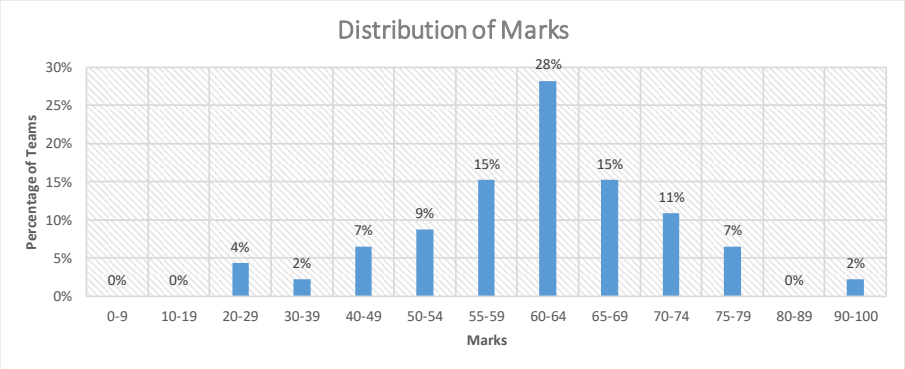
Feedback for all



This document contains feedback to research assignment 1. The feedback is quite high level, and I did usually not comment on minor mistakes (e.g. using an incorrect normalization method) as we had not covered these aspects at the time of writing assignment 1. However, in your second assignment, please ensure to do at least some of the following points:

- do some feature selection
- calculate confidence intervals or statistical significance between results,
- normalize / standardize data where appropriate
- Explain if and how you encoded variables (e.g. one-hot encoding)
- Do a few other things we covered in the lecture (e.g. clean the data).
- Do hyperparameter optimization, and state the hyper parameters. In the current assignment, the vast majority did not mention what the hyperparameter were they used. You should provide e.g. a table listing the parameters.
- Visualize the dataset better (e.g. distribution of attributes/values/classes)
- Think of appropriate (non-ML) baselines (something like “random” or “most popular”) [this may not be applicable to all tasks]
- Do outlier detection and find (and deal with) missing values
- Provide references /URLs to the datasets you used
- Use cross validation instead of using e.g. a 70/30 split; and hold out an extra test set, unless there is a good reason to not doing so.
- I have mentioned several times in the lecture that cutting off the y-axis is almost never a good practice as it provides a wrong perception of the true results. I did not deduct marks this time, but if I see cut-off axes in assignment 2, I will deduct marks.
- Make more use of the machine learning libraries features. It is really easy to visualize datasets, identify missing values, plot correlation heat maps, etc with scikit-learn etc. Use this.
- Many of you have not stated which ML library they used. For instance, you just wrote that you used Logistic Regression but not if you implemented it yourself (which I would not advise to do), or whether you used e.g. scikit-learn. Please change this in the next report.

Overall, many of you received rather few marks for the results & discussion section because many reports lacked critical thinking and demonstrating that you really understood what you did. To give

you one example: a few groups trained classification models that achieved accuracies of slightly above 50% for a binary classification task in a balanced dataset. In other words, the models performed not much better than a random guess (which would also be correct in 50% of the cases). If you have such results, and just report them without mentioning that these results are surprisingly poor (i.e. not better than a random baseline), you will lose marks.

I copied & pasted passages from the assignments and commented on them. I suggest that everyone reads every comment. I usually did not comment on a mistake in an assignment, if I commented on the same mistake already for another assignment.

I have not yet read and marked all assignments, but, frankly, I strongly assume that the comments for the outstanding assignments will not differ much from the current comments.

Task 104

Does the effectiveness of machine learning algorithms differ due to number of training data points?

Commented [JB1]: Think carefully about framing a question. The answer to that question is either "yes" or "no" (and it is pretty obvious that the answer will be "yes"). A much better question is one that allows a more differentiated answer. For instance "To what extent does… differ?" or "To what extent is the effectiveness of ML affected by the number of data points"?

1 Introduction

Machine learning can be defined as an application that provides systems the ability to automatically learn and improve without being explicitly programmed[1]. Machine Learning systems are created by Machine Learning algorithms.

Machine Learning algorithms take in a number of previously defined data points which include the inputs and the output and "train" the system to predict the output given a set of data points.

Commented [JB2]: Remove for second assignment. Focus on information that is directly relevant to the research question

The research question for this paper is how many data points is necessary to effectively train a machine learning algorithm.

Commented [JB3]: It's good that you have a question here, but it raises another big question: What does it mean to effectively train a ML algorithm?

The dataset chosen was a set of results for the video game Dota2. The result indicated whether a given team won (1) or lost (-1) a match. This meant that a simple binary classification could be used. A binary classifier is the

Commented [JB4]: Good description.

Each model that was trained was quadratic in nature in order to try and avoid problems with overfitting and underfitting. In order to add a quadratic feature a column was chosen from the data set that was then squared and then added back into the data set.

Commented [JB5]: ?

4 Results and Discussion

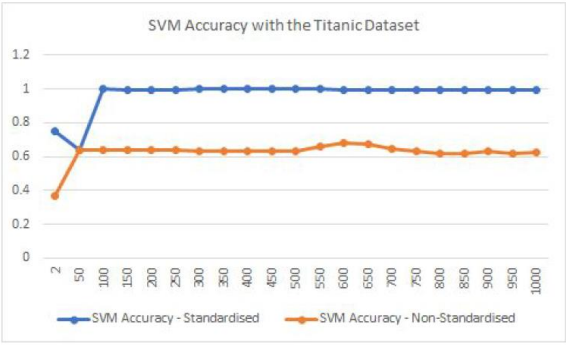
The computed F-Score was used a means of measuring the "effectiveness" of the overall training achieved. The gathered F-Scores were then plotted against the data point amount in order to see what effect an increased amount of data points has on the results.

As shown in the above graph, the outputted F-Score varies wildly while the amount of inputted data points is low. This variance decreases as the amount of data points increase. While there do exist F-Scores from low data point amount that equal higher data point amounts, these are random in nature and not repeatable on further runs and tests. The higher data point amount consistently output the same result over several iterations.

[1] "
[2] "
[3] "

These results revealed a new factor in defining how effective a training set is, that being the replicability of a given data point amount in consistently achieving high F-Score values across multiple runs.

Given this data the meaning of "effectiveness" has shifted from solely tied to the output metric but also to how reliable the given metric is. These results show that as the amount of data points increase, the variance of the outputted results decreases.



Commented [JB6]: When you have a research question, it is your responsibility to think of what data you need to answer the question appropriately. For some questions, using a rather small dataset with e.g. 14k instances may be fine. However, if your task is finding out how the size of a dataset impacts ML performance, than you should use a (very) large dataset. Otherwise, how can you draw any conclusions? In the example here, all you can say is that the performance remains rather stable between 2k and 14k instances. But how about 100k? 1m?
Also, the statement "the F-Scores varies widely" is not ideal.

1. You cut off the y-axis. Hence, ignoring the one outlier for ~500 datapoints, the F-score varies between 0.68 and 0.64. That is not a lot.
2. You use a small dataset, and start with small batches with a few data points. Apparently, you did one analysis with a 60/40 split. With that little data, some variance is to expect. It would be better to use k-fold cross validation to get smoother results. Or, if you use a simple 60/40 split, to discuss and mention that the variance was to expect.

Commented [JB7]: If you have a research question or goal (which you had), then you must answer that question in your result section (which you didn't). If you have some additional findings beyond the original research goal, then that's great and can be mentioned in addition, but not instead of the answer to the research question. Also, the fact that variance decreases with increasing data points is the very nature of statistics.

Commented [JB8]: If your ML model achieves an accuracy of almost 100%, then the chance is high that something is wrong with your data. Data leakage could be one reason https://www.google.ie/search?q=machine+learning+data++leakage (of course, it can also be that everything is alright, and the data is just highly suitable for machine learning).

Either way, if you receive a near-perfect accuracy with just 100 training instances, then this needs discussion (at least mention that this is a surprisingly high accuracy, and if you have potential explanations… even better). For instance, you could reference other researchers who have receive

Fig 4.1.1 Room Occupancy Accuracy Levels
Blue: standardised data, orange: non standardised data.
Chart showing the sudden dips in accuracy decline with more data when using the perceptron algorithm.

The results of the room occupancy data [Fig 4.1.1] show that with increased data amount, the accuracy of the algorithm tends to smooth out. The lowest points of the graph, at data amount 3000 and 6500, show a 3.27% increase in accuracy with increased data amount.

Commented [JB9]: I cannot follow your conclusion (accuracy is smoothing out). To me the differences look like random fluctuation. You would have to calculate confidence intervals here to backup your claim.

The results of the Titanic dataset [Fig 4.3.2] on the other hand convey that increased data amount can achieve an increase in accuracy. This can be seen in the fact that with 2 data points the accuracy was only 75.4% but with 100 data points the accuracy was 99.68% and stayed relatively the same up to 1000 data points.

Commented [JB10]: Saying that increasing data leads to increased accuracy because accuracy for 2 instances was lower than for 100 instances, does not seem appropriate in the given context. First, using 2 instances is never enough in machine learning. Second, when talking about "more and more data" nowadays, then this refers to having millions or even billions of instances. If one can achieve 99.68% accuracy with 100 instances, then this implies that the amount of data is irrelevant.

3 RELATED WORK

In [3], it was proposed that training size should be defined by specifying confidence interval widths for classification algorithm in bio spectroscopy field. As mentioned in [4], increasing the training dataset will overfit the model. It was found in [5] that how the performance of models vary with the training dataset size in biomedical applications. The investigation in [6] describe about how much training data is

Commented [JB11]: Explain in more detail

4.1 Data Sets

Two multivariate datasets from the UCI machine learning library were used for the research, "Bike Sharing" and "Bank Marketing" dataset. Former dataset has 17,379 observations and 16 features recorded for two years at day-hour level. The target variable is the number of bikes rented and features are environmental conditions at the hour. The later dataset is a bank customer level data which has 45,212 observations and 17 features. Target variable is if the contacted customer subscribed to the bank term deposit or not & the features are client information.

Commented [JB12]: Good. Visualization would be better.

4.2 Data Pre-processing

In both the datasets, null values were treated, and EDA was performed to understand the datasets. Label encoding was applied on categorical variables. In "Bike Sharing" dataset, new features like Sunday flag (day is Sunday or not) & day period ("noon", "evening" etc.) were created.

Commented [JB13]: It's good that you "treated null values", but how did you do it?

Commented [JB14]: good

4.3 Train data splitting

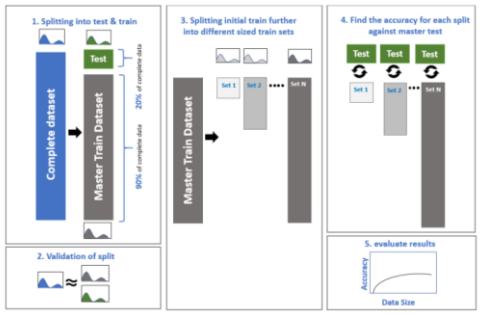


Figure 1: Data splitting approach

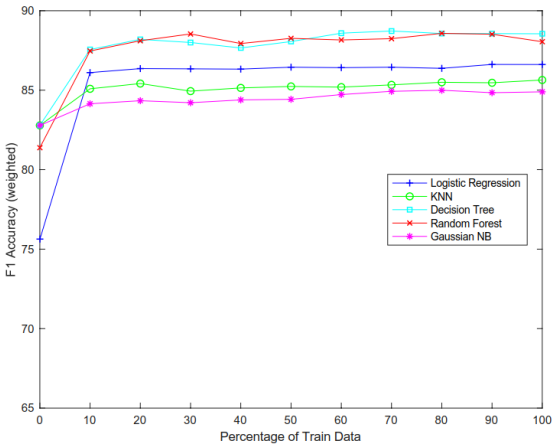
To analyze the impact of data amount on the performance of ML algorithms, the dataset was split into test and train parts in ratio of 20:80. (Fig. 1) Ten training datasets were further generated from the initial train set. The difference between size of consecutive training sub sets was constant, i.e 10% of initial train data size. Also, distributions of all test & train datasets were compared with that of the complete dataset to verify sample selection.

Commented [JB15]: there is not really a need to visualize the training/testing data splitting. If you say that e.g. 10% of the data was hold out for final testing, and the remaining 90% were used for k-fold cross validation (k=10), then everyone knows what you did

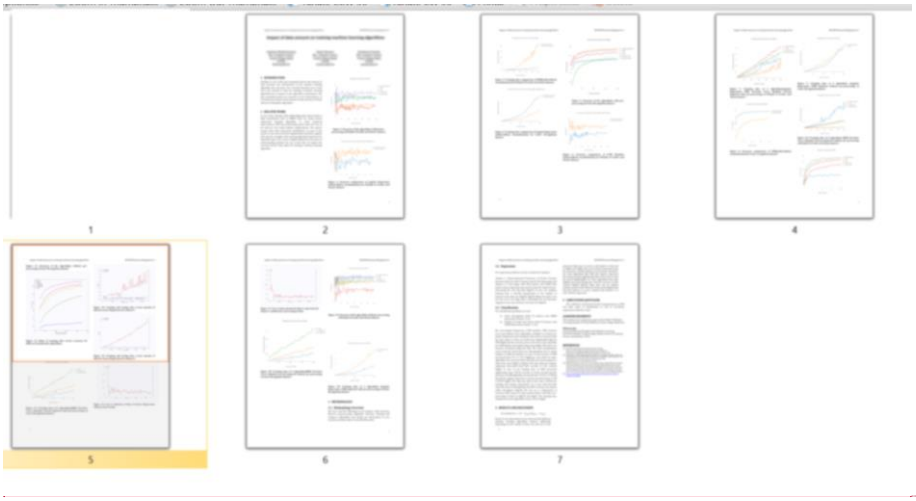
4.4 Model Building

As dependent variable in the "Bike sharing" dataset is continuous, linear regression, support vector regression and random forest algorithms were implemented. In "Bank Marketing" dataset, as dependent variable is binary true/false, logistic regression, K nearest neighbour (KNN), decision tree, random forest & Gaussian Naïve Bayes (GNB) algorithms were implemented. All the models were implemented on the individual training sub-sets generated from the initial train dataset.

Commented [JB16]: You used a lot of algorithms, and two datasets, which is very good. You might save some time (and still receive the same marks), if you just treat both datasets the same, i.e. as a classification problem by transforming the continuous data into categories. Anyway, the way you did it is also fine.



Commented [JB17]: IMHO it would make more sense to show the absolute number of training instances, not the relative number.



Commented [JB18]: I like illustrations, but these are far too many. Please find a way to summarize your results better.

3.2. Collecting datasets

For Logistic Regression, we chose a 'simple' dataset, and a 'complex' one ('complex' dataset has over 3X the features of a 'simple' one).

Commented [JB19]: Good, you explain what the difference between the two datasets is

Following is the description of the classification datasets (all multivariate, binary class):

1) Census Income Data (Simple dataset): Contains **14** features in total (**6** of which were used) and **33k** instances. This is a multivariate, binary class dataset
2) Dota2 Game Results (Complex dataset): Contains **116** features in total (**113** of which were used) and **92k** instances.

The Linear Regression Dataset (Beijing PM2.5 Data) contains **11** features (**8** of which were used) and **40k** instances.

Commented [JB20]: What is the target?

We developed scripts that measured performance as follows:

1. Randomly select **n** training samples from dataset having **N** instances.
2. Train the model using the **n** samples
3. Test the model against the remaining (**N-n**) samples by performing cross validation.
4. Calculate metrics
5. Increase **n** and repeat step #1.

Commented [JB21]: Not needed

1 INTRODUCTION

The amount of data required to train machine learning algorithms is an important question in the field. The optimal amount of training data gives maximum accuracy while minimising the amount of time needed for the algorithm to train.

However, it is not fully understood what that optimal amount of data is, and what effect the learning algorithm would have on that amount. Our research examines the effect of dataset size on various machine learning algorithms.

Commented [JB22]: The layout is not really according to the ACM template.

4 RESULTS AND DISCUSSION

4.1 Tensorflow results

Commented [JB23]: Structure your report based on what you want to achieve. It doesn't really matter what the results of Tensorflow and skikit learn are in this task. Choose the headings based on the algorithms or dataset size or so.

Task 103

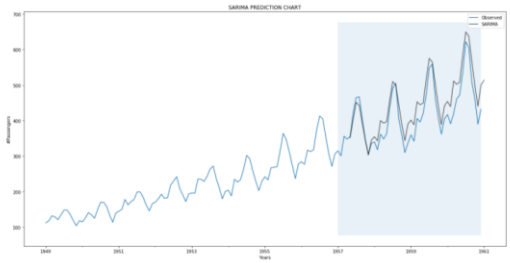


Figure 5: SARIMA future prediction after 1957.

In above *Figure 5* blue line is showing the actual observation and black line is showing the predicted value using SARIMA.

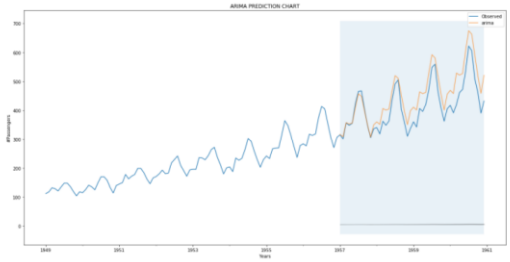


Figure 6: ARIMA future prediction after 1957.

Algorithm	MSE	MASE
SARIMA	0.0076	0.076
ARIMA	0.0113	0.0904

Table 1: MSE and MASE value for SARIMA and ARIMA (Monthly)

Algorithm	MSE	MASE
SARIMA	50.9616	7.1359
ARIMA	50.8746	7.1299

Table 2: MSE and MASE value for SARIMA and ARIMA (Quarterly)

Word Count - 1053

Commented [JB24]: Each figure for itself is rather meaningless. Merging them would provide a much better way to compare the two algorithms with each other. As a general guideline: If you have to figures both with the same x and y-axes, and both showing the same thing (e.g. for different algorithms), then it probably make sense to just have one figure and plot both algorithms in that one figure.

Commented [JB25]: Your goal was to compare how effective time series forecasting is based on granularity. Hence, it would be highly beneficial to have one table showing results for monthly and quarterly (and yearly) results instead of having two tables.

Looking at the numbers it seems that quarterly prediction errors are 6,500 (!) times higher than for monthly predictions. This needs a detailed discussion. This seems very weird to me. How did you calculate MSE? This needs explanation.

Commented [JB26]: Are you aware that there was a "hard word limit" ?

"Are single-metric results a true reflection of an algorithm's performance as it pertains to time?", this question posed a problem to our research, single metric values do not provide a thorough insight into the performance of a model, but merely give us an idea of a model's accuracy. Our goal in this research assignment, is to determine the most accurate model out of the nine we chose, and implement them on a suitable dataset.

Commented [JB27]: Too vague. I wouldn't understand what you really want to do. Also, if you say you want to determine the "most accurate" model, one will expect that you use accuracy as evaluation metric (which you didn't do).

Task 101

1 INTRODUCTION

Machine learning utilizes man-made reasoning to gain insights from data. With increasing amounts of data available, the benefit that machine learning creates for organisations has grown hugely.[1] Many machine learning tools are open source. This can both encourage innovation and allow for faster problem solving and troubleshooting of issues.[2] This is a comparative study of open source frameworks, focusing on three of the most popular tools in 2018; TensorFlow, Sci-Kit Learn and PyTorch.[3] The performance of these tools will be compared for three commonly used algorithms; Linear Regression, Logistic Regression and K-Nearest Neighbours Classification (KNN). The end goal is to make a recommendation for users.

Commented [JB28]: •You do not need to explain what machine learning is. Focus on the specific problem you are tackling
•There is no problem stated.
•There is no research question stated; and the goal ("comparative study") is too vaguely described.
•What exactly do you consider as "performance"? Time? Accuracy?

Overall, the introduction (and the following sections) did not really relate to the original task that I gave, i.e. finding out if and how the implementations of the same algorithms differ. Generally, this is ok (I don't mind if you deviate from the original task a little bit, but if you do, then it is even more important that you describe the problem and goal precisely).

3.1 Dataset and Pre-Processing

The dataset "Weather in Szeged 2006-2016" is publicly available on Kaggle. It contains 96453 entries with the following information.

- Formatted Date
- Summary
- Precip Type
- Temperature (C)
- Apparent Temperature (C)
- Humidity
- Wind Speed (km/h)
- Wind Bearing (degrees)
- Visibility (km)
- Loud Cover
- Pressure (millibars)
- Daily Summary

Three random samples of the dataset were taken of size – 15582 (small), 25014 (medium) and 55857 (large).

Commented [JB29]: Please justify your decisions in more detail. Why did you split your dataset into three samples? Or is this the training, validation and test dataset? If so, please use these terms.

Please clearly state what your label or target variable is. I could guess from the following charts, but you mention it explicitly.

4 RESULTS & DISCUSSION

4.1 Linear Regression

Implementation of a linear regression model produced similar results for prediction performance across all three frameworks (R^2 value ≈ 0.99). (Appendix B) There were differences in the training time of models, Sci-Kit took 0.07s, TensorFlow took 3s, whereas PyTorch took 11s when training the medium dataset. The training time of TensorFlow and Sci-Kit was only slightly affected by data size, whereas PyTorch was significantly affected.

Commented [JB30]: The key results, i.e. the performance metrics of the different frameworks, must not be presented in the appendix but directly in the result section. I am not marking the appendix.

4.4 Ease of Use

Sci-Kit provides an ease of use for Linear Regression, Logistic Regression and KNN with built in methods that didn't exist in the other frameworks. TensorFlow and PyTorch provide built in estimators/ functions, however they work with tensors. This required a large amount of data conversion between data

Commented [JB31]: Assessing the "ease of use" was not goal of the assignment. I suggest to focus on the given task only.

1 Introduction

Nowadays, many machine learning algorithms are implemented by various machine learning frameworks. The variation on the frameworks may lead to some changes in the performance even when implementing the same machine learning algorithm, hence we implemented linear regression model and logistic regression model via both the Scikit-learn library and the TensorFlow library, and to see if there is some performance difference between this two usually used frameworks by comparing the predicted accuracy, decision boundary, and mean square error between different implemented models.

Commented [JB32]: The second sentence in the introduction is too long. Same applies to some other sentences in the paper. Try to pay attention to ease the reading.

Overall, this introduction is better than the previous one, but would also benefit from a clearer research question. The problem could also be explained in more detail (but it's ok).

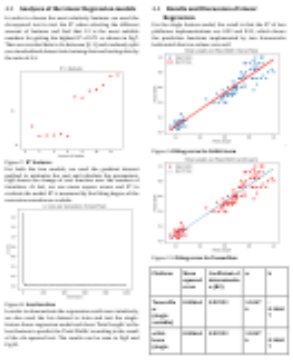
2 Related Works

Hands-On Machine Learning with Scikit-Learn and TensorFlow. It introduces two Python frameworks.

Scikit-learn and TensorFlow, which contains a range of techniques, starting with simple regression and progressing to deep neural networks.

TF.Learn: TensorFlow's High-level Module for Distributed Machine Learning. This paper briefly introduced the difference between TensorFlow and Scikit-learn..

Commented [JB33]: This related work section is not appropriate. The sentences are not coherent, and the section does not contain a single reference. Also, related work should be about other research that tried to achieve the same goal that you are trying to achieve. Please read my writing guidelines.



4 Performance Comparison

4.1 Analyses of Logistic Regression models.

We used the number 0, 1, 2 to represent the three iris species and stored this dataset to an array, and then used the standardization to perform the data scaling. For the purpose of getting the most representable features, we applied the chi-squared test to our standardized dataset and accordingly

1 INTRODUCTION

Machine learning (ML) applications and systems have blossomed in recent years as large volumes of data have become publicly available over the Internet. The value-adding analysis of such data has drawn the attention of an increasing number of academics and organizations, and has spurred the development of a wide array of open-source ML frameworks. These frameworks facilitate the construction and implementation of the machine learning pipeline that would normally be engineered by specialized teams. Performance of a given ML framework depends, therefore, on the design paradigms used by the developers in the implementation of the ML algorithm, and is subject to differ across frameworks. In turn, a ML algorithm implemented in two different frameworks may yield distinct results despite employing the same hyperparameters, or respond differently to changes in the values of the hyperparameters. Researchers, before initiating their study, are thereby presented with the task of inferring the ML framework that yields the best results despite having little information about the framework's performance compared to the others.

This paper aims to shed some light on helping researchers to rapidly evaluate the performance of cross-framework ML algorithms by presenting a comparative study of the consistency of the implementation of three machine learning algorithms, namely linear regression, logistic regression and neural networks in three different platforms: TensorFlow [8], Sci-Kit [6] and Microsoft Azure Machine Learning Studio (Azure ML) [2]. (We show that FRAMEWORK yields the best results for classification algorithms, while FRAMEWORK provides the best results for regression algorithms.)

Commented [JB34]: Overall, you had good visualizations. However, also the results should be visualized. If your goal is to analyse how similar three frameworks perform, then there needs to be one "main" chart for your main result, i.e. the performance of the three frameworks.

Commented [JB35]: Please use the headings that I suggested. Explaining how you encoded variables belongs to the methodology.

Commented [JB36]: This rather reads like an abstract that summarizes everything. Try to focus only on the background, research problem and goal. Also, using almost a third of your word count for a section that counts 10% of your marks, is probably not the best idea.

Balaji and Allen conducted a benchmark study of the results of open source automatic machine learning (AML) solutions [3]. Auto-sklearn, TPOT, auto_ml and H2Os AML solutions, all of which use Sci-kit, were tested against a compiled set of 30 regression and 57 classification datasets sourced from OpenML. Mean squared error (MSE) and weighted F1 were the set optimization metrics for regression and classification datasets test cases respectively. They show that auto-sklearn provides more accurate results for classification algorithms, having an average weighted F1 score of 0.75, while TPOT presents the best results for regression algorithms, having an inverted MSE of 0.9. Their experiment also supports the fact that different implementations of algorithms can yield different results even when the same toolkit (Sci-kit) is used.

Commented [JB37]: You clearly and precisely explain what the results from other researchers are. Very good. This is how a related work section should be.

Table 3: Comparison of Linear Regression Implementations

Platform	MAE	RMSE	CoD
Tensorflow	9.550613	14.460996	0.919245
Scikit-Learn	9.610823	14.323557	0.91636
Azure ML	9.095903	14.69058	0.913357

Commented [JB38]: Having a table like this is good (although it should be in the result section, not in the methodology section). Even better would be to visualize the data. Also, the many decimals are not necessary and just make it more difficult to read the numbers (or is it important whether MAE is 9.56 or 9.550613?)

Table 5: Comparison of Logistic Regression Implementations

Platform	TPR
Tensorflow	0.995013
Scikit-Learn	0.982801
Azure ML	0.991534

Commented [JB39]: Having only one metric is not ideal. Calculating additional metrics based on the confusion matrix should be a matter of minutes and would create additional value.

4 RESULTS & DISCUSSION

In summary, our results show that implementations of machine learning algorithms in different machine learning libraries are consistent to within 2% for all metrics used.

Commented [JB40]: Good. This is what I missed in many other assignments. If you explain a problem and state a research question in the introduction, then there needs to be an answer to the question in the result section. You clearly say whether or not the implementations differ and how you define this (e.g. a 2% margin). Well done, although it could be explained in a bit more detail.

3 METHODOLOGY

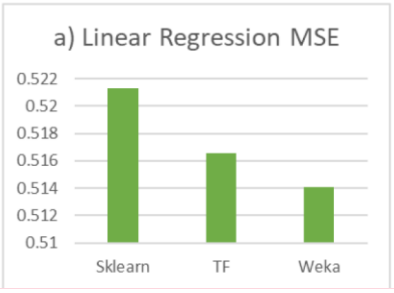
In this experiment, three Machine learning algorithms and frameworks were chosen.

Commented [JB41]: Ideally, you start a methodology section by explaining what you will do to answer the research question, and what your expectations are. For instance, write something like
"To answer the research question, we compare the performance of three algorithms from three ML libraries on two datasets. The three frameworks are…, the three algorithms are… and performance is measured as …. We expect that, if all implementations are similar, the performance will be similar. If we would observe high differences in the performance, this would be an indication for differences in the implementation". This would give the reader an overview what he or she can expect to read in the next sections.

3.1 Dataset

The dataset used was Google Play Store Apps: web-scraped data of 10k Play Store apps. This dataset contains all the details of the applications on Google Play. There are 13 features that describe an individual app. [3]

Commented [JB42]: Please visualize the dataset, and provide more detail



Commented [JB43]: Generally, this is what I want to see: a chart comparing the performance (e.g. RMSE) of the three frameworks. However, please don't cut off the axes. We discussed this plenty of times in the lecture.

2 RELATED WORK

Most of the studies on the comparison of different frameworks survey it from a Deep neural network perspective [Example [3] and [4]]. These studies take into account only Images as datasets [Example [5]]. Moreover, Only Speed and Time and not the Accuracy of the predictions is taken as a factor in the evaluation metrics of the said studies [Example [6]]. Scalability i.e. CPU and GPU compatibility and performance is a part of the evaluation metrics in comparison of different ML Libraries.

To the best of our Knowledge, no study on the performance comparison of different ML libraries has been done from the Data analytics viewpoint taking into account not just the time taken for the generation of predictions but also the accuracy of the predictions.

Commented [JB44]: Very good (I am not sure if the statement is true that no other studies have looked at accuracy but for the sake of the assignment I don't mind).

3 METHODOLOGY

Two different datasets are used, heights & weights dataset for linear regression and iris dataset for logistic regression. The height & weight dataset represents the linear relation of heights with respect to weights. The Iris dataset classifies

Commented [JB45]: These are two very simple datasets. It might be better to have one small and simple dataset (few features), and one larger more complex dataset (many features).

Also, as it seems you haven't used the second dataset? As you write later "We used only one type of dataset each for both of our algorithms." This is irritating.

. We used K-Fold to evaluate the

Commented [JB46]: What value is "k"? 5? 10? …?

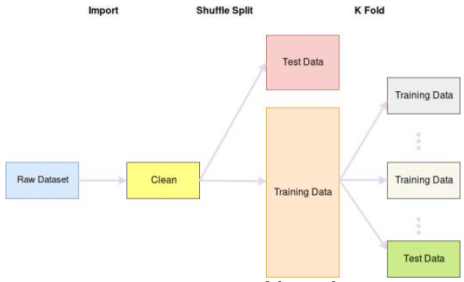


Figure 1: Diagram of the pipeline

Fig. 1 shows the workflow that we used to process the raw dataset. At the first stage of the pipeline, we clean the data that is, we removed incorrect data from the dataset. Then, we randomly split the data into 2 parts, training data and testing data. 10% of data was assigned to testing data and remaining data was kept as training data. We used K-fold cross-validation to separate the data into K subsets. Each of the k subsets includes training and testing data.

Commented [JB47]: It's nice to have such an illustration. The naming ("training data") is not ideal though because it's rather the "training & evaluation" data, right? Also, your sentences "At the first stage of the pipeline, we clean the data that is, we removed incorrect data from the dataset." Needs more explanation. What is incorrect data (how do you identify it)? How many instances did you remove, how many did remain?

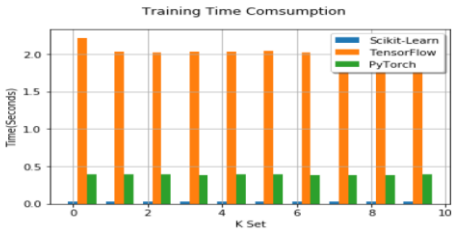
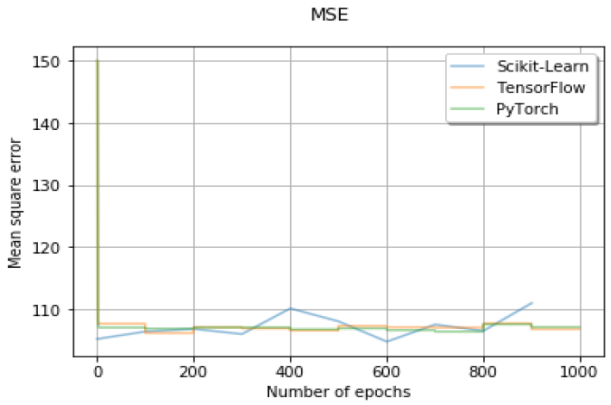


Fig. 2 Training Time Consumption for Linear Regression

Commented [JB48]: Good. If your goal is to compare the performance of three algorithms, then there need to be one illustration showing the performance of the three algorithms (exactly as you did).

Similarly, Fig. 4 demonstrates the mean square error as the number of epochs increases for Linear Regression. Since the

Commented [JB49]: I think your figure numbering got messed up. Figure 4 has the title "Accuracy…" and not Mean Square Error. Please ensure that all numbers are correct. I don't have the time to figure how what figure the description is referring to if numbers are not correct.



Commented [JB50]: Don't cut off axes please. The illustration looks weird to me. Needs more explanation. Did you use this for linear regression? You don't need 1000 epochs for linear regression and such a small dataset. It would be interesting to see the x-axis from 0 to… 20 to see more details on the strongly decreasing error.

Task 302

2 RELATED WORK

The concept of curriculum learning was discussed by Bengio et al. in 2009 [2]. Results from the study indicated that curriculum learning showed significant improvements in generalisation and the speed of convergence of the training process.

Collier and Beel discovered that the selection of the optimal syllabus is dependent on the task and that a proposed automated curriculum learning technique - *Predictive Gain*, performs competitively against hand-crafted syllabuses [3].

Commented [JB51]: Ok; One or two additional references would be better.

3 METHODOLOGY

3.1 Data Set

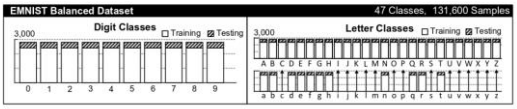


Figure 1: EMNIST balanced image classification data set[1]

3.1.1 Translation Data. For our second dataset we used a set of 150,000 English-French bilingual pairs, that vary in length and complexity, to create a multi-class classification model that can translate sentences. The data to be processed is cleaned initially of any punctuation or letter-case.

Commented [JB52]: And what is the first dataset? It's not explained in the manuscript (at least not before you explain the second dataset). Presenting just figure 1 (implying you used the EMNIST dataset), without explaining it, doesn't count.

Generally, try to explain the big picture first. For instance "We aim to explore the effectiveness of Curriculum Learning in the context of machine translation". The dataset we used for this task is xxx".

age and its mean.[1] For the language translation the data is organised by length and complexity of each sentence. Which creates

Task 0 will contain the *'easiest'* samples while Task N will contain the *'hardest'*.

Commented [JB53]: How do you calculate "complexity"? And how do you calculate e.g. "easiest"

4.1 Syllabus Setups

For our test, we used the following weightings:

Weightings	Stationary	Look Back	Look Forward	Balanced
Back Weighting	0	0.3	0	0.15
Current Weighting	1	0.7	0.7	0.7
Future Weighting	0	0	0.3	0.15

We chose to use a subdivision of ten tasks. We evaluated them after task was completed. For baseline comparison we evaluate after every 10k samples.

Commented [JB54]: Belongs to the "methodology", not "results"

4.2.1 Sorted Model. This model trained on a difficulty sorted dataset fared worse than our other baseline and other models. While no bug has been found so far, a possible explanation is that the model overfits to the simple examples early on, causing the model to become heavily biased towards simple models and failing to generalize to non-trivial examples until the backend of the dataset, causing a slow but present growth towards the end.

Commented [JB55]: You try to find explanations why a model doesn't perform as one may expect. Good!

4.3 Conclusions

A naive attempt at curriculum learning can cause biases to form in your model towards simpler data. When hand-crafting a difficulty metric, having a well distributed range of difficulties can help overcome these biases. We believe our final point of observation can stand to serve curriculum learning well. All our task based models performed better than our baseline initially. We believe that an accelerated curriculum[3] could prevent biases from forming on simple data and allow a model to quickly 'squeeze' out relevant information from its current task.

Commented [JB56]: I miss a little bit the "so what". Maybe I am misunderstanding the charts, but it looks to me that the "unsorted" baseline performs best. In other words, curriculum learning is not effective?

Task 203

Oguto Jo and Jacques W. have carried out a research to identify the order of importance of algorithms, based on the accuracy achieved as an outcome. However, there exists no research about the impact of individual algorithms over the ensemble models. We have tried to utilize the conclusions from the research by Jo and Jacques to identify the order of inclusion of models in our ensemble models.

Commented [JB57]: If you explain what others did, you must provide a reference.

A. Datasets used

Multiple open-source datasets (12, overall 24) for each category of problems were used. These datasets were taken from online platform "Kaggle". These datasets were pre-

Commented [JB58]: I couldn't find in the entire paper a list of the datasets. You need to name them.

VI. REFERENCES

[1] A comparison of random forests, boosting and support vector machines for genomic selection

[2] Comparison of 14 different families of classification algorithms on 115 binary datasets

Commented [JB59]: Not appropriate. Please provide author names, and other information that allows me finding the documents.

Task 202

2.Related work

There are many published research paper/articles that provide language identification with different number of languages. For example: the evaluation that provided by King and Dehdair(2008)[2] that achieved 99% accuracy with 300 types of language and 500 bytes input, Brown(2013)[3],"whatlang", that obtains 99.2% accuracy of classification with 1100 languages and a short 65 character test string. These works demonstrate that even with many languages, the data could still be accurate which indicate the method are quite reliable. However, the other work from Vatanen et al (2010) [4] that obtained only 62.8% with less than 10 character sample and 281 languages. Thus, it made us think the relation between the length of the sentence/word would affect the accuracy of the method/framework. There are more evaluations of the method that has been developed but the accuracy is more or less the same as their tested data are not as special as the three examples above.

Commented [JB60]: Very good related work, though the format of the document is not appropriate, and the English could need some editing.

In NLP applications, the size of training data influences the performance of the system, as we show by Halevy et al. (2009) [5]. To avoid this problem, we decided to use two different sizes of training data - 1MB and 2MB. If the size of training data affects the result, then all methods used with 2MB should outperform the same methods used with 1MB of data.

Commented [JB61]: Why do you measure data size in MB?

We present the results of three pre-trained naive Bayes (NB) LD implementations on a Wikipedia dataset and show that they perform poorly when faced with languages that have are strongly related to other languages.

Commented [JB62]: Not totally clear to me what the highlighted text means (what does "highly related" means?).

4 RESULTS

All three LD algorithms perform similarly well according to the recall and F1 metrics. Differences in the average recall rate between the algorithms can be largely attributed to differences in languages included in the algorithms. Once controlling for

Table 1. Performance of langid.py on Indo-European Languages

Language	Recall	F1	Language	Recall	F1
Indo-European Average	88.578%	0.51			
Afrikaans	94.0%	1.00	Kurdish	97.4%	1.00
Albanian	95.2%	1.00	Latin	85.4%	0.85
Aragonese	55.8%	0.89	Latvian	98.6%	1.00
Armenian	96.8%	1.00	Lithuanian	96.8%	1.00
Assamese	9.8%	0.98	Luxembourgish	92.0%	0.93
Belarusian	96.0%	1.00	Macedonian	98.4%	0.97
Bengali	91.4%	0.52	Marathi	93.8%	0.99
Bokmål	2.2%	0.92	Modern Greek	99.2%	0.99
Bosnian	8.6%	0.48	Norwegian Nynorsk	79.8%	0.99
Breton	97.0%	0.97	Occitan	71.8%	0.99
Bulgarian	94.4%	1.00	Oriya	96.0%	1.00
Catalan	91.8%	0.98	Panjabi	99.4%	1.00
Croatian	89.2%	0.48	Persian	99.8%	0.84
Czech	98.2%	0.99	Polish	99.2%	0.99
Danish	90.4%	0.99	Portuguese	95.2%	0.98
Dutch	97.0%	0.94	Pushto	76.2%	1.00
English	98.6%	0.40	Romanian	97.6%	1.00
Faroese	64.8%	0.98	Russian	98.6%	0.62
French	99.2%	0.67	Serbian	90.0%	1.00
Galician	92.6%	0.97	Sinhala	96.2%	1.00
German	97.6%	0.84	Slovak	96.6%	1.00
Gujarati	93.2%	1.00	Slovene	93.4%	0.99
Haitian Creole	99.0%	0.99	Spanish	98.0%	0.62
Hindi	97.2%	0.98	Swedish	98.2%	0.99
Icelandic	99.2%	0.74	Ukrainian	98.6%	0.99
Irish	98.2%	1.00	Urdu	89.8%	0.98
Italian	93.4%	0.95	Walloon	97.8%	1.00

This table shows the recall rate and F1 statistic of langid.py on the Wikipedia data for languages belonging to the Indo-European language family.Please see Appendix A for a qualitative discussion. *Source:* Authors' own calculations based on Wikipedia data and langid.py [Lui and Baldwin 2012; Thoma 2018].

Table 3. Performance of cld2.py

Language	Recall	F1	Language	Recall	F1
Afrikaans	0.99	0.93	Albanian	0.97	0.98
Arabic	0.99	0.88	Armenian	0.98	0.99
Azerbaijani	0.97	0.97	Basque	0.99	0.99
Belarusian	0.97	0.98	Bengali	0.88	0.71
Bosnian	0.45	0.51	Bulgarian	0.95	0.95
Burmese	1.00	1.00	Catalan	0.93	0.79
Central Khmer	0.88	0.94	Chinese	0.52	0.68
Croatian	0.61	0.55	Czech	0.96	0.97
Danish	0.91	0.91	Dhivehi	1.00	1.00
Dutch	0.94	0.90	English	0.99	0.31
Estonian	0.96	0.96	Finnish	0.99	0.97
French	0.98	0.69	Galician	0.96	0.90
Ganda	0.96	0.89	Georgian	0.98	0.98
German	0.97	0.69	Greek	0.99	0.91
Gujarati	0.96	0.98	Haitian	0.99	0.99
Hindi	0.98	0.96	Hungarian	0.97	0.99
Icelandic	1.00	0.70	Indonesian	0.89	0.86
Irish	0.97	0.97	Italian	0.93	0.54
Japanese	0.99	0.99	Kannada	0.98	0.99
Kazakh	0.99	0.92	Kinyarwanda	0.94	0.76
Kirghiz	0.99	0.87	Korean	0.99	0.99
Kurdish	0.98	0.98	Lao	0.87	0.93
Latvian	0.97	0.98	Lithuanian	0.97	0.98
Macedonian	0.93	0.96	Malagasy	1.00	0.99
Malay	0.86	0.89	Malayalam	0.99	0.99
Maltese	0.99	0.99	Marathi	0.95	0.93
Nepali	0.96	0.85	Oriya	0.96	0.98
Panjabi	0.98	0.99	Persian	1.00	0.71
Polish	0.99	0.98	Portuguese	0.92	0.92
Romanian	0.97	0.96	Russian	0.99	0.81
Scottish Gaelic	0.97	0.98	Serbian	0.94	0.87
Sinhala	0.94	0.97	Slovak	0.98	0.98
Slovenian	0.90	0.95	Spanish	0.94	0.74
Sundanese	0.92	0.95	Swahili	0.96	0.90
Swedish	0.91	0.95	Tagalog	0.98	0.95
Tajik	0.83	0.91	Tamil	0.99	0.99
Telugu	0.94	0.97	Thai	0.99	0.95
Turkish	0.98	0.87	Ukrainian	0.98	0.98
Urdu	0.89	0.81	Uzbek	0.99	0.97
Vietnamese	0.98	0.99	Welsh	0.99	0.99
Yiddish	0.98	0.99			

This table shows the recall rate and F1 statistic of cld2.py on the Wikipedia data. *Source:* Authors' own calculations based on Wikipedia data and cld2.py [Dick Sites 2013; Thoma 2018].

Table 4. Performance of langdetect

Language	Recall	F1	Language	Recall	F1
Average	95.0 %	0.95			
Afrikaans	99.2%	0.99	Macedonian	98.2 %	0.99
Albanian	96.6%	0.98	Malayalam	98.8 %	0.99
Arabic	99.8%	0.95	Marathi	95.8 %	0.98
Bengali	90.6%	0.95	Modern Greek (1453-)	99.2 %	0.99
Bulgarian	96.4 %	0.98	Nepali (macrolanguage)	97.6 %	0.99
Catalan	93.8 %	0.98	Norwegian	0.00 %	0.00
Chinese	70.6 %	0.82	Panjabi	99.4 %	0.99
Croatian	98.4 %	0.99	Persian	99.6 %	0.99
Czech	98.6 %	0.99	Polish	99.8 %	0.99
Danish	95.2 %	0.97	Portuguese	94.6 %	0.97
Dutch	97.0 %	0.98	Romanian	98.4 %	0.99
English	99.8 %	0.70	Russian	98.4 %	0.98
Estonian	94.6 %	0.97	Slovak	98.0 %	0.99
Finnish	99.8 %	0.999	Slovenian	97.4 %	0.98
French	99.2 %	0.96	Somali	95.6 %	0.98
German	96.4 %	0.93	Spanish	97.6 %	0.97
Gujarati	93.0 %	0.96	Swahili (macrolanguage)	93.8 %	0.97
Hebrew	99.8 %	0.99	Swedish	99.2 %	0.99
Hindi	98.2 %	0.98	Tagalog	97.4 %	0.99
Hungarian	97.0 %	0.98	Tamil	99.0 %	0.99
Indonesian	96.2 %	0.98	Telugu	95.8 %	0.98
Italian	93.6 %	0.95	Thai	98.8 %	0.99
Japanese	98.0 %	0.99	Turkish	98.2 %	0.97
Kannada	99.8 %	0.99	Ukrainian	98.8 %	0.99
Korean	98.8 %	0.89	Urdu	88.2 %	0.94
Latvian	99.2 %	0.99	Vietnamese	98.2 %	0.99
Lithuanian	97.4 %	0.98	Welsh	99.4 %	0.99

This table shows the recall rate and F1 statistic of langdetect.py on the Wikipedia data. *Source:* Authors' own calculations based on Wikipedia data and langdetect.py [Dick Sites 2013; Thoma 2018].

Commented [JB63]: This is important for everyone: Please make it as easy as possible for me to understand your results. If you write that e.g. all algorithms performed similarly (and this is your main finding), then this must be somehow to be seen in a single table or illustration, and this table / illustration must be referenced in the text. In the particular example, when I read "All three LD algorithms perform similarly well." I must first identify that this conclusion is drawn based on tables 1,3,4 and then I must look at all the dozens of numbers in the tables to get to the same conclusion. It would be much easier if there was a single table showing the overall performance of all three algorithms.

Task 107

1 INTRODUCTION

It is estimated that 2.5 quintillion bytes of data [1] is produced each day in the current scenario. Companies have started to embrace this data driven era by investing in analytical solutions for improving their productivity. There is a paradigm shift from generalized marketing to customer targeted marketing. Social media platforms allow business to target users based on market segments where gender is an important demographic segment. Twitter is a rich source of public data and profile information, but it does not store gender details. Hence, it is a challenge to identify the gender of users with the available information namely tweets and description. Our aim is to identify the most important features that influence the gender of a twitter profile using the public data available on twitter.

Commented [JB64]: Not really necessary. Remove for second report

3.2 Feature Generation

As a part of feature generation, we processed the text data i.e. tweets and description into numerical features namely number of characters, number of emojis, number of hashtags etc. which resulted in a total of 23 new features. In addition, we created flag variables for features having more than 10 categories.

Commented [JB65]: It's good that you created new features. Would be even better if you listed them e.g. in a table or at least appendix.

We removed features which produced minimal variation and had maximum null values. Further, we formatted the collected and generated data into a consistent format for model building. We also created dummy variables for categorical features and then summarized the data to user level. Finally, we got 71

Commented [JB66]: It's good that you removed irrelevant features, but please explain this in more detail. What exactly does "minimal variation" and "maximum null values" mean? How many featured overall did you remove? How did you create dummy variables? With one-hot encoding (-1)?

We captured popular 3600 male and female names each from the USA social security national names dataset of the year 2000

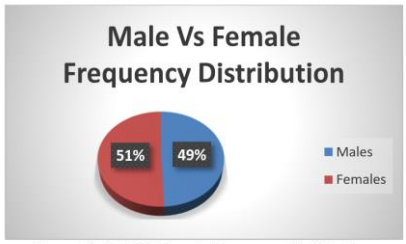


Figure 1: Male Vs Female Frequency Distribution

Commented [JB67]: If you captured 3,600 female and male names, why is the distribution not 50:50?
When presenting numbers, please ensure that they add up, or explain differences.

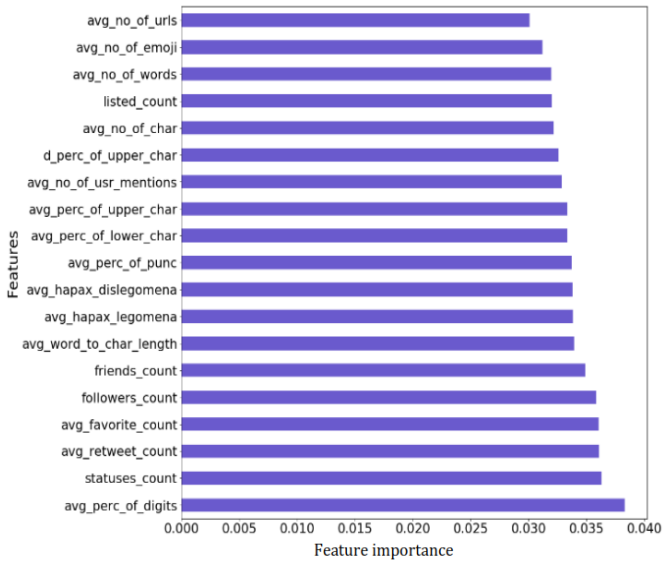


Figure 5: Top predictors in Random Forest.

Commented [JB68]: Illustrations must be self-explanatory. It remains unclear what "feature importance" is – correlation with the class label? Even having it explained in the text is not enough. It must be clear from the figure.

Algorithm/Metric	Kernel	Train/Test	Precision	Recall	F1-Score	Accuracy
Logistic Regression	-	Train	0.91	0.91	0.91	0.909
		Test	0.50	0.51	0.50	0.506
Support Vector Machines	Linear	Train	0.91	0.90	0.90	0.902
		Test	0.49	0.50	0.49	0.499
	Poly	Train	0.92	0.92	0.92	0.92
		Test	0.52	0.52	0.51	0.524
Naïve Bayes	-	Train	0.88	0.88	0.88	0.877
		Test	0.52	0.52	0.52	0.519

Table 1. Metrics involving text as a feature

For every classifier the analysis of text(tweet) as well as text and description were considered. First, we will discuss the results obtained for text (Table 1):

For the training set we see SVM performing better than Logistic regression and Bayes with 92% accuracy
The same is observed for the testing data as well, SVM classifier performs better than the other two taken into consideration

Commented [JB69]: Don't discuss the training error, it's not relevant.

Important for everyone: If your (test) accuracy is around 50%, and you have a balanced dataset, then this means your algorithm is not better than a random guess. Such a result implies that something is wrong in your setup, and this needs to be discussed. I will not mark you down for poor results, but you will lose marks if you don't see that something must be wrong, and you don't address it.

The dataset '**Twitter User Gender Classification**' from Kaggle was used. This data had been sorted by humans into being from a male, female or branded Twitter account. This data was presented as an Excel spreadsheet containing the text of a tweet, the Tweet author's perceived gender, a confidence interval for that gender assessment, the author's Twitter name, handle, profile description, and details about the Tweet and author such as date Tweet was posted, date the profile was created, number of favourited/liked Tweets, etc. This data was first trimmed to contain only the details of users who were male or female with an assessment confidence of 1. This yielded ~10,000 of the original 20,000-item data set. The data was then reduced to show only the text of the Tweet

Commented [JB70]: Good description of the dataset. An additional visualization would have been even better.

3 METHODOLOGY

We found a dataset [9] that included 14,000 Twitter users and their already-classified genders. We stripped the rest of the user data and only kept the username and the gender. Using the Twitter API, we collected the data of 9,500 of these users and stored it in a JSON. About 7,000 of these were male or female and the rest were brands or unknowns.

Commented [JB71]: You must report the distribution of your class labels (male; female). Writing that 7,000 instances are either class A or class B is meaningless. That is the same as saying "100% of the used data is either male or female".

Research Question: To explore what data properties on a twitter profile may be indicative of their gender, to allow public bodies to better understand their demographic.

Commented [JB72]: This is not a question.

Features	Male Precision	Male Recall	Female Precision	Female Recall	Accuracy %
created_at	0.54	0.41	0.54	0.67	54
favourites_count	0	0	0.51	1	51
colour	0.5	0.97	0.67	0.06	50
listed_count	0	0	0.51	1	51

Commented [JB73]: "Male Precision" is a rather uncommon metric. Precision is typically calculated over all classes. If you use an uncommon metric such as "Male Precision", you must define it and provide the formula in the text (I have an idea what you did; but the main evaluation metric that you used should not be subject to speculation).

5 LIMITATIONS AND OUTLOOK

The main limitation that we encountered, was that a major part of our data was not linearly separable (particularly the numeric data). A polynomial kernel was used to cater for this fact, but still the

Commented [JB74]: This is almost always the case, hence it is not a limitation that needs to be mentioned. If at all, it could be mentioned in the methodology as a justification for using a polynomial kernel.

3.2 Pre-Processing

The labeled data contains the gender values 'Male', 'Female', 'Brand', and 'Unknown'. Only 'Male' and 'Female' are considered, 65% of the data in total. To normalize numerical features, we implement vector normalization. Where possible, categorical data is converted into Boolean. Some non-uniform values are discarded, while others, like profile image, are set aside as potential future features. For other categorical features, we identify metrics to facilitate comparison. For profile descriptions, we count the number of words, number of hashtags, references to social media, and hyperlinks. Our finalized list is: account creation date, favorites, tweets, description length, hashtags in description, social media references, username length, default color scheme use, tweet length, mentions ('@' in a tweet), tweet hashtags, hyperlinks used.

Commented [JB75]: Good. Short and precise. Though, some more justifications would be beneficial

Table 1: Optimum XG Boost Feature Values

Parameters	Values
Booster Type	B tree
Evaluation Metric	logloss
Step Size (eta)	0.1
Tree Depth	6
Child Weight	10
Split Value	0.7

Commented [JB76]: Also good. Presenting the hyperparameters in a table.

In this study, we used various machine learning methods. Logistic Regression yields good results with an accuracy of 62.65% on training data and 62.22% on test data. Surprisingly, using the more complex SVM outputs slightly less accuracy, 60.72%, on test data and 61.69% on training data. The highest accuracy is

Commented [JB77]: Good. Explain that some of your results are "surprising", if they differ from your expectations. If you had presented an idea, why SVM may perform poorly, this would have been even better.

Column	Description
Gender	one of male, female, or brand (for non-human profiles)
gender:confidence	a float representing confidence in the provided gender
profile_yn:confidence	confidence in the existence/non-existence of the profile
Description	the user's profile description
fav_number	number of tweets the user has favorited/liked
link_color	the link color on the profile, as a hex value
retweet_count	number of times the user has retweeted (or possibly, been retweeted)
sidebar_color	color of the profile sidebar, as a hex value
tweet_count	number of tweets that the user has posted

Commented [JB78]: Presenting the features in a table is a good idea. Ideally you highlight which one is the target variable (even if it's obvious)

A. Data collection

A readily available csv dataset containing a list of tweets and related twitter profile information such as tweet-counts, favourite counts, user biography, etc. was taken from Kaggle [5]. The dataset also provides labelled data on the user gender; male, female or brand.

Commented [JB79]: This is not enough detail

Task 106

This paper considers 3 ML algorithms: Kernel SVM (sklearn), CNN and DCGAN (Tensorflow-Keras). These are applied to 2 datasets - MNIST for SVM and CNN[6], and Cifar10 for DCGAN[7]. MNIST contains 70K greyscale images of handwritten digits of size 28*28 pixels. Cifar10 contains 60K color images consisting of 10 classes of size 32*32 - we used a 6K car-class subset.

Commented [JB80]: Good. The most important information (algorithms, datasets, ML libraries) is given straight away (and then further details follow).

For Kernel SVM we chose to vary the shape of Kernels applied for classification problem. "rbf" and "poly" kernels take C and Gamma as parameters of the bell curve shape. "Linear" kernel takes one parameter-C. We conducted a grid search on cross-validation of the dataset to optimize the parameters of these kernels.

For CNN we conducted a random search for optimal values on the number of epochs and number of hidden layers, and timings for those. Secondly, we implemented different cost functions over the range of learning rates (Adam, Gradient Descent, RMSProp, Momentum and AdaGrad Optimizers).

GAN - we utilize hyperparameters recommended by DCGAN's author as initial starting-point. For ADAM optimizer we change learning rate, momentum term and Batch size in turn. We then explore the effectiveness of different optimizers, while keeping their relating intrinsic settings such as learning rate the same.

We plot values for multi-dimensional searches using contour plot with colorbar, as this demonstrates visually where the values peak. We used multiple linear plots for GAN to show how multiple values of a single parameter affect the training process over time(epochs), resulting in different convergence speeds.

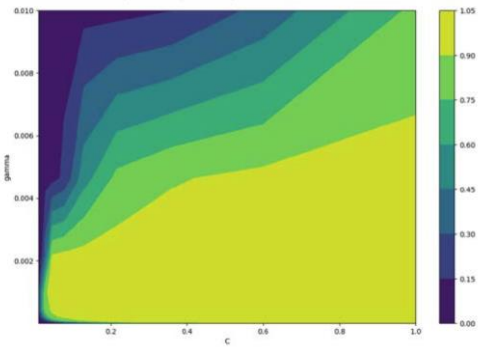


Fig.1 RBF Kernel shape – Accuracy value for parameters C and Gamma.

Commented [JB81]: This is good, but given that the focus of your work is to identify the impact of hyperparameters, it would be better to have one table that provides a concise overview of all hyperparameters.

Commented [JB82]: Why is the scale for accuracy going up to 1.05?

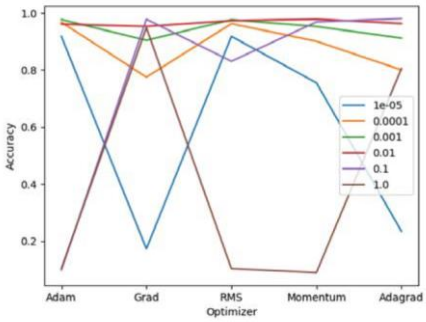
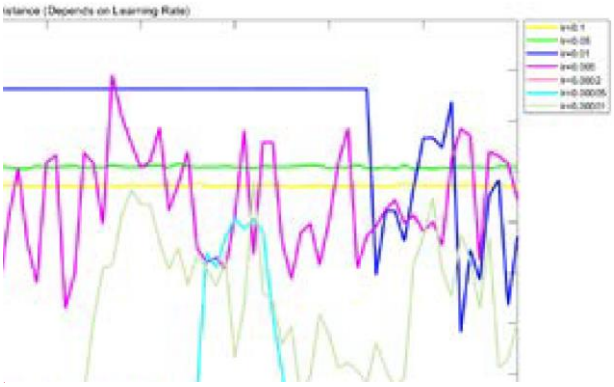


Fig. 5 Learning rate performance depending on the optimizer (x-axis).



Commented [JB83]: Bar chart would be more suitable

Commented [JB84]: I can't give marks for things I can't read.

Commented [JB85]: You obviously did a lot of work and many analyses. However, **for everyone**: At the end of your paper you must answer the research question and give a concrete summary. If your goal is to "demonstrate the effects of varying the hyperparameters and optimization strategies" then you should say something like that a fully optimized model was e.g. 5 times more effective than a default model. Or that parameter c had an impact of up to 140% on the performance.

1 INTRODUCTION

Machine learning algorithms have two types of parameters:

1.1 Model parameters. which are internal to the model and whose value can be trained from the given data.

1.2 Hyper parameters. a configuration which is external to the model and whose value cannot be trained from data. e.g. penalty parameter of error term in support vector machine, minimum sample split in decision trees.

These hyperparameters are generally specified by the user and are often tuned according to a given problem.

Hyperparameters hold importance because they directly control the behavior of training algorithm and have a significant impact on the performance of the model which is being trained. These parameters can be optimized using different methods to minimize the loss function or to increase the predictability of machine learning algorithm.

To achieve hyperparameter optimization, we need to consider

Our goal is to analyze the impact of hyperparameter optimization on machine learning algorithms and validate the effectiveness of optimizing hyperparameters over unoptimized ones.

2.1 Pre Context

The goal of many machine learning tasks can be summarized as training a model f which minimizes some predefined loss function $L(f(x^{(test)}); y)$ on given test data $x^{(test)}$ [1]. Common loss functions include mean squared error and error rate. One of the major challenges in machine learning is to select an appropriate level of model complexity. A complex model generalizes poorly to overfitting whereas a very simple model can result in under fitting in the data.

2.2 Sampling the parameter space

Apart from optimizing the hyperparameters manually, sampling the parameter space is achieved by three commonly used methods:

2.2.1 Grid Search. Grid search is an exhaustive search process through a manually specified subset of the hyperparameter space of a learning algorithm. A grid search algorithm needs to be guided by some performance metrics, which is generally measured by cross-validating with the training set.

2.2.2 Random searching. Exhaustive searching of all combinations in grid search is replaced by selecting them randomly in Random search. This can applied to the discrete set of hyperparameters, but also generalizes to continuous and mixed spaces.

2.2.3 Bayes optimization. Bayesian optimization builds an evolutionary probabilistic model which does a function mapping from hyperparameter values to the objective values on a validation set.

Commented [JB86]: Not necessary to explain. This is text-book knowledge.

Commented [JB87]: Good.

Commented [JB88]: Not necessary. This is text-book knowledge

Commented [JB89]: Again, text-book knowledge. Just say, which optimization strategies you used. The related work must relate to other research that wanted to find out what the effect of hyper parameter optimization is.

3 METHODOLOGY

A pre processed data is generated for binary classification with 10 features and each having independent gaussian distribution. This data has also been used in Hastie et al. 2009.

3.1 Estimators. Logistic regression, Support vector machine and Random forest classifier estimators are used to analyze hyperparameter optimization hypothesis.

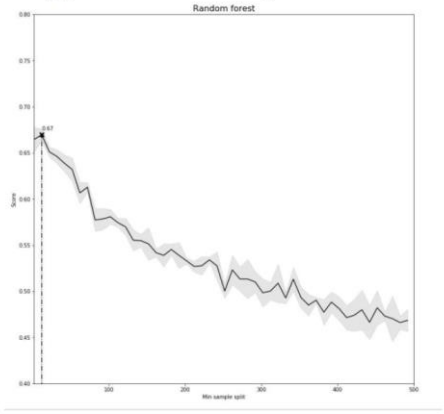
3.2 Parameter space. Parameter space is found by iterating over range of values and testing the estimator to find an optimal search space over database.

3.3 Sampling parameter space. Sampling of parameter space is done using Grid searching and random searching and is decided heuristically.

3.4 Cross validation. Grid based search algorithm is used for optimization of score over parameter space using K-fold cross validation scheme.

3.5 Scoring. Scoring is done by using mean accuracy of predictions.

4.1 Random Forest Classifier. Minimum number of samples required to split an internal node is optimized. With an increase in the hyper-parameter we are stopping the further split of a node to get a better fit on the data. As can be seen from the plot of score vs minimum sample split, accuracy is less in the beginning due to overfitting with the increase in the parameter value accuracy improves and reaches an optimal value after which it starts decreasing again because of under fitting.



Plot 1. Random forest score

4.2 Support Vector Machine (RBF). SVM-RBF consists of two main hyper-parameters namely penalty parameter C and γ.

Commented [JB90]: This is not enough detail on the dataset. What is the content? What is it about?

Commented [JB91]: The upper part shows your methodology, the lower part the results. Your results don't seem to reflect what you claim to have done (methodology).

4.2 Discussion

Bayesian learning produces the most accurate results, producing a mean squared error 6% lower than the next best algorithm, which was random search (see table 2). Grid search converged on its solution the quickest but was liable to miss

Commented [JB92]: Good. You quantify the difference between the different results

No pre-processing was carried out, excluding ensuring no missing values. Only numerical variables were explored, categorical variables were excluded. Plotting all paired combinations of

Commented [JB93]: Decisions like these need a justification – even if you say you didn't have time is better than not providing any justification.

2 Methodology

The *Student Performance* dataset, from the UCI Machine Learning Repository[1], was used in this study. It contained the academic performance of students attending two Portuguese secondary schools, along with seconardary background information on each student. This included grades, absences, health status, and study times, among others. The data was presented as a CSV file with string headers. The set contained 649 students' data. 226 students attended School 1 and the remaining 423 attended school 2.

Commented [JB94]: You need to say what the target variable is. Don't let me guess (academic performance? Or which school they attended?). And what does "academic performance" mean? "Passed" or "fail" (a classification problem) or the marks (e.g. 73, which would be a regression problem). Also, are the "grades" the same as "academic performance" or is it two different things?

3.1 Algorithms

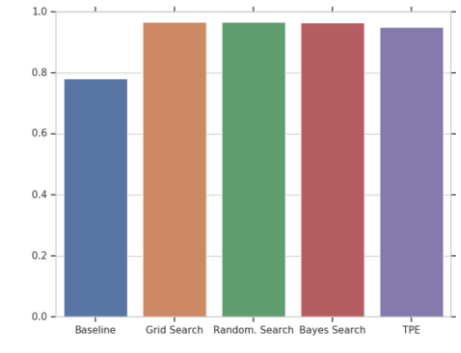
We chose several machine learning algorithms in order to ascertain whether the effects of hyper parameter tuning are exhibited universally by ML algorithms. This work focuses on the following classification algorithms: Support Vector Machine (SVM), Random Forest (RM) and Multi-Layer Perceptron Classifiers (MLPC), all of which have an extensive set of hyper-parameters to choose from [1]. We selected the hyper-parameter space for tuning based on the default values provided in scikit-learn and the related literature.[2] [1] Due to our limitations in processing power we limited the size of these search-spaces.

Commented [JB95]: Good explanation and justification

3.2 Datasets

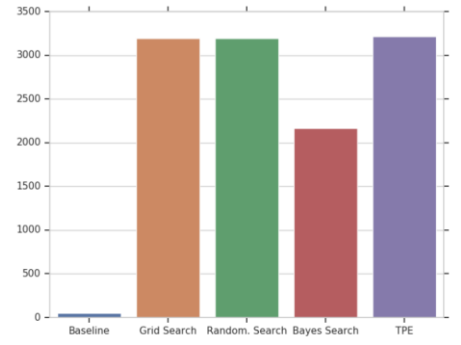
Based on the assumption that HPT can be applied in any context, we selected scikit-learn's digits dataset[3] and publicly available census data [4]; both requiring minimal preprocessing. The census data was cleaned by removing all rows with missing features and converting

Commented [JB96]: Especially for hyper parameter tuning it would be interesting to see the effect on larger datasets. If you still have the time for assignment 2, use at least one small and one large dataset.



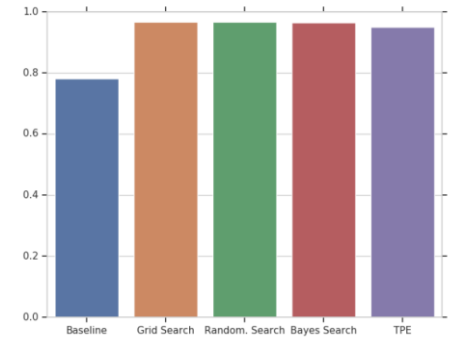
Commented [JB97]: Please add data labels so i can see the exact numbers. See my writing guidelines.

Figure 1: SVM on Census data - Tuning time



Commented [JB98]: What is the metric? Seconds? Minutes? …?

What is the baseline? In the text you mention that Grid Search is the baseline but here Grid Search is listed additionally



Commented [JB99]: It is quite surprising that all strategies perform almost alike and almost perfect. This needs more discussion (and ideally error checking your code and data).

Task 105

Performance of Naive Bayes and Random Forest on Noisy Dataset

ABSTRACT

The performance of a machine learning algorithm depends upon data quality and inductive bias of algorithm[4]. In this project, we applied data cleaning and feature engineering using variable importance to deal with the noisy dataset of IDA 2016 challenge and compared their impact on Naive Bayes and Random Forest. The proposed framework performs better with naive Bayes, in terms of minimizing the cost of repair, when compare with Random Forest, by the difference of $3,280.

This research paper measures the performance of Naive Bayes and Random Forest on the noisy data-set to minimize the cost of repair for Scania Heavy trucks.

3.1 Dataset

The dataset contains the information of working of several components. It is categorized into 2 classes on the basis of Air Pressure System(APS). Therefore, it is a binary classification problem. The first class contains all those component whose failure depends on APS and second class contains the components that are not related to the APS system.

3.1 Dataset

The dataset contains the information of working of several components. It is categorized into 2 classes on the basis of Air Pressure System(APS). Therefore, it is a binary classification problem. The first class contains all those component whose failure depends on APS and second class contains the components that are not related to the APS system.

3.2 Data Import

The dataset contains the information of working of several components. It is categorized into 2 classes on the basis of Air Pressure System(APS). Therefore, it is a binary classification problem. The first class contains all those component whose failure depends on APS and second class contains the components that are not related to the APS system.

Commented [JB100]: You did not really conduct the task that I gave. My question was "how strong exactly is the impact of noise on machine learning performance?". You answer the question "How well do Naïve Bayes and RF perform on noisy data?" that is a significant difference.

Anyway, it's ok and will not affect your marks. You can also continue with this task for your second assignment.

Commented [JB101]: Based on this description, I do not have a clear picture how the dataset looks like.

Commented [JB102]: Find the error… :-)

the and percentage of missing values each feature. The dataset contains features with more than 75% of missing values, which are noise. Furthermore, we analyze the data to find outliers using

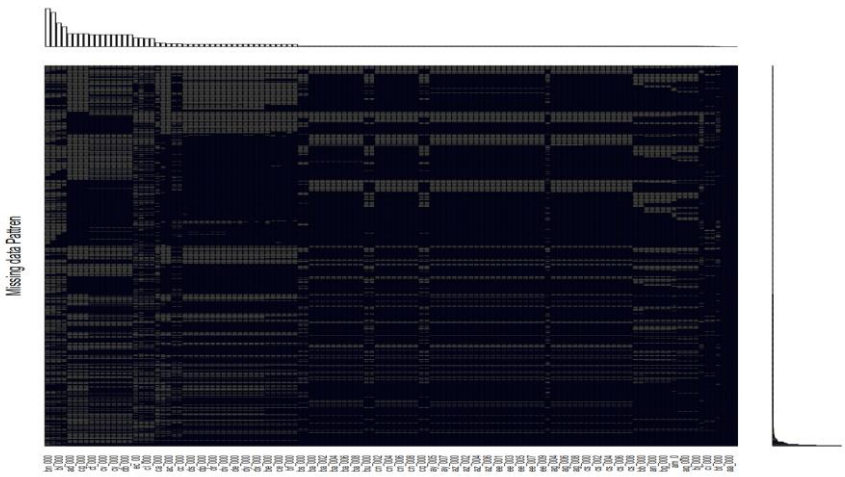


Figure 1: The graphs shows pattern of missing values w.r.t. features.

MACHINE LEARNING ALGORITHM	DATASET	DESCRIPTION	Implementation
Linear Regression	stockprices.csv	Stock prices of Google and Amazon for different dates. The goal is to predict Google's stock price given Amazon's stock price	own implementation
Linear Regression	boston.csv	Contains the prices for different houses in the Boston area with different features related to the house. There are 13 variables in the feature set based on which the price of the house is to be predicted.	sklearn
Logistic Regression	digit_image_class	Consists of the grayscale values for different digit images mapped its respective digit class	sklearn
Logistic Regression	wisconsin.csv	Used to predict whether a cancer is benign or malignant. It has 31 variables based on which the prediction is made.	own implementation
k-means	data.txt	2-D Data-points, which are clustered into 3 groups	own implementation
k-means	image.txt	3-D Data points, which are the RGB values of an image source. The goal is to divide the image pixels into clusters	own implementation

Figure 1: Table 1 (Algorithms used, base datasets and implementation type)

Commented [JB103]: Missing data is not (necessarily) noise.
Noise = Errors = False/inaccurate information

Commented [JB104]: Such a chart is not providing a lot of value

Commented [JB105]: I am not seeing the benefit of using different implementations of the same algorithms on different datasets. What is the reasoning? How does it help to answer the research question?

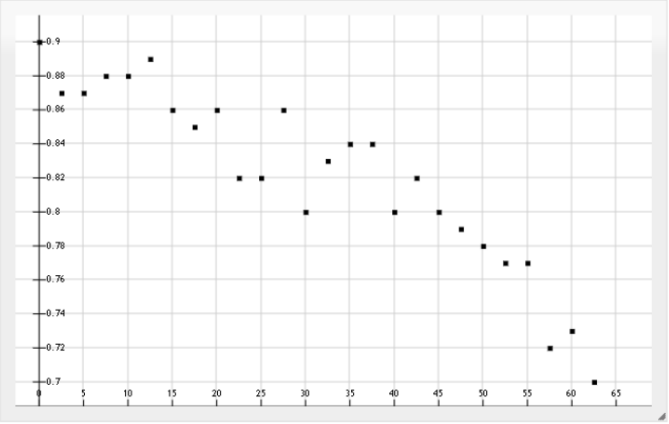


Figure 4: Result for digit_image_class dataset

Commented [JB106]: You must provide labels for the x and y axis. If you don't do this, I would have to guess but I cannot give marks for "Hmm… *probably* these results show that …". Also, it is not sufficient to explain the axes in the text. An illustration must be self explaining.

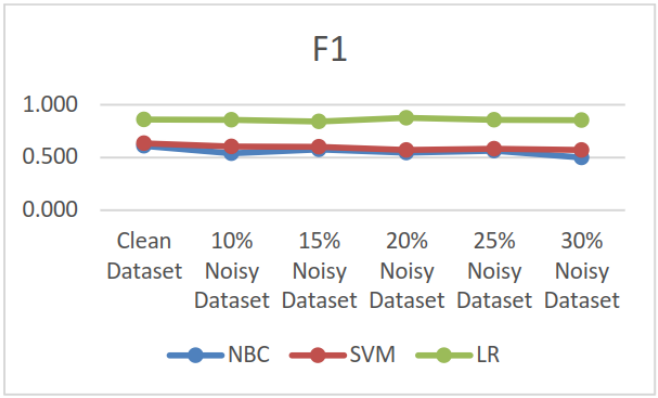


Figure 1. Variation of F1 score of the classifiers for increasing levels of noise

Commented [JB107]: Good, this is an illustration that I would expect for the given task. An additional data table would be good directly below the chart. However, the results seem awkward. Noise seems to have very little impact and LR performs far better than SVM? Such results need a detailed discussion and most likely something is wrong in your experiment. Also, add more than 30% noise for assignment 2. Based on your report, I assume you didn't do any feature selection and just used all features. Probably many of them are irrelevant, i.e. noise. This means, adding just a little bit of additional noise has no impact. Next time, do feature selection and other optimization techniques (missing values, …hyper parameter optimization).

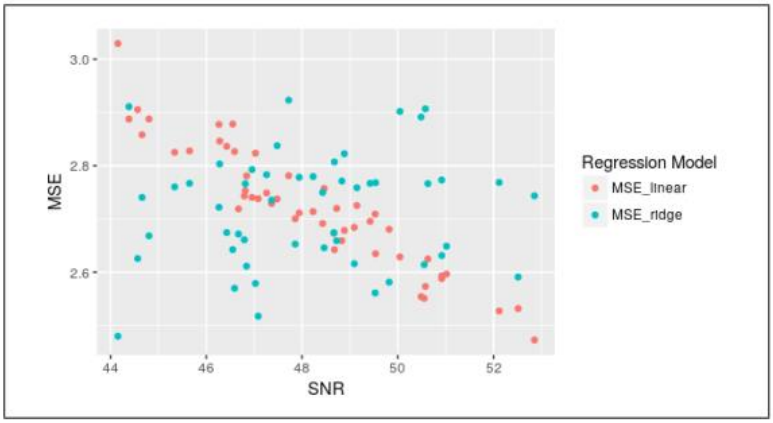


Figure 4: The Mean Square Error plotted as a function of the Signal to Noise Ratio in the observed model, for $n = 10,000$

Commented [JB108]: Don't cut off the axes. I consider this highly manipulative. It provides a wrong perception. If at all, then provide both the "normally" scaled image, and the excerpt.

4 RESULTS

As a sanity check, we examined the basic linear model with only the 6 currencies as predictors. The Dollar Index in the testing dataset was predicted with reasonable low error (Figure 2). The residuals were approximately Gaussian (Figure 3). Given this we were happy to proceed.

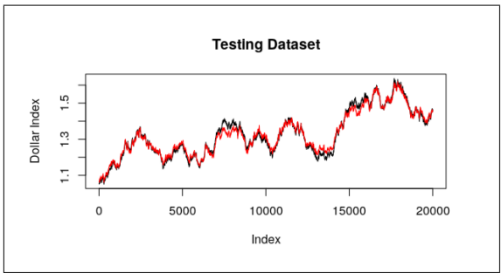


Figure 2: The testing dataset (black) and the predicted values from the gradient descent model (red)

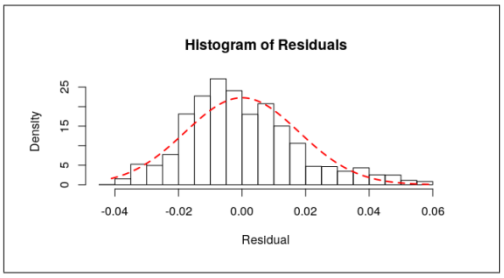


Figure 3: A histogrtam of residuals from prediction on the testing set. The corresponding Gaussian cirve is shown in red.

For the smaller dataset ($n = 10,000$), we observed the expected trend for the linear model: as the SNR increased, the MSE decreased (Figure 4). We observed a strong negative correlation between the SNR and MSE, $\rho = -0.942$. For the ridge model, however, there was no obvious trend between the SNR and the MSE $\rho = 0.109$. This was expected, as penalized regression models should identify predictors that are less relevant to the model, and so the noise variables should have less of an impact on prediction [10].

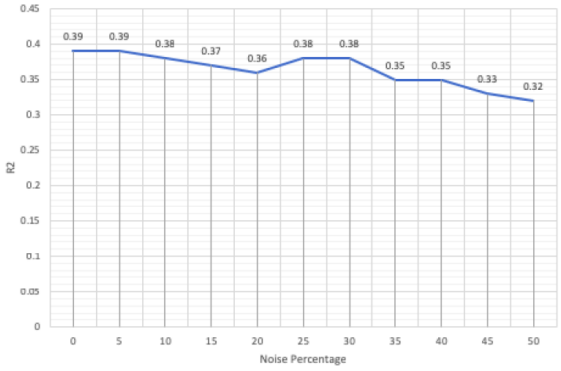
The larger dataset ($n = 100,000$) exhibited broadly similar trends to the smaller dataset, but the results were skewed by an outlier (Figure 5). This outlier corresponds to using zero noise predictors. Counterintuitively, the SNR seems to increase when one noise predictor was added. This phenomenon persisted in other datasets with the same number of observations (generated in the same manner), but not in their counterparts with $n = 10,000$.

As before, there is a negative correlation between the SNR and the MSE for the linear model, albeit not as strong $\rho = -0.628$. When

Commented [JB109]: **To all:** It is really important that you provide the "so what", i.e. you answer the research question or say if and how you achieved your research goal. If you look at this example, you see that the students did some decent analyses. But what can we learn from it, or more precisely, what do *you* see in the results? How strong is the impact of noise on machine learning performance? It is not said explicitly here, in other words the question is not answer and I cannot give full marks.

4.1 Regression Algorithms

With increase in tuple noise by 50%, linear regression experiences 7% decrease in R2 and 69% increase in RMSE. However, up to 30% of noise, the performance degradation is not severe in terms of R2, though RMSE increases steadily (Figure 4, Figure 5). The above interpretation is evident from the fit, shown in Figure 6 and Figure 7.



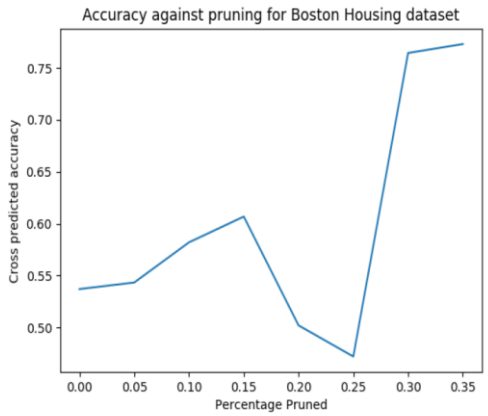
Commented [JB110]: Very good. Both, the illustration and the quantification ("increasing noise by 50% leads to …"). This is actually "actionable information", i.e. information that provides real benefit

Task 102

The question that we set out to answer was how dataset pruning affects machine learning performance. We tried to answer this question in two different ways. The first was to see the effects of dataset pruning on a given algorithm and see if pruning the data in a certain way would reduce, increase or keep the error and prediction rate the same. Secondly to compare the effects of pruning on different algorithms. If it was more effective on certain algorithms, we could ask why this might be.

decide which examples are troublesome. Their resu
follows.

Noise	Full	Pruned
0%	0.0918	0.0904
10%	0.1150	0.1118
20%	0.1604	0.1232
30%	0.3454	0.1853



Initially there is an improvement in the accuracy as some of the rows with missing data are removed, there is then a large drop in performance as possible the small number of missing value rows are weighted highly. Then once all missing value data is removed there is a large spike in accuracy. In this case pruning can improve the accuracy of a model.

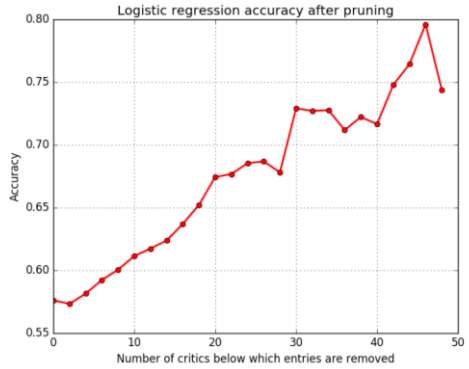
Commented [JB111]: in this task you need to be careful about language. Many of you wrote that dataset pruning would improve accuracy (or other metrics). This is not correct. The accuracy only appears better. "Pruning" means that researchers remove e.g. all movies from their dataset with 20 or less ratings. This means, they remove e.g. 30% of all their data. The resulting accuracy is higher as if the researchers had done the training and testing on all data. However, this is not improving accuracy in the real world, unless you want a system that can only e.g. give recommendations for movies with 20+ ratings.

Also, very few of you reported how much data remained after pruning the dataset. If you do something (be it feature removal, removing instances, filling with missing data ….) you need to report the "consequence", i.e. how much data (features / instances / …) were there before and after?

Commented [JB112]: Good: You describe what you want to do.
Not so good: You don't explain why you want to do it, i.e. there is no problem being described (e.g. that many researchers use pruned dataset and report good results on the algorithms, but it remains unclear how the algorithms would perform un unpruned datasets). You also should explain what dataset pruning is.

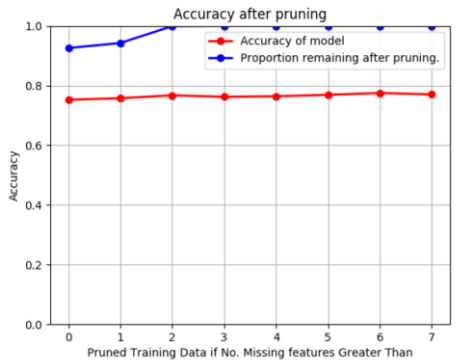
Commented [JB113]: What is the metric being shown here?

Commented [JB114]: It is good that you describe the chart but you should explain/assume what the line is somewhat weird (one would expect a straight line). Why might this be?



In the case of Logistic Regression, the results showed a similar increase in accuracy as data was pruned. This also matches with the results from Mudrakarta *et al*. The following graph illustrates our findings.

In the case of Neural Networks training data was pruned based on the number of incomplete features in the training samples. There was not a clear increase in accuracy as incomplete training data was pruned, the accuracy hovered just below 80%. It may be that pruning sparse categorical datasets does not improve accuracy, that neural nets do not benefit as much from pruning, that the limited range of sparseness of this data set made any benefit to pruning difficult to measure or a combination of these reasons.



5 Limitations and Outlook

One of our main limitations for this project was our methodology for how we pruned the data. Our methods for pruning the data were quite naïve and based on gut-instinct rather than any statistical or scientific method. If we had more time and expertise,

Commented [JB115]: Good. That's a chart that I would have expected.

Commented [JB116]: Good. Compare your results with the related work. Do your results meet your expectations? Are they surprising? Different? …?

Commented [JB117]: This is not what I would consider as dataset pruning. However, I am aware that different people use the term "pruning" differently. Hence, it is so important to clearly define it in the beginning.

Commented [JB118]: No, overall your methodology was good. If you look at research papers, you will that many researchers apply simple pruning methods such as "we removed all movies with less than 20 ratings".

datasets, which involves preprocessing datasets to remove suboptimal data. An important aspect of such work is the pruning of datasets, which involves preprocessing datasets to remove suboptimal data. This aims to reduce the effect of 'bad' training examples, such as noise-affected or mislabeled data, which can otherwise cause over-fitting to outliers. This study examines the effects of dataset pruning on machine learning

Commented [JB119]: This is not what I consider dataset pruning (see comment at the top of the section – to me, dataset pruning is e.g. removing all movies from a dataset with less than 20 ratings). Anyway, it is good that you explain what you consider as dataset pruning. It will not affect your marks negatively that you did something slightly different from what I expected.

model scores, if not carefully executed it can lead to bias and suboptimal model estimates [2]. Further research identified similar trends. The results of dataset pruning was gauged using eight benchmark tests, with five tests showing improvements, illustrating the often inconsistent nature of dataset pruning [3]. Anelia Angelova also notes "that the learning algorithm might be better off when some training

Commented [JB120]: How big were the improvements? Be more specific.

