Completely Missing the Question/Guidelines

Question 1: To what extent does the effectiveness of machine-learning algorithms depend on the size and complexity of the data? [200-300 words]

> *"We split the data into train and test, both of them are a sample/part of the whole data. An Immediate thing that comes to mind is that "more data means better metrics". Let us consider an example for estimating the age of all the teenagers in a country. Having a small random sample often would be enough if you assume it is distributed normally. On the other case consider an example to detect the facial features of each and every person in a country. You would require a lot of data since the facial features of each and every person is unique. Big samples will be needed to unveil the useful needed to predict the outcome and throw away data which are not useful, So the effectiveness of size on data depends upon the type of data and work that has to be done on data. But the size of the data is not only an important factor. Sometimes the response variables that you want to estimate can be complex. If your model has Polynomial relation between features and if you are trying to fit a linear model, it would be an impossible task other than some particular cases. So other than size of data, complexity of the model also plays an important role in determining the response variables. There are no general ways to tell the exact size of data that you would require for a particular model. This is the reason that a particular algorithm has different metrics for different size of the train and test data as shown in the results above."*

➔ Reads like a textbook or repetition from the lecture. The answer is not related to the students' own work.

Given the general guidelines, marking scheme and instructions for task 1, I did mark that answer with 0.

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. Looking only at the performance of your best performing machine-learning algorithm on "The SUM dataset (without noise)": how well was machine-learning suitable to solve the task of predicting a) the target value and b) the target class? Consider in your assessment, how well a simple rule-based algorithm would have performed. [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,~~ one with at least 10,000 instances, and three with at least 100,000 instances (ideally, all four have 100,000+

A good answer to a research question:

1. Describes the results objectively. For instance, "Our analysis showed that algorithm B performed better than algorithm A, regardless of the metric being used." There should be no room for discussions here. If someone would say "I disagree", then either you or the 'someone' would have done something severely wrong. It is also important that you are as specific as possible. Do not write "As can be seen from the table, algorithm A performed better than algorithm B". Instead, be specific and write e.g. "As can be seen from the table, algorithm A performed better than algorithm B with a precision of 0.67 vs. 0.82".
2. Discusses the findings and draws (potentially subjective) conclusions. For instance, "Based on the results, it seems that algorithm A is more suitable for the classification of images than algorithm B". Of course, ideally, people will agree with your conclusions, but it would be ok in some circumstances, if people would draw different conclusions or say "But..."
3. States its limitations. For instance, "However, it should be kept in mind that we only used low-resolution images. It would be interesting to see how the algorithms perform on high resolution images".

specific as possible. Do not write "As can be seen from the table, algorithm A performed better than algorithm B". Instead, be specific and write e.g. "As can be seen from the table, algorithm A performed better than algorithm B with a precision of 0.67 vs. 0.82".

Describe what you see, and summarize

Most simple way:

- *Algorithm A had an accuracy of 0.3 and a precision of 0.5*
- *Algorithm B had an accuracy of 0.2 and a precision of 0.6*
- *Algorithm C had an accuracy of 0.7 and a precision of 0.8*
- *Algorithm D…*
- *…*
- *Algorithm Z…*

Commented [JB1]: This takes long to read and provides rather little information.

Higher level

- *Algorithm A performed better than algorithms B, E, F, and Z when measured with accuracy*
- *Algorithm B performed better than A, and G when measured with precision*
- *Overall, algorithm C performed best, regardless of which metric was used.*

Commented [JB2]: Less text, yet more information and more relevant information.

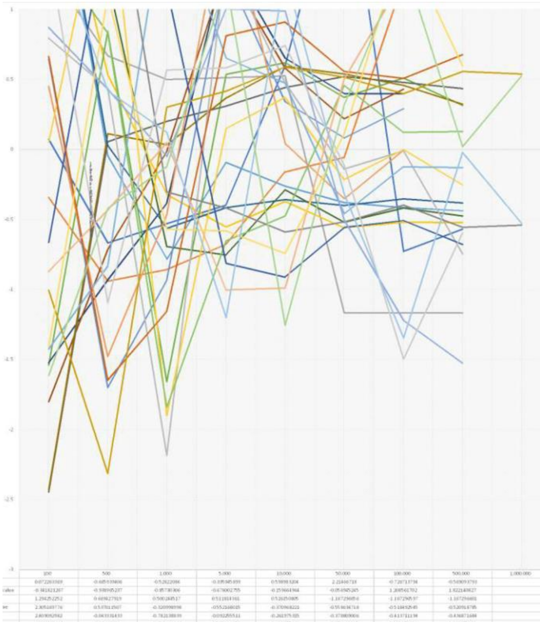
Even higher

- *In some situations, the ranking of algorithms differed based on the metric being used. For instance, …. In other situations, …*

Commented [JB3]: This is the piece of information that is really relevant.

Consistency of Descriptions with the actual chart

Example: “As can be seen”



As can be seen in the graph the larger and more complex datasets tended to create more efficient algorithm outputs as their size grew

Commented [JB4]: From that chart, nothing can be seen.

Example: Plain Wrong

Logistic Regression - The SUM dataset without noise - Accuracy	0.8	0.88	0.89	0.858	0.868	0.8635	0.85995	0.86472	0.864424899	0.864362925

Let us consider a simple example of applying Logistic Regression on “The SUM without noise” dataset. ***We can see the upward trends in the graph*** *(w.r.t accuracy)* ***as the chunks size increases****.*

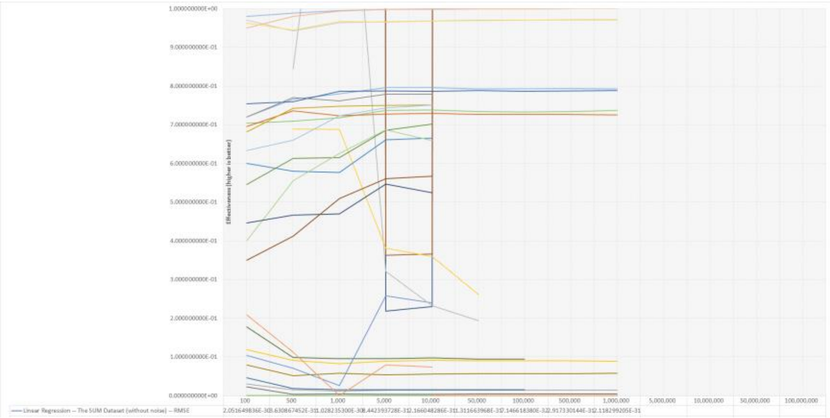
Commented [JB5]: This is wrong. The accuracy remains stable from 500 instances onwards

Example:

> *Size of the dataset has a huge impact on machine learning algorithms, and behaves differently when the size of the chunk increases. Lets take case of logistic regression in Skin_nonskin data set when we calculate accuracy it increases from 0.8826 to 0.9197 as the size.*

Commented [JB10]: Is this really a huge difference (4% increase)? However, this demonstrate the importance to provide specific numbers, so readers can judge themselves if the difference is "huge" or not.

Describe things that are directly striking



How come that the brown line is vertical; like a rectangle?

	4.5542E-10	3.32134E-10	3.70007E-10	3.95142E-10	5.13039E-10	5.84186E-10	9.45784E-10	1.12355E-09	2.78781E-09	5,(
Linear_Regression; The SUM Dataset (without noise); r2_score	1	1	1	1	1	1	1	1	1	
Linear_Regression; The SUM Dataset (with noise); RMSE	0.112415015	0.081220603	0.070511874	0.070225483	0.069467547	0.069826501	0.070037207	0.069901669	0.069878816	
Linear_Regression; The SUM Dataset (with noise); r2_score	0.974468156	0.98323711	0.984888517	0.985521295	0.985487037	0.985742867	0.985602871	0.985634536	0.985637482	
Linear_Regression; Skin_NonSkin; RMSE	0	0	0	0	0	0	0.300798054	0.312561427		
Linear_Regression; Skin_NonSkin; r2_score	1	1	1	1	1	1	-0.075366098	-0.069858938		
Linear_Regression; YearPredictionMSD; RMSE	4.086243218	6.902232814	12.18166748	9.384081992	9.22811069	9.623874252	9.720619225	9.730498832	9.554946351	
Linear_Regression; YearPredictionMSD; r2_score	-10.91526803	-1.31353382	-1.072157629	0.142051043	0.143525934	0.174759473	0.193941768	0.234719767	0.205475407	
Decision_Tree_Regression; The SUM Dataset (without noise); RMSE	0.07071951	0.010950226	0.006522655	0.001225498	0.000590895	0.000590895	5.94E-05	2.376142618	1.16E-05	
Decision_Tree_Regression; The SUM Dataset (without noise); r2_score	0.23848634	0.999962083	0.99998977	0.999999626	0.999999903	0.999999996	0.999999999	1	1	
Decision_Tree_Regression; The SUM Dataset (with noise); RMSE	0.165901012	0.135183673	0.109920497	0.112129169	0.106821425	0.104037729	0.104102676	0.102015545	0.101349951	
Decision_Tree_Regression; The SUM Dataset (with noise); r2_score	0.898191163	0.96178746	0.967521507	0.96619129	0.966972708	0.969327037	0.969220288	0.969743558	0.969981797	
Decision_Tree_Regression; Skin_NonSkin; RMSE	1	1	1	1	1	1	0.121192985	0.160750115		
Decision_Tree_Regression; Skin_NonSkin; r2_score	0	0	0	0	0	0	0.189047542	0.174469972		
Decision_Tree_Regression; YearPredictionMSD; RMSE	5.527358613	9.623717503	18.32805016	13.43972755	13.28015997	13.6731939	13.58577275	13.46125977		
Decision_Tree_Regression; YearPredictionMSD; r2_score	-22.35144959	-2.330481039	-4.6518156	-0.882966657	-0.804607284	-0.675982535	-0.581905348	-0.521727405		
Logistic_Regression; The SUM Dataset (without noise); Accuracy	0.905505051	0.896161418	0.907241307	0.904606888	0.903805684	0.904040512	0.904479958	0.905323987	0.905105	
Logistic_Regression; The SUM Dataset (without noise); f1_macro	0.743544892	0.74693199	0.715051917	0.617340541	0.589111612	0.579427572	0.621187287	0.687006545	0.62385	
Logistic_Regression; The SUM Dataset (with noise); Accuracy	0.927373737	0.905885308	0.90299287	0.898023504	0.897202108	0.898620345	0.897770049	0.898385986	0.89807	
Logistic_Regression; The SUM Dataset (with noise); f1_macro	0.800521156	0.737251095	0.68389729	0.635627241	0.573140538	0.568741079	0.553335251	0.667598206	0.611041	
Logistic_Regression; Skin_NonSkin ;Accuracy							0.897172196	0.914473004		
Logistic_Regression; Skin_NonSkin; f1_macro							0.895980173	0.869335346		
Logistic_Regression; YearPredictionMSD; Accuracy	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]				
Logistic_Regression; YearPredictionMSD; f1_macro	0.332898	0.26740443	0.118401961	0.053513094	0.028084819	0.014144576				
Decision_tree_Classifier; The SUM Dataset (without noise); Accuracy	0.99	0.998	0.996	1	1	0.99998	1	0.999998	1	
Decision_tree_Classifier; The SUM Dataset (without noise); f1_macro	0.655555556	0.886666667	0.792053044	0.875	0.975	0.799354839	0.84	0.879885342	0.94	
Decision_tree_Classifier; The SUM Dataset (with noise); Accuracy	0.97	0.96	0.956	0.959	0.9585	0.95716	0.95569	0.956606	0.956471	
Decision_tree_Classifier; The SUM Dataset (with noise); f1_macro	0.607395941	0.82796031	0.737508964	0.790059857	0.906969002	0.731683577	0.767766177	0.812956075	0.87025	
Decision_tree_Classifier; Skin_NonSkin; accuracy	1	1	1	1	1	1	0.978351798	0.988770069		
Decision_tree_Classifier; Skin_NonSkin; f1_macro	1	1	1	1	1	1	0.97793224	0.981841041		
Decision_tree_Classifier; YearPredictionMSD; Accuracy	0.276010101	0.297640813	0.178417752	0.07926813	0.066480651	0.053278884	0.054767107	0.056413		
Decision_tree_Classifier; YearPredictionMSD; f1_macro	0.180243575	0.205755999	0.123528292	0.070130007	0.052051929	0.028034329	0.031036155	0.038724		
#NAME?	0.18	0.20	0.22	0.24	0.26	0.29	0.32	0.35	0.39	

Dataset size

Why are there no numbers?

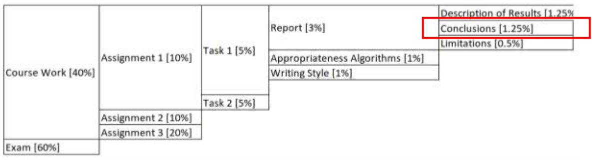
Describe only relevant information

> *In terms of usability of the frameworks, the easiest to use was Weka, as all you were required to do was interact with a GUI which takes care of the software. In terms of coding, it was*

Commented [JB11]: Although interesting to know, this does not relate to the question and does not contribute towards the marks.

much easier to use sklearn, as it has pre-defined libraries for all the regressions, classifications etc.

Conclusions



From the guidelines that I gave you:

> Discusses the findings and draws (potentially subjective) conclusions. For instance, "Based on the results, it seems that algorithm A is more suitable for the classification of images than algorithm B". Of course, ideally, people will agree with your conclusions, but it would be ok in some circumstances, if people would draw different conclusions or say "But…"

Question 1: To what extent does the effectiveness of machine-learning algorithms depend on the size and complexity of the data? [200-300 words]

Good answer

> *The size of the dataset has only an effect on machine-learning performance until a certain threshold, and for most datasets this threshold is rather low. […] The complexity of the data seemed to have a negative impact on performance, especially when used with simple algorithms. […]*

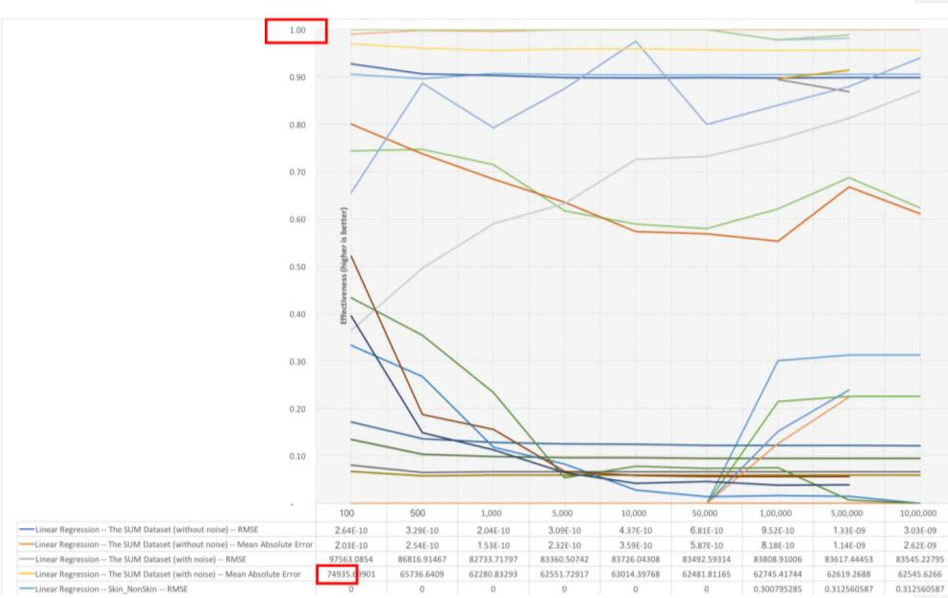
Commented [JB12]: Good answer to the first part of the question (size)

Commented [JB13]: Good answer to the second part of the question (complexity)

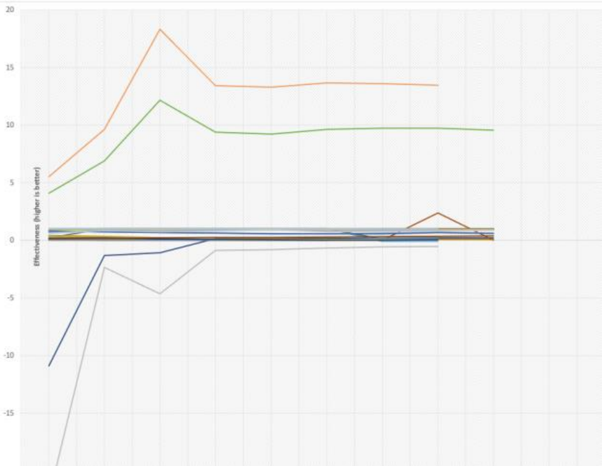
Misc

Formatting / Layout Errors

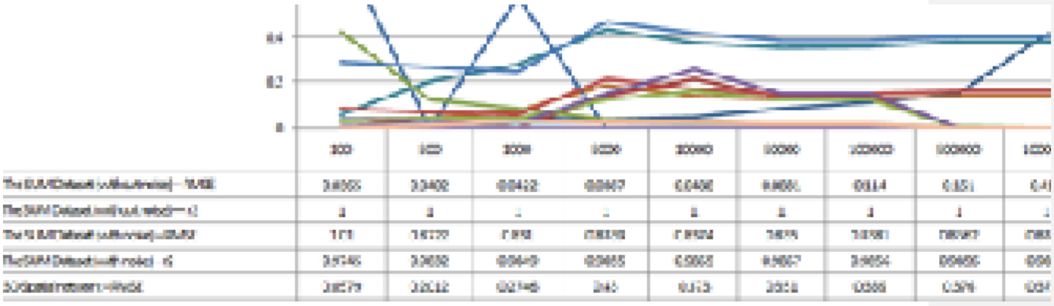
<u>Scaling (numbers not being displayed in the chart)</u>



<u>Scaling (2)</u>



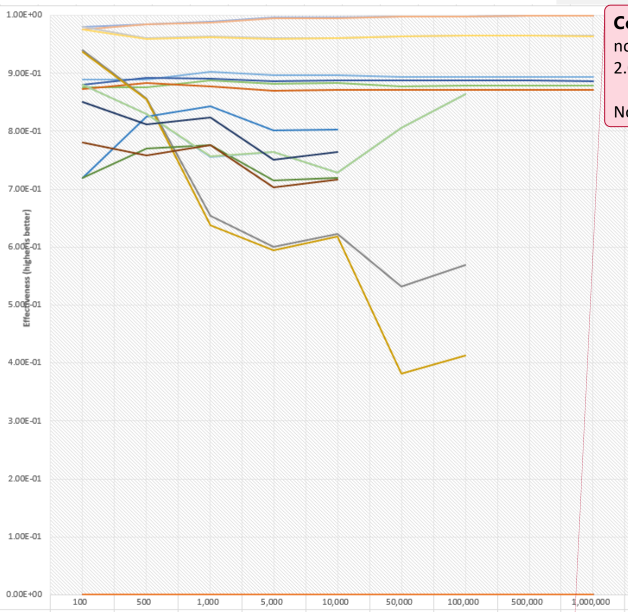
<u>Quality of Images</u>



<u>Thousand-Separator</u>

-1	100	500	1,000	5,000	10,000	50,000	1,00,000	5,00,000	10,00,000	50,00,000
	0	0	0	0	0	0	0	0		

<u>Number Formats</u>



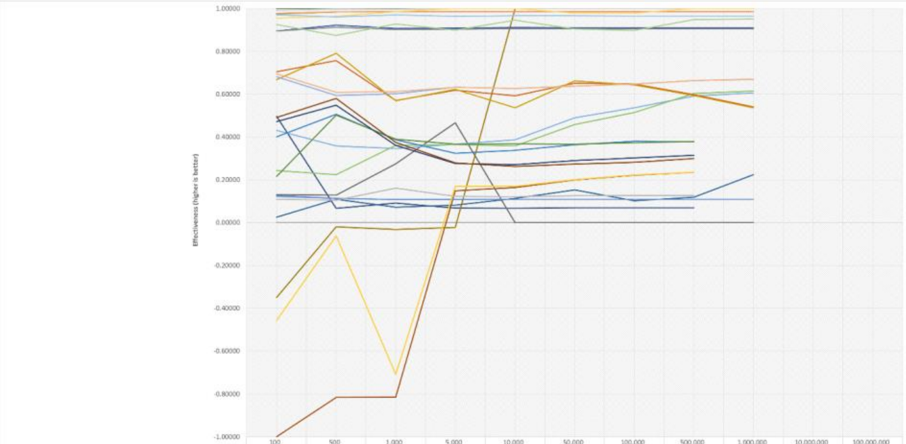
	100	500	1,000	5,000	10,000	50,000	100,000	500,000	1,000,000
Linear Regression -- The SUM Dataset (without noise) -- RMSE	5.15E-10	3.92E-10	3.90E-10	4.34E-10	5.80E-10	7.65E-10	1.16E-09	1.27E-09	3.91E-09
Linear Regression -- The SUM Dataset (without noise) -- MSE	2.65E-19	1.54E-19	1.52E-19	1.89E-19	3.36E-19	5.85E-19	1.35E-18	1.61E-18	1.53E-17
Linear Regression -- The SUM Dataset (with noise) -- RMSE	100966.2067	87215.59732	83101.82439	83389.6154	83744.85148	83496.65896	83811.30682	83618.3378	83545.5908
Linear Regression -- The SUM Dataset (with noise) -- MSE	10194174898	7606560416	6905913217	6953827956	7013200150	6971692057	7024335151	6992026416	6979865743
Linear Regression -- 3D Spatial Network -- RMSE	8.027454702	19.40951122	18.44380719	17.74658718	18.41490399	18.23942627	18.7031596		
Linear Regression -- 3D Spatial Network -- MSE	64.44002899	376.7291259	340.1740237	314.9413567	339.1086891	332.6766707	349.8081789		
Linear Regression -- Online News Popularity -- RMSE	234702236.1	3247718.572	1394722.664	11690.65262	14518.4907				
Linear Regression -- Online News Popularity -- MSE	5.51E+16	1.05E+13	1.95E+12	136671358.6	210786572.1				
Bayesian Regression -- The SUM Dataset (without noise) -- RMSE	49.2354196	71.16380456	56.34528054	36.75146049	16.18741114	55.82662274	52.51851494	58.35926421	35.77182214
Bayesian Regression -- The SUM Dataset (without noise) -- MSE	2424.126543	5064.28708	3174.790639	1350.669848	262.0322794	3116.611806	2758.194411	3405.803719	1279.623259
Bayesian Regression -- The SUM Dataset (with noise) -- RMSE	96183.49931	86934.06366	83039.53482	83387.33314	83744.07741	83496.63668	83811.30307	83618.33753	83545.5907
Bayesian Regression -- The SUM Dataset (with noise) -- MSE	9251265539	7557531424	6895564343	6953447329	7013070501	6971688337	7024334522	6992026371	6979865726
Bayesian Regression -- 3D Spatial Network -- RMSE	8.049985021	19.42074919	18.4444199	17.74794658	18.41501647	18.23956944	18.70320846		
Bayesian Regression -- 3D Spatial Network -- MSE	64.80225884	377.165499	340.1966254	314.9896078	339.1128317	332.6818933	349.8100067		
Bayesian Regression -- Online News Popularity -- RMSE	2663.894625	4198.741985	3960.105147	11616.53249	14502.77183				
Bayesian Regression -- Online News Popularity -- MSE	7096334.576	17629434.26	15682432.78	134943827.2	210330390.9				
Logistic Regression -- The SUM Dataset (without noise) -- Accuracy	0.89	0.89	0.902	0.897	0.8965	0.8932	0.8944	0.894376	0.894118072
Logistic Regression -- The SUM Dataset (without noise) -- F1_WEIGHTED	0.876420809	0.876305813	0.887202119	0.88225597	0.883216605	0.877677717	0.879047389	0.879132824	0.878799353
Logistic Regression -- The SUM Dataset (with noise) -- Accuracy	0.88	0.892	0.891	0.887	0.8871	0.8876	0.88763	0.887264	0.886970309
Logistic Regression -- The SUM Dataset (with noise) -- F1_WEIGHTED	0.872605042	0.883161147	0.877892848	0.869897044	0.871032812	0.871416576	0.871461343	0.871026765	0.870674593
Logistic Regression -- 3D Spatial Network -- Accuracy	0.94	0.856	0.654	0.601	0.623	0.5327	0.56914		
Logistic Regression -- 3D Spatial Network -- F1_WEIGHTED	0.937062937	0.855709319	0.637207467	0.595115595	0.61865389	0.381518085	0.41286353		
Logistic Regression -- Online News Popularity -- Accuracy	0.72	0.826	0.843	0.801	0.8027				
Logistic Regression -- Online News Popularity -- F1_WEIGHTED	0.72	0.769749262	0.775821399	0.714471162	0.719063799				
NearestNeighbors Regression -- The SUM Dataset (without noise) -- Accuracy	0.98	0.984	0.989	0.9958	0.9962	0.99822	0.99857	0.999522	0.999633316
NearestNeighbors Regression -- The SUM Dataset (without noise) -- F1_WEIGHTED	0.97502994	0.984303087	0.987854614	0.995068797	0.995656629	0.99817754	0.998555471	0.999516344	0.999623388
NearestNeighbors Regression -- The SUM Dataset (with noise) -- Accuracy	0.98	0.96	0.963	0.9602	0.9606	0.96402	0.96473	0.964494	0.964427482
NearestNeighbors Regression -- The SUM Dataset (with noise) -- F1_WEIGHTED	0.97502994	0.95989665	0.961868837	0.959618528	0.960260591	0.963981957	0.964678481	0.964477913	0.964403528
NearestNeighbors Regression -- 3D Spatial Network -- Accuracy	0.88	0.83	0.756	0.7648	0.7288	0.80592	0.86451		
NearestNeighbors Regression -- 3D Spatial Network -- F1_WEIGHTED	0.88	0.829883552	0.756273718	0.764804967	0.728641484	0.805888849	0.864587875		
NearestNeighbors Regression -- Online News Popularity -- Accuracy	0.85	0.812	0.824	0.7514	0.7645				
NearestNeighbors Regression -- Online News Popularity -- F1_WEIGHTED	0.781081081	0.75899181	0.77622395	0.703189089	0.716465674				

Dataset size

Commented [JB14]: The number format of the y-axis is not helpful.
2.00E-01 vs. 0.2.

Not consistent to write once 0.72 and then 0.7697490213

More Commented Examples



	100	500	1,000	5,000	10,000	50,000	100,000	500,000	1,000,000	10,000,000	100,000,000
Linear_Regression -- New York City Taxi Trip Duration -- RMSE	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.000000		
Linear_Regression -- New York City Taxi Trip Duration -- R2	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.000000		
Linear_Regression -- The SUM dataset/without noise -- RMSE	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.000000		
Linear_Regression -- The SUM dataset/without noise -- R2	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.000000		
Linear_Regression -- The SUM dataset/with noise -- RMSE	0.12731	0.11297	0.10770	0.10841	0.10885	0.10853	0.10894	0.10869	0.108597		
Linear_Regression -- The SUM dataset/with noise -- R2	0.97428	0.98317	0.98489	0.98551	0.98548	0.98574	0.98560	0.98563	0.985637		
Linear_Regression -- YearPredictionMSD -- RMSE	0.49697	0.06717	0.09169	0.06826	0.06671	0.06936	0.06991	0.06992			
Linear_Regression -- YearPredictionMSD -- R2	-1.00000	-0.81580	-0.81497	0.14927	0.16413	0.20008	0.22207	0.23472			
Bayesian_Ridge -- New York City Taxi Trip Duration -- RMSE	0.13067	0.12848	0.27508	0.46577	0.00000	0.00000	0.00000	0.00000	0.000000		
Bayesian_Ridge -- New York City Taxi Trip Duration -- R2	-0.34940	-0.02003	-0.03120	-0.02287	1.00000	1.00000	1.00000	1.00000	1.000000		
Bayesian_Ridge -- The SUM dataset/without noise -- RMSE	0.02564	0.10901	0.07256	0.08116	0.11343	0.15256	0.10264	0.11829	0.224718		
Bayesian_Ridge -- The SUM dataset/without noise -- R2	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.000000		
Bayesian_Ridge -- The SUM dataset/with noise -- RMSE	0.12252	0.11304	0.10830	0.10908	0.10954	0.10922	0.10963	0.10938	0.109288		
Bayesian_Ridge -- The SUM dataset/with noise -- R2	0.97628	0.98334	0.98491	0.98552	0.98549	0.98574	0.98560	0.98563	0.985637		
Bayesian_Ridge -- YearPredictionMSD -- RMSE	0.10981	0.10462	0.16094	0.12297	0.12106	0.12619	0.12720	0.12722			
Bayesian_Ridge -- YearPredictionMSD -- R2	-0.45617	-0.06170	-0.70804	0.16931	0.16879	0.20038	0.22217	0.23472			
Logistic_Regression -- New York City Taxi Trip Duration -- Accuracy	0.43081	0.35792	0.34615	0.36600	0.38759	0.48932	0.53657	0.59134	0.605622		
Logistic_Regression -- New York City Taxi Trip Duration -- Precision_Macro	0.24336	0.22422	0.36108	0.36511	0.35841	0.45806	0.51461	0.60266	0.614507		
Logistic_Regression -- The SUM dataset/without noise -- Accuracy	0.89439	0.92284	0.90746	0.90904	0.91134	0.91031	0.90924	0.91055	0.910393		
Logistic_Regression -- The SUM dataset/without noise -- Precision_Macro	0.70485	0.75659	0.57053	0.61765	0.59250	0.65162	0.64720	0.59754	0.540258		
Logistic_Regression -- The SUM dataset/with noise -- Accuracy	0.89460	0.91256	0.90135	0.90289	0.90593	0.90393	0.90410	0.90477	0.904077		
Logistic_Regression -- The SUM dataset/with noise -- Precision_Macro	0.66818	0.79128	0.56941	0.62233	0.53666	0.66246	0.64304	0.59340	0.536560		
Logistic_Regression -- YearPredictionMSD -- Accuracy	0.40100	0.50638	0.38761	0.32361	0.33819	0.36378	0.37971	0.37762			
Logistic_Regression -- YearPredictionMSD -- Precision_Macro	0.21540	0.50142	0.38943	0.36696	0.36724	0.36606	0.37360	0.37915			
Random_Forest -- New York City Taxi Trip Duration -- Accuracy	0.68076	0.59391	0.60202	0.63200	0.62689	0.63782	0.64724	0.66306	0.66953		
Random_Forest -- New York City Taxi Trip Duration -- Precision_Macro	0.69460	0.60848	0.61133	0.63136	0.62619	0.63710	0.64702	0.66396	0.67014		
Random_Forest -- The SUM dataset/without noise -- Accuracy	0.99187	0.99804	0.99604	1.00000	1.00000	0.99998	0.99999	1.00000	1.00000		
Random_Forest -- The SUM dataset/without noise -- Precision_Macro	0.95556	0.96364	0.98548	1.00000	1.00000	0.97818	0.97900	0.99998	1.00000		
Random_Forest -- The SUM dataset/with noise -- Accuracy	0.97056	0.96003	0.96998	0.96379	0.96580	0.96542	0.96447	0.96494	0.96458		
Random_Forest -- The SUM dataset/with noise -- Precision_Macro	0.92500	0.87434	0.92731	0.89904	0.94582	0.90501	0.89726	0.94849	0.95157		
Random_Forest -- YearPredictionMSD -- Accuracy	0.47165	0.54812	0.36114	0.27580	0.27128	0.29018	0.30248	0.31433			
Random_Forest -- YearPredictionMSD -- Precision_Macro	0.49100	0.57944	0.37503	0.27924	0.26311	0.27302	0.28172	0.29930			

Dataset size

Original Text:

The effectiveness of machine learning algorithms is always related to the complexity and size of the data. However, the extent of this relation changes according to the content and the structure of the dataset.

In most of the cases, after a certain amount of training data size, the quality of the machine learning model remains stable. Therefore, we cannot always say, training the algorithm using more samples gives much better results.

If there is no noise in the dataset, the effect of the training data size decreases. Furthermore, if the dataset is well structured like the SUM datasets, this decrease in the effectiveness of size will be more striking. On the other hand, increase in the noise and complexity affects the performance of machine-learning algorithms negatively and increases the effectiveness and importance of the training data size. Especially if we look at the classification algorithms, with complex datasets we need more samples to make accurate predictions. In conclusion, the performance of machine learning algorithms decreases with more complex and highly

Commented [JB15]: Such a general statement is not relevant to answer the question. Generally, be careful with words like "always" because there are almost always exceptions, which makes your statement wrong.

Commented [JB16]: The structure is good (1. Neutral description; 2. Interpretation/Conclusion). However, I can´t see how the chart would lead to that description/conclusion. To me, it seems that most of the algorithms have the same performance regardless of using 100 or 1,000,000 instances.

Commented [JB17]: How is this statement supported? Specific numbers/example is necessary.

Commented [JB18]: What are the complex datasets? I don´t know all the details of all the 24 datasets in Dropbox.

polluted datasets. We may need to increase the data size to get better performance. With less polluted and complex datasets, the effectiveness of data size decreases and performance of machine-learning algorithms increases.

Commented [JB19]: This might be right, but I cannot really see it from the chart.

My suggestion (focusing only on dataset size, not complexity):

Results: Looking at the results, there are three types of algorithms. First, those with a stable performance regardless of the dataset size. For instance, RMSE and R2 for both the NY Taxi dataset and the SUM dataset (without noise) remain identical (0.0 and 1.0 respectively) for all dataset sizes. Second, those that have a slight decrease/increase with increasing dataset size. For instance, …. Third, there are some cases in which the performance shows a weird trend when measured with R2. Explain…

Conclusion: Based on the results, it seems that for most algorithms the dataset size had a marginal or even no effect on performance.

Limitations: The observation contradicts out expectation that algorithms usually perform better when trained on more data. The following reasons might be causing this discrepancy. First, we always used all features for the training. Since many features are probably of little relevance, they might negatively impact the training when there is lots of data. Second, …

Misc

The performance of the machine learning algorithm with the size of the data is very interesting. We would have assumed that as the number of instances in the data increases, the effectiveness of the machine learning algorithm also increases. This is generally true if the machine learning algorithm is a good fit for your data. But using an ineffective algorithm remains ineffective or can even decrease in performance as the size of the data increases. For example, trying to fit linear regression to a highly non- linear problem. Using more data does not improve the effectiveness of this, in fact more data can highlight just how bad it is. In our chart we can see that applying linear support vector regression to skin non skin fluctuated in performance as the number of instances went up – getting a value of 0.36 and 0.26 for 500 and 100,000 instances respectively.

Commented [JB20]: Stating your hypothesis/assumption is a good idea, especially if your results contradict the assumption.

Commented [JB21]: Also good. A specific example demonstrating that the above made claim ("performance decreases with more data") is actually true.

Misc

Question 1: To what extent does the effectiveness of machine-learning algorithms depend on the size and complexity of the data? [200-300 words]

Commented [JB22]: Very long (but overall good) description of the results. But the "so what" is mostly missing. Considering that you even used 370 words, the answer is not ideal. Btw. exceeding the word limit decreases your marks.

For our analysis we considered diverse datasets and ran four different machine learning algorithms with different metrics. For the first dataset (SUM without Noise), using a chunk of 10,000 instances the regression algorithm Linear Regression gives a RMSE of 5.21E-10 v/s 0.057985 of SVR, whereas Variance is 1 v/s 0.996638. For the same dataset, the classification algorithm Logistic Regression gives an Accuracy of 0.97389739 against 0.99719972 of SVC, *while F1 Score is 0.939138244 v/s 0.996522297. After 10,000 instances we observe that there are no major changes in the metrics scores and at some point the data becomes constant.*

For the Second dataset (SUM with Noise), Linear Regression gave a RMSE of 83737.0267 v/s 0.120505 of SVR, whereas Variance is 0.98550806 v/s 0.985479 while using a chunk of 10000 instances. When we applied classification algorithms on the same data set, Logistic Regression gave an accuracy of 0.949294929 against 0.96779678 of SVC and F1 Score was 0.939138244 v/s 0.967210506. After 1000 instances there were no major changes.

The Third Dataset we used was the 3D Roads Network, which has less than five features. Linear Regression gave a RMSE of 18.41490399 v/s 1.015049 of SVR, and variance was -0.023001494 v/s -0.030324 for 10000 instances. Classification Algorithm Logistic Regression achieved an Accuracy 0.0372 against 0.1941 of SVR, and an F1 score of 0.007532246 v/s 0.199101537. Considering that the RMSE and Variance of the Logistic Regression model is higher than that of SVR, we can conclude that it's an algorithm that improves better with the increase of instances.

The Fourth Dataset we used was the Buzz in Social Media which was having more than 70 features and was therefore the most complex dataset. An RMSE of 192.2731385 was achieved by Linear Regression against an RMSE of 0.195155 achieved by SVR, Variance was 0.960730584 v/s 0.97389739 for 10,000 instances. Logistic Regression gave an Accuracy of 0.0807 and F1 Score of 0.05363639 against SVR's Accuracy of 0.1002 and F1Score of 0.078037143. As the data size increase, we see SVR performing better than Linear Regression. Similarly, SVC was better than Logistic Regression.

Thus to summarize, the complexity increases most in the Buzz in Social Media Dataset, SVR performs the best in Regression and SVC performs the best in classification.

Commented [JB23]: Good to explicitly say how many features the dataset had.

Commented [JB24]: Are the decimals really necessary? Does it really matter if it is *192 or 192.2731385?*

Commented [JB25]: This paragraph and much of the previous text does not really answer the question

Good and even better answer for Task 1, Question 1

Original Text:

> *Overall we believe the size and complexity of the data affect different algorithms differently. In respect to linear regression the low complexity dataset "Skin NonSkin" would provide worse results with increased chunk size decreasing from a perfect RMSE and mean absolute error metrics score of 0.0 in all chunks upto 50,000 to 0.34306 and 0.22578 for chunk sizes 500,000 in RMSE and mean absolute error respectively. This is however different for a high complexity data set with Year Prediction MSD yielding better results for both higher chunk sizes having a rapid improvement of 92.3% in RMSE for chunks sizes of 100 to 500,000 with the values 1.38833 and 0.10732.*

I would have given even more marks if the "so what" had been made more clear. For instance (next text is underlined):

> *Overall we believe the size and complexity of the data affect different algorithms differently.* <u>Our experiments indicate that for low-complexity datasets, the optimal performance is already achieved with rather few instances, and using more data even decreases performance. For instance,</u> *in respect to linear regression the low complexity dataset "Skin NonSkin" would provide worse results with increased chunk size decreasing from a perfect RMSE and mean absolute error metrics score of 0.0 in all chunks upto 50,000 to 0.34306 and 0.22578 for chunk sizes 500,000 in RMSE and mean absolute error respectively. […]*

Commented [JB26]: Good opener (straight to the point, i.e. the answer to the question). Now, an explanation need to be coming that makes the reader agree with the answer…

Commented [JB27]: … and here it comes. An explanation that for low complexity datasets it seems not to be effective to use large amounts of data…

Commented [JB28]: … yet for high complexity datasets it is different. Very well.

Commented [JB29]: This is also good: specific numbers that help the reader to understand your reasoning.

Good and not-so-good answer for Question 2, Task 1

Q. 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]

Not so good answer:

> *Rule based algorithms are defined as the algorithms whose rules are already defined and they work solely on the basis of their rules. As there is no possibility to make these algorithms learn so this is a disadvantage due to which they are less efficient. Rule based algorithms would not be effective in understanding the relation between the variables of the "The SUM dataset (without noise)" dataset, as the features have varying properties .It contains features which are multiplied by 1.2 each time. So, the Simple Linear Regression and Logistic Regression model is better suited.*

Commented [JB30]: Not relevant (though this did not deduct marks as long as the remaining answer is relevant)

Commented [JB31]: Not correct. Actually, a rule-based algorithm would perform perfectly well on the SUM dataset without noise. I provided the perfect rule in the readme file that explained how the target value was calculated. If you know the rule, you can calculate the target perfectly

Good answer:

> *Of the two regression algorithms used, Ridge-Regression performed perfectly as measured by both metrics we used. An average RMSE of 0 and R2 score of 1 means that each prediction made was correct. Ridge-Regression allows for a stricter adherence to the data points, which means that something as linear as the SUM dataset is a perfect match for it. K-nearest-neighbours outperformed Logistic Regression by both F1 score and Accuracy but was not as efficient as successful as regression. A rule based algorithm would work perfectly (and much quicker) for this dataset as it was constructed by a simple mathematical rule.*

Commented [JB32]: Ideally, the specific numbers would have been given to show how good the algorithm was.

Commented [JB33]: Ideally, you would explicitly mentioned the rule, or mentioned that the rule was given already in the instructions.