models due to dealing with noise, scaling data, one hot encoding etc. And although the model performed well on non-preprocessed data, the whole process was a very good training and learning experience.

Experiment 1 - Cleaned and scaled data for datasets downloads, uploads

Experiment performed on cleaned and scaled ‘download’ and ‘upload’ datasets

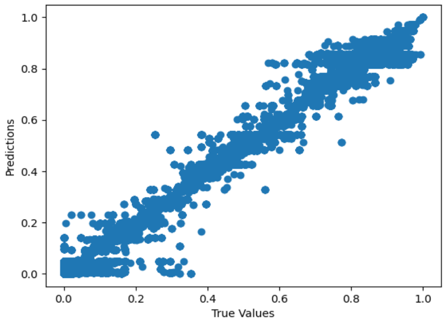
model - decision tree regressor, target - ‘throughput

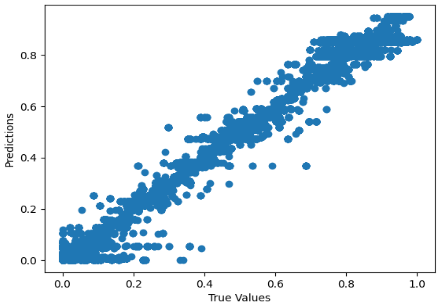
dataset cleaned and scaled ‘downloads’ number of attribute (predictors) - 30 MSE: 1.7094 *10^-5 RMSE: 0.0041 R-squared: 0.9998	

dataset cleaned and scaled ‘uploads’ number of attribute (predictors) - 17 MSE: 5.9559 * 10^-5 RMSE: 2.4404 * 10^-5 R-squared: 0.9999	

Throughput prediction (and tp_cleaned for ‘uploads’) was well predicted by decision tree model and it is visible on the plot and in measures. Predictions were consistent with test data.

model - decision tree regressor, target - ‘speed’

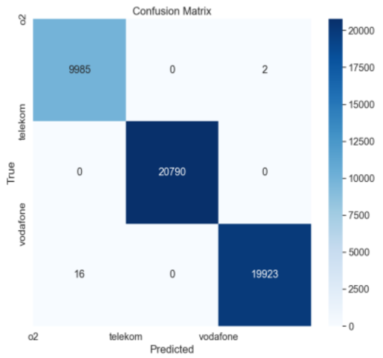
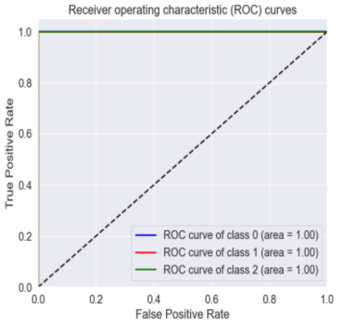
dataset cleaned and scaled ‘downloads’ number of attribute (predictors) - 30 MSE: 0.001 RMSE: 0.0318 R-squared: 0.9917	

dataset cleaned and scaled ‘upload’ number of attribute (predictors) - 17 MSE: 0.0009 RMSE: 0.03 R-squared: 0.992	

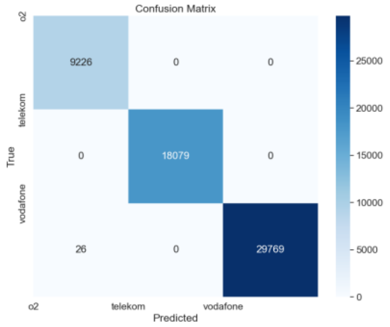
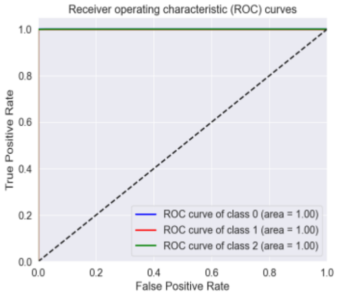
Prediction for speed was not as ideal as in the previous target because predicted data was not 100% consistent with testing data (true values). Nevertheless in most cases prediction was correct and overall rating is very good. E.g. R2 measure 1 is an ideal prediction and the result was very close to that.

model - decision tree classifier, target - 'mobiProv_name'

<table>
<tr>
<td>‘download’ number of attribute (predictors) - 28

Accuracy: 0.9996
Precision: 0.9996
Recall: 0.9996
F1 Score: 0.9996</td>
<td>
</td>
<td>
</td>
</tr>
</table>

