The (un?) reasonable effectiveness of data

This goal of this task is twofold. 1) Get familiar with scikit-learn and some of its machine-learning algorithms; 2) Find out how the effectiveness of machine learning differs for simple and complex tasks on different datasets and dataset sizes. More precisely, your goal is to answer the following questions:

1. To what extent does the effectiveness of machine-learning algorithms depend on the size (i.e. number of instances) and complexity (i.e. number of features) of the data? [200-300 words]
2. Think about how well a rule-based algorithm would have performed on "The SUM dataset (without noise)" for predicting a) the target value and b) the target class. Answer the question: How well would the performance of that rule-based algorithm be compared to the best performing machine-learning algorithm that you tried? [100 words max]

Details:

1. Use scikit-learn.
2. Choose two regression and two classification algorithms
    a. Regression
        i. Linear Regression
        ii. One algorithm of your choice
    b. Classification
        i. Logistic Regression
        ii. One algorithm of your choice
3. Choose four datasets, each with at least 100,000 instances
    a. One dataset of your choice with a low complexity (maximum of 5 features – you can also use a dataset with more features, but only pick 5 features or less)
    b. The SUM dataset (without noise)
    c. The SUM dataset (with noise)
    d. One dataset of your choice with a medium complexity (around 30+ features)
4. Divide your datasets into chunks of 100; 500; 1,000; 5,000; 10,000; 50,000; 100,000; 500,000; 1,000,000; 5,000,000; 10,000,000; 50,000,000; 100,000,000 instances.
    a. If a dataset has e.g. 9.x million entries, feel free to consider this equivalent to 10 million
    b. Always use the first x instances from the dataset to create the chunks, i.e. do not pick randomly e.g. 100k instances from the whole dataset but take the first 100k instances.
5. Apply the machine learning algorithms with each chunk (consider each chunk as separate dataset on which you train and test your algorithms). Keep in mind that even a dataset for regression usually can be used for classification (introduce your own classes if necessary), and vice versa.
    a. If you discover that it would take too long to train an algorithm on one of the large chunks (or the model wouldn't fit into your memory), feel free to not use that algorithm on that chunk.
    b. Use a 10-fold cross validation. For the very large chunks: If the 10-fold cross method causes the training and evaluation to take too long, use a 70/30 split instead (and mention it in your report)
    c. Use RMSE (if possible) and another metric of your choice for the evaluation of the regression algorithms

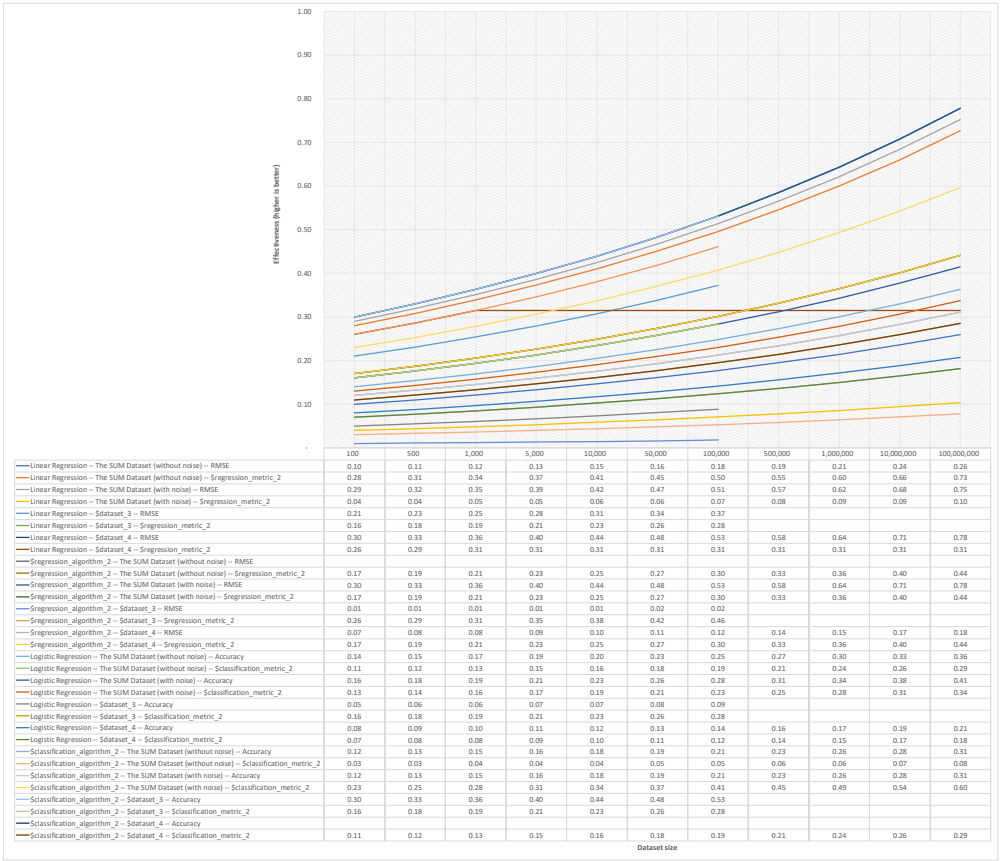
d. Use accuracy (if possible) and another metric of your choice for the classification algorithms.