<table>
<tr>
<td>‘upload’
number of attribute (predictors) - 15

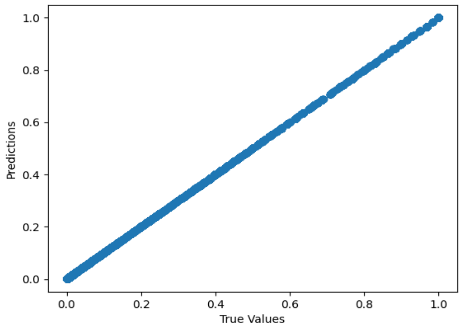
Accuracy: 0.9995
Precision: 0.9995
Recall: 0.9995
F1 Score: 0.9995</td>
<td>
</td>
<td>
</td>
</tr>
</table>

The classification problem for the target indicating the name of the mobile provider was also very good. Accuracy is nearly 1 and just in few cases models were wrong and not consistent with test data.

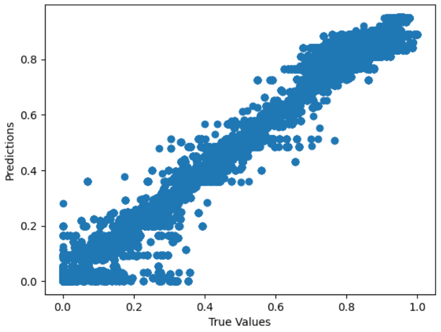
Experiment 2 - Cleaned and scaled for a dataset merged.

Experiment performed on cleaned and scaled merged dataset.

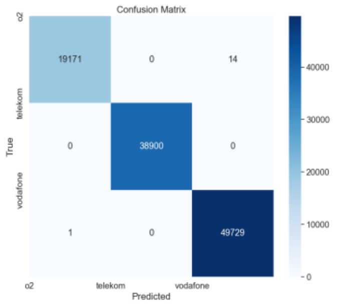
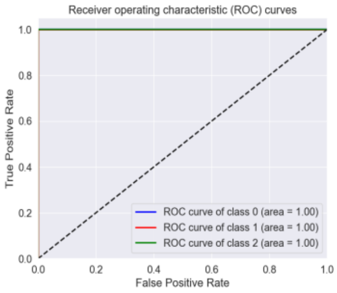
model - decision tree regressor, target - ‘throughput

dataset cleaned and scaled merged number of attribute (predictors) - 23 MSE: 3.3590 * 10^-9 RMSE: 5.7956*10^-5 R-squared: 0.9999	

model - decision tree regressor, target - ‘speed’

dataset cleaned and scaled merged number of attribute (predictors) - 23 MSE: 0.0011 RMSE: 0.0326 R-squared: 0.9910	

model - decision tree classifier, target - 'mobiProv_name'

dataset cleaned and scaled merged number of attribute (predictors) - 21 Accuracy: 0.9998 Precision: 0.9998 Recall: 0.9998 F1 Score: 0.9998		

Same model and same targets was tested on slightly different data. Data that were cleaned and scaled was put as datasets into models prediction. Results were also very good and almost ideal. Classification problem had fewer mistakes.

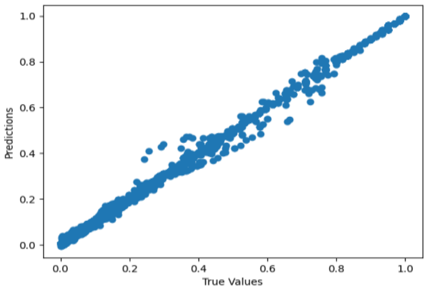
Experiment 3 - Different models.

experiment performed on different models: gradient boosting, random forest and linear regression.

model - gradientboosting regressor, target - ‘throughput’

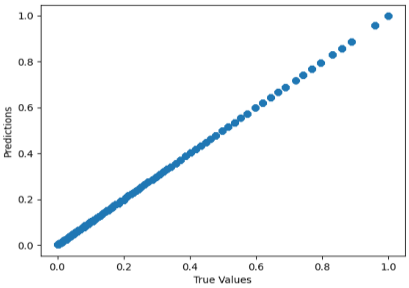
dataset cleaned and scaled ‘downloads’
number of attribute (predictors) - 30

MSE: 2.0721 * 10^-5
RMSE: 0.0045
R-squared: 0.9997



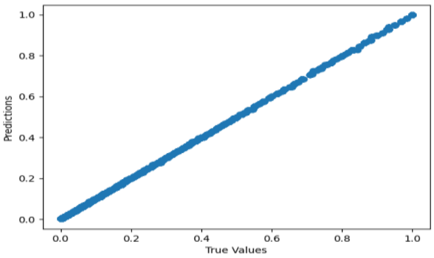
dataset cleaned and scaled ‘uploads’
number of attribute (predictors) - 17

MSE: 1.0898 * 10-6
RMSE: 0.0011
R-squared: 0.9999

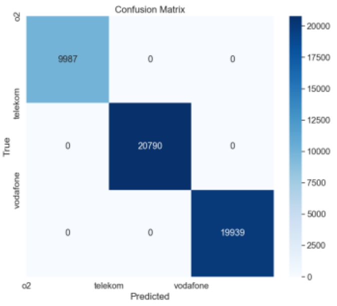
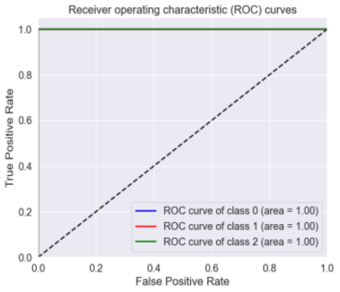
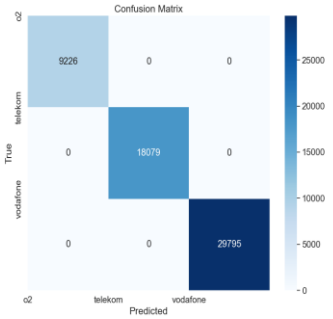
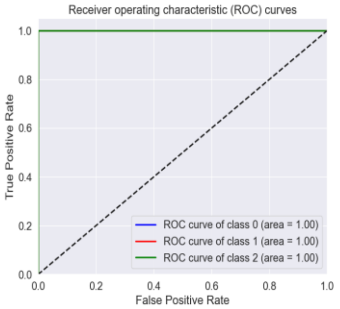
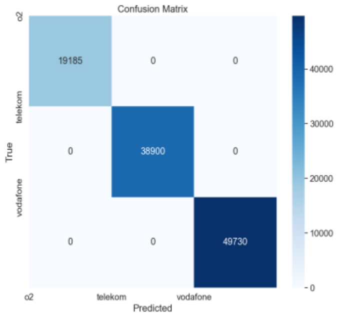
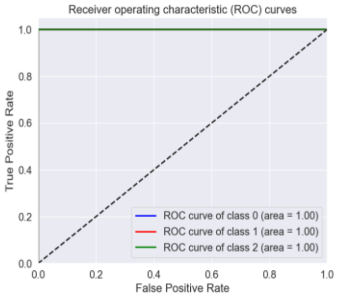


dataset cleaned and scaled ‘merged’
number of attribute (predictors) - 23

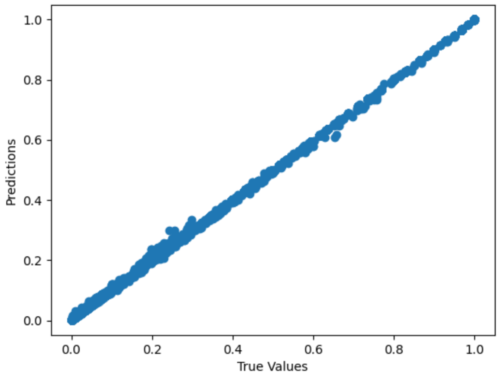
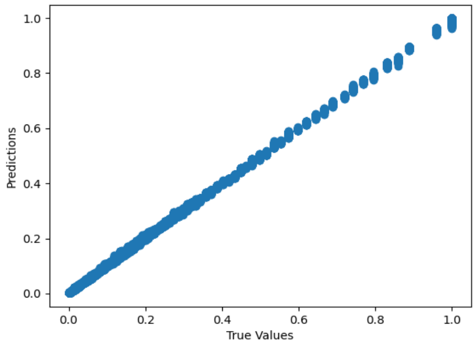
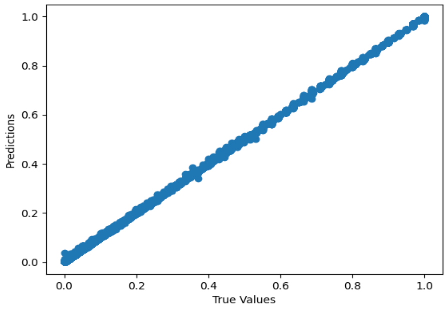
MSE: 1.0749 * 10^-6
RMSE: 0.001
R-squared: 0.9999



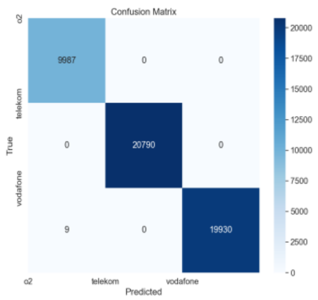
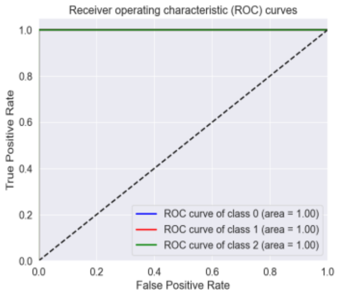
model - Gradient Boosting classifier, target - 'mobiProv_name'

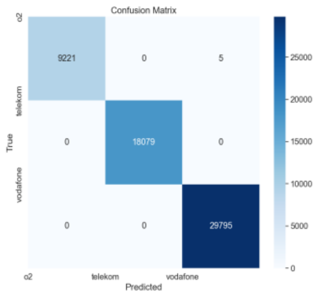
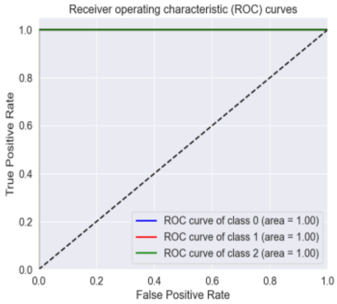
dataset cleaned and scaled ‘downloads’ number of attribute (predictors) - 28 Accuracy: 1.0 Precision: 1.0 Recall: 1.0 F1 Score: 1.0		
dataset cleaned and scaled ‘uploads’ number of attribute (predictors) - 15 Accuracy: 1.0 Precision: 1.0 Recall: 1.0 F1 Score: 1.0		
dataset cleaned and scaled ‘merged’ number of attribute (predictors) - 21 Accuracy: 1.0 Precision: 1.0 Recall: 1.0 F1 Score: 1.0		

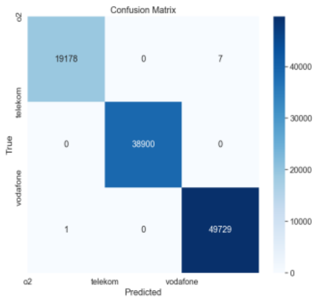
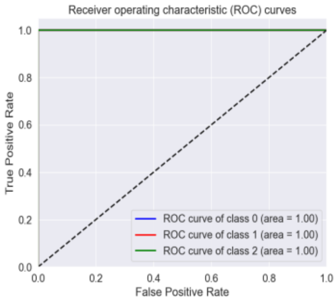
model - RandomForestRegressor, target - throughput’

dataset cleaned and scaled ‘downloads’ number of attribute (predictors) - 30 MSE: 2.0161 * 10-6 RMSE: 0.0014 R-squared: 0.9999	
dataset cleaned and scaled ‘uploads’ number of attribute (predictors) - 17 MSE: 9.952 * 10^-6 RMSE: 0.0031 R-squared: 0.9998	
dataset cleaned and scaled ‘merged’ number of attribute (predictors) - 23 MSE: 1.3376 * 10^-6 RMSE: 0.0011 R-squared: 0.9999	

model - RandomForestClassifier, target - 'mobiProv_name'

dataset cleaned and scaled ‘downloads’ number of attribute (predictors) - 28 Accuracy: 0.9998 Precision: 0.9998 Recall: 0.9998 F1 Score: 0.9998		

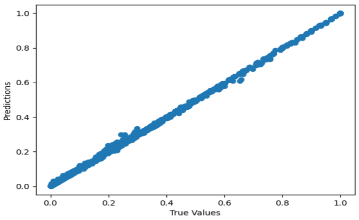
dataset cleaned and scaled uploads number of attribute (predictors) - 15 Accuracy: 0.9999 Precision: 0.9999 Recall: 0.9999 F1 Score: 0.9999		

dataset cleaned and scaled ‘merged’ number of attribute (predictors) - 21 Accuracy: 0.9999 Precision: 0.9999 Recall: 0.9999 F1 Score: 0.9999		

model - LinearRegression, target - throughput’

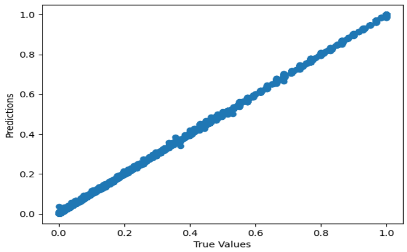
dataset cleaned and scaled ‘downloads’
number of attribute (predictors) - 30

MSE: 7.368546880968007e-27
RMSE: 8.584024045264556e-14
R-squared: 1.0



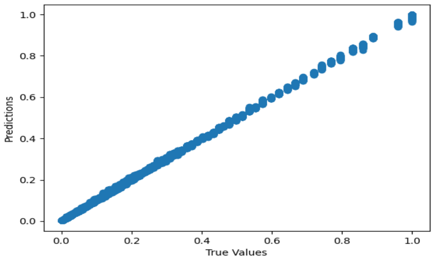
dataset cleaned and scaled ‘uploads
number of attribute (predictors) - 17

MSE: 1.5029 * 10^-27
RMSE: 3.876 * 10^-14
R-squared: 1.0



dataset cleaned and scaled ‘merged’
number of attribute (predictors) - 23

MSE: 2.7122 * 10^-27
RMSE: 5.2079 * 10^-14
R-squared: 1.0



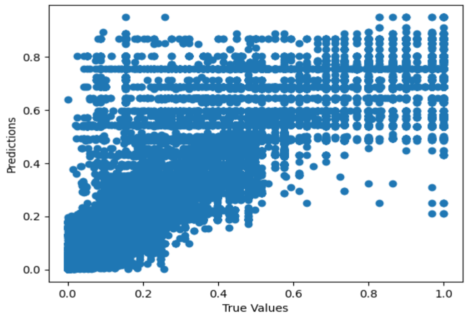
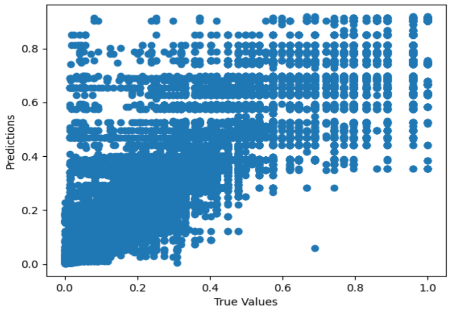
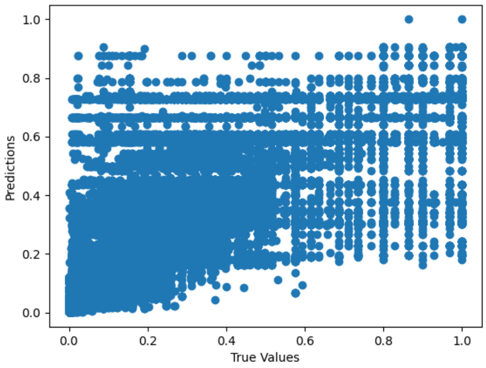
Different models performed very well. Gradient boosting for classification problem and linear regression performed perfectly. Error measure were very low.

Experiment 4 - Lower number of attributes during prediction (10 attributes)

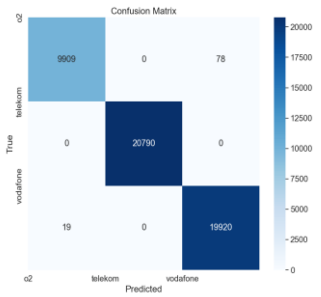
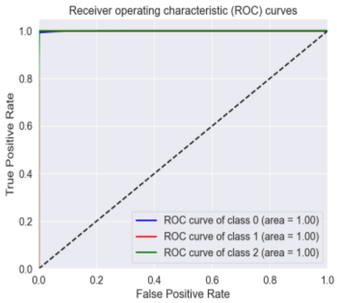
Experiments for all attributes were already performed in previous experiments. Experiment on ten attributes acting as predictors - 'mcsindex','longitude', 'latitude', speed','rsrq','rsrp','rssi','earfcn','cqi','mobiProv_name'

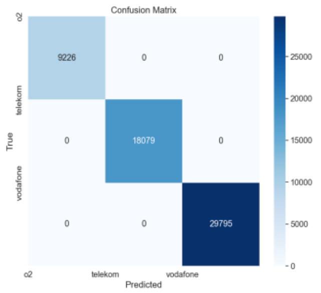
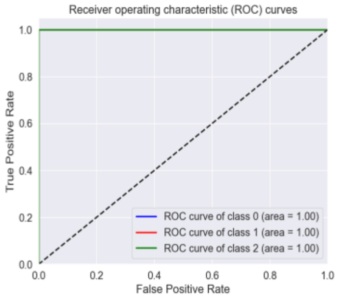
//in the table there is a number of attributes equal 12 but in reality this is 10 attributes and higher number is due to one-hot encoding applied to some attributes.

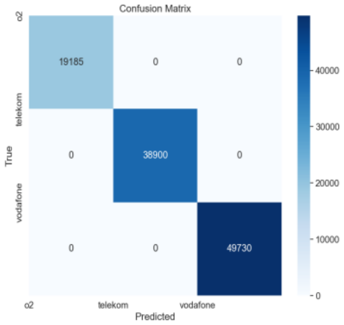
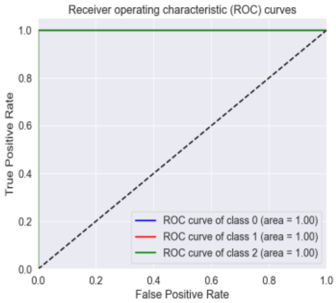
model -decision tree regression, target - throughput'

dataset cleaned and scaled 'downloads' number of attribute (predictors) - 12 MSE: 0.0205 RMSE: 0.1432 R-squared: 0.7717	
dataset cleaned and scaled 'uploads' number of attribute (predictors) - 12 MSE: 0.0111 RMSE: 0.1057 R-squared: 0.8355	
dataset cleaned and scaled 'merged' number of attribute (predictors) - 12 MSE: 0.0161 RMSE: 0.127 R-squared: 0.7341	

model - decision tree classifier, target - 'mobiProv_name'
//on 9 attributes because no mobiProv_name is the target.

dataset cleaned and scaled ‘downloads’ number of attribute (predictors) - 9 Accuracy: 0.9980 Precision: 0.9980 Recall: 0.99808 F1 Score: 0.99808		

dataset cleaned and scaled ‘uploads’ number of attribute (predictors) - 9 Accuracy: 1.0 Precision: 1.0 Recall: 1.0 F1 Score: 1.0		

dataset cleaned and scaled ‘merged’ number of attribute (predictors) - 9 Accuracy: 1.0 Precision: 1.0 Recall: 1.0 F1 Score: 1.0		

Lower number of attributes aggravate results a little for prediction for ‘throughput for decision tree for regression problem. For classification problem, the results were still very good.

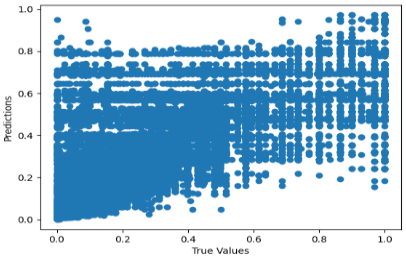
Experiment 5 - Lower number of attributes (5 attributes)

Experiment on five attributes acting as predictors - 'speed','rsrq','rsrp','rssi','earfcn'.

model -decision tree regression, target - throughput’

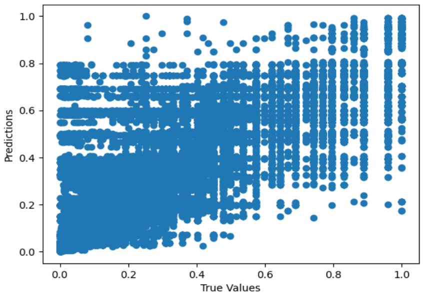
dataset cleaned and scaled ‘downloads’
number of attribute (predictors) - 5

MSE: 0.0312
RMSE: 0.1767
R-squared: 0.6522



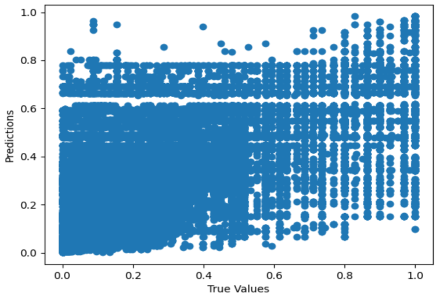
dataset cleaned and scaled ‘uploads’
number of attribute (predictors) - 5

MSE: 0.0166
RMSE: 0.1289
R-squared: 0.7556



dataset cleaned and scaled ‘merged’
number of attribute (predictors) - 5

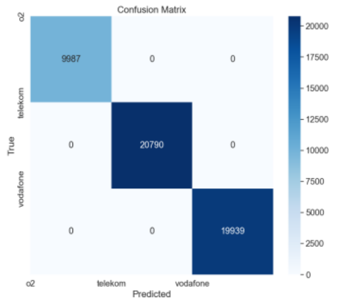
MSE: 0.0228
RMSE: 0.1511
R-squared: 0.6238

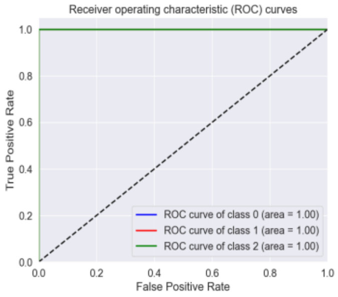


model - decision tree classifier, target - 'mobiProv_name'

dataset cleaned and scaled ‘downloads’
number of attribute (predictors) - 5

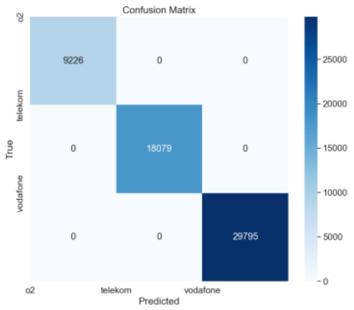
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1 Score: 1.0

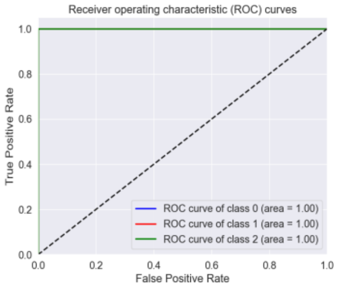




dataset cleaned and scaled ‘uploads’
number of attribute (predictors) - 5

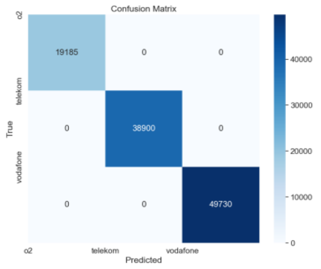
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1 Score: 1.0

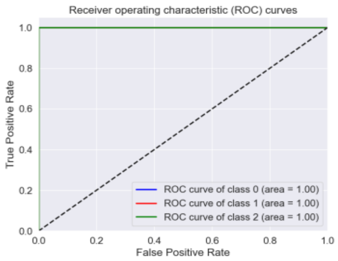




dataset cleaned and scaled ‘merged’
number of attribute (predictors) - 5

Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1 Score: 1.0

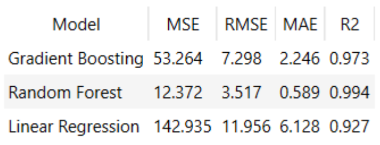
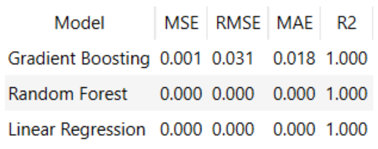
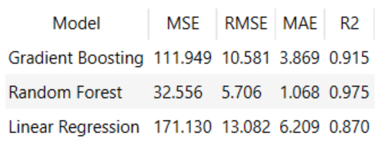




Experiment with lowest number of attributes, which is 5, aggravated results for decision tree ‘throughput’ prediction even more. Test on classification problems was still very good. It would indicate that the model learned patterns that recognize network providers easily.

Experiment 6 - Predicting with Orange tool

Target: tp_claened

dataset cleaned ‘downloads’	dataset cleaned ‘uploads’	dataset cleaned ‘merged’
		

dataset cleaned ‘downloads’

Model	MSE	RMSE	MAE	R2
Gradient Boosting	53.264	7.298	2.246	0.973
Random Forest	12.372	3.517	0.589	0.994
Linear Regression	142.935	11.956	6.128	0.927

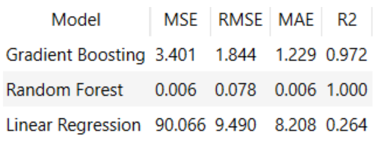
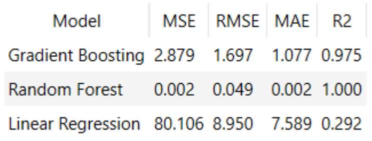
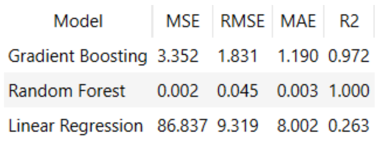
dataset cleaned ‘uploads’

Model	MSE	RMSE	MAE	R2
Gradient Boosting	0.001	0.031	0.018	1.000
Random Forest	0.000	0.000	0.000	1.000
Linear Regression	0.000	0.000	0.000	1.000

dataset cleaned ‘merged’

Model	MSE	RMSE	MAE	R2
Gradient Boosting	111.949	10.581	3.869	0.915
Random Forest	32.556	5.706	1.068	0.975
Linear Regression	171.130	13.082	6.209	0.870

Target: speed

dataset cleaned ‘downloads’	dataset cleaned ‘uploads’	dataset cleaned ‘merged’
		

dataset cleaned ‘downloads’

Model	MSE	RMSE	MAE	R2
Gradient Boosting	3.401	1.844	1.229	0.972
Random Forest	0.006	0.078	0.006	1.000
Linear Regression	90.066	9.490	8.208	0.264

dataset cleaned ‘uploads’

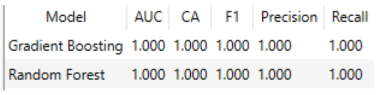
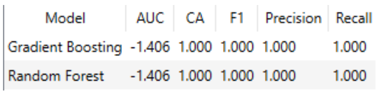
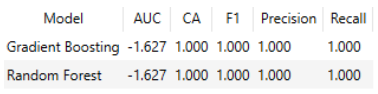
Model	MSE	RMSE	MAE	R2
Gradient Boosting	2.879	1.697	1.077	0.975
Random Forest	0.002	0.049	0.002	1.000
Linear Regression	80.106	8.950	7.589	0.292

dataset cleaned ‘merged’

Model	MSE	RMSE	MAE	R2
Gradient Boosting	3.352	1.831	1.190	0.972
Random Forest	0.002	0.045	0.003	1.000
Linear Regression	86.837	9.319	8.002	0.263

Target: mobiProv_name

//Linear Regression is not available in this test.

dataset cleaned ‘downloads’	dataset cleaned uploads’	dataset cleaned merged’
		

dataset cleaned ‘downloads’

Model	AUC	CA	F1	Precision	Recall
Gradient Boosting	1.000	1.000	1.000	1.000	1.000
Random Forest	1.000	1.000	1.000	1.000	1.000

dataset cleaned uploads’

Model	AUC	CA	F1	Precision	Recall
Gradient Boosting	-1.406	1.000	1.000	1.000	1.000
Random Forest	-1.406	1.000	1.000	1.000	1.000

dataset cleaned merged’

Model	AUC	CA	F1	Precision	Recall
Gradient Boosting	-1.627	1.000	1.000	1.000	1.000
Random Forest	-1.627	1.000	1.000	1.000	1.000

Created pipeline in Orange tool was saved in .ows format and attached to the project. Experiment was conducted on different targets - ‘throughput’, ‘tp_cleaned’ and ‘mobiProv_name’ and on different datasets that were presented in previous sections and are presented in the table above. Also to conduct prediction different models were used: Gradient Boositing, Random Forest and Linear Regression. Linear Regression was not used in classification problems relating to the target ‘mobiProv_name’ - mobile network provider. Results were similar and as good as previous experiments except prediction of ‘speed’ for linear regression which was little worse.

Conclusions

Firstly, datasets were briefly analyzed to discover the meaning of data, problems, obvious patterns and relations in order to plan a way of working with these datasets. Next project pipeline and program structure were established. Program was written in python and loading, cleaning, analyzing, scaling and predicting were performed. These steps were repeated through different experiments using different variables. Results were saved, evaluated and presented.

At the cleaning stage missing data, duplicates and outliers were diagnosed. It was decided to handle duplicates and to leave outliers because of their very small percentage of occuring.

At the analysis stage some dependencies and relation between attributes were discovered. Attributes 'tbs0' and ' tbs1' are in strong correlation with 'tp_cleaned' and 'throughput'. After merging, high correlation can be observed in attributes like chipsettime and gpstime, cellid and earfcn, rsrp and rssi, tp_cleaned and tbs, tp_cleaned and throughput. Analysis part was performed on loaded data 'downloads', 'uploads' and both merged datasets.

Scaling data was performed giving user possibility to either normalize or standardize numeric data.

At the prediction and evaluation stage targets, attributes and models were chosen and prediction was performed through different experiments. Choosing models for prediction was dependent on the target. It was either a regression problem or classification problem. Prediction targets were 'throughput' or 'tp_cleaned', 'speed' and 'mobiProv_name' which was for the name of a mobile network provider. Experiments involved, changing different datasets, changing different models, setting lower number of attributes and the last experiment was done in another environment - Orange tool. Results were saved and some visualizations were made. Different models performed very well, with some models achieving perfect results. The likelihood of overfitting was low, as the models were likely recognizing patterns and dependencies within the attributes, and similar results across different models and on testing data confirmed this. The preprocessing phase was detailed and optimized the data for machine learning models, which led to a successful training and learning experience.