The expected deliverables are

1. A CSV file with your results.
   a. The content of the CSV file must follow exactly (!) as specified in the example file.

	100	500	1000	5000	10000	50000	100000	500000	1000000	10000000	100000000
Linear_Regression; The SUM Dataset (without noise); RMSE	0.14	0.154	0.1694	0.18634	0.204974	0.225471	0.248019	0.27282	0.300102	0.330113	0.363123944
Linear_Regression; The SUM Dataset (without noise); $regression_metric_2	0.01	0.011	0.0121	0.01331	0.014641	0.016105	0.017716	0.019487	0.021436	0.023579	0.025937425
Linear_Regression; The SUM Dataset (with noise); RMSE	0.11	0.121	0.1331	0.14641	0.161051	0.177156	0.194872	0.214359	0.235795	0.259374	0.285311671
Linear_Regression; The SUM Dataset (with noise); $regression_metric_2	0.1	0.11	0.121	0.1331	0.14641	0.161051	0.177156	0.194872	0.214359	0.235795	0.259374246
Linear_Regression; $dataset_3; RMSE	0.29	0.319	0.3509	0.38599	0.424589	0.467048	0.513753	#N/A	#N/A	#N/A	#N/A
Linear_Regression; $dataset_3; $regression_metric_2	0.17	0.187	0.2057	0.22627	0.248897	0.273787	0.301165	#N/A	#N/A	#N/A	#N/A
Linear_Regression; $dataset_4; RMSE	0.19	0.209	0.2299	0.25289	0.278179	0.305997	0.336597	0.370256	0.407282	0.44801	0.492811067
Linear_Regression; $dataset_4; $regression_metric_2	0.11	0.121	0.1331	0.1331	0.1331	0.1331	0.1331	0.1331	0.1331	0.1331	0.1331
$Regression_Algorithm_2; The SUM Dataset (without noise); RMSE	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
$Regression_Algorithm_2; The SUM Dataset (without noise); $regression_metric_2	0.26	0.286	0.3146	0.34606	0.380666	0.418733	0.460606	0.506666	0.557333	0.613066	0.67437304
$Regression_Algorithm_2; The SUM Dataset (with noise); RMSE	0.3	0.33	0.363	0.3993	0.43923	0.483153	0.531468	0.584615	0.643077	0.707384	0.778122738
$Regression_Algorithm_2; The SUM Dataset (with noise); $regression_metric_2	0.14	0.154	0.1694	0.18634	0.204974	0.225471	0.248019	0.27282	0.300102	0.330113	0.363123944
$Regression_Algorithm_2; $dataset_3; RMSE	0.26	0.286	0.3146	0.34606	0.380666	0.418733	0.460606	#N/A	#N/A	#N/A	#N/A
$Regression_Algorithm_2; $dataset_3; $regression_metric_2	0.03	0.033	0.0363	0.03993	0.043923	0.048315	0.053147	#N/A	#N/A	#N/A	#N/A
$Regression_Algorithm_2; $dataset_4; RMSE	0.06	0.066	0.0726	0.07986	0.087846	0.096631	0.106294	0.116923	0.128615	0.141477	0.155624548
$Regression_Algorithm_2; $dataset_4; $regression_metric_2	0.07	0.077	0.0847	0.09317	0.102487	0.112736	0.124009	0.13641	0.150051	0.165056	0.181561972
Logistic_Regression; The SUM Dataset (without noise); $Accuracy	0.06	0.066	0.0726	0.07986	0.087846	0.096631	0.106294	0.116923	0.128615	0.141477	0.155624548
Logistic_Regression; The SUM Dataset (without noise); $classification_metric_2	0.09	0.099	0.1089	0.11979	0.131769	0.144946	0.15944	0.175385	0.192923	0.212215	0.233436821
Logistic_Regression; The SUM Dataset (with noise); $Accuracy	0.22	0.242	0.2662	0.29282	0.322102	0.354312	0.389743	0.428718	0.47159	0.518748	0.570623341
Logistic_Regression; The SUM Dataset (with noise); $classification_metric_2	0.11	0.121	0.1331	0.14641	0.161051	0.177156	0.194872	0.214359	0.235795	0.259374	0.285311671
Logistic_Regression; $dataset_3; $Accuracy	0.15	0.165	0.1815	0.19965	0.219615	0.241577	0.265734	#N/A	#N/A	#N/A	#N/A
Logistic_Regression; $dataset_3; $classification_metric_2	0.03	0.033	0.0363	0.03993	0.043923	0.048315	0.053147	#N/A	#N/A	#N/A	#N/A
Logistic_Regression; $dataset_4; $Accuracy	0.05	0.055	0.0605	0.06655	0.073205	0.080526	0.088578	0.097436	0.107179	0.117897	0.129687123
Logistic_Regression; $dataset_4; $classification_metric_2	0.06	0.066	0.0726	0.07986	0.087846	0.096631	0.106294	0.116923	0.128615	0.141477	0.155624548
$Classification_Algorithm_2; The SUM Dataset (without noise); $Accuracy	0.11	0.121	0.1331	0.14641	0.161051	0.177156	0.194872	0.214359	0.235795	0.259374	0.285311671
$Classification_Algorithm_2; The SUM Dataset (without noise); $classification_metric_2	0.23	0.253	0.2783	0.30613	0.336743	0.370417	0.407459	0.448205	0.493025	0.542328	0.596560766
$Classification_Algorithm_2; The SUM Dataset (with noise); $Accuracy	0.05	0.055	0.0605	0.06655	0.073205	0.080526	0.088578	0.097436	0.107179	0.117897	0.129687123
$Classification_Algorithm_2; The SUM Dataset (with noise); $classification_metric_2	0.28	0.308	0.3388	0.37268	0.409948	0.450943	0.496037	0.545641	0.600205	0.660225	0.726247889
$Classification_Algorithm_2; $dataset_3; $Accuracy	0.2	0.22	0.242	0.2662	0.29282	0.322102	0.354312	#N/A	#N/A	#N/A	#N/A
$Classification_Algorithm_2; $dataset_3; $classification_metric_2	0.14	0.154	0.1694	0.18634	0.204974	0.225471	0.248019	#N/A	#N/A	#N/A	#N/A
$Classification_Algorithm_2; $dataset_4; $Accuracy	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
$Classification_Algorithm_2; $dataset_4; $classification_metric_2	0.07	0.077	0.0847	0.09317	0.102487	0.112736	0.124009	0.13641	0.150051	0.165056	0.181561972

   b. The file name must follow exactly (!) the pattern “task1, $team_id, performance by dataset size, data.csv”, e.g. “task1, team_01, performance by dataset size, data.csv”
2. A PNG image illustrating your results, and named “task1, $team_id, performance by dataset size, chart.png”, e.g. “task1, team_55, performance by dataset size, chart.png”. The image might look like:



You can create the image either directly with scikit-learn (it should look roughly like the

example); or after creating the CSV file, you just copy and paste the results to the Excel template, which automatically creates an image (you only need to store it as PNG then).

3. A PDF report that
    a. Contains the team id, individual student IDs, and the total time that was spent to complete the task
    b. Answers the above-mentioned questions
    c. Contains the created image (please ensure that the quality of the image is good. It doesn't matter if we must zoom in e.g. 400%, but the numbers etc. must be readable at some zoom level).
    d. Provides a brief overview (table) of the used datasets, algorithms, etc.
    e. Lists the contributions of all team members
    f. Contains additional information if required
    g. Is in about the same format as the report-template and named "task1, $team_id, performance by dataset size, report.pdf"
4. Your source code (no compiled/binary files) in the sub folder /task1, $team_id, performance by dataset size, source code/. Ensure you follow the "developer guidelines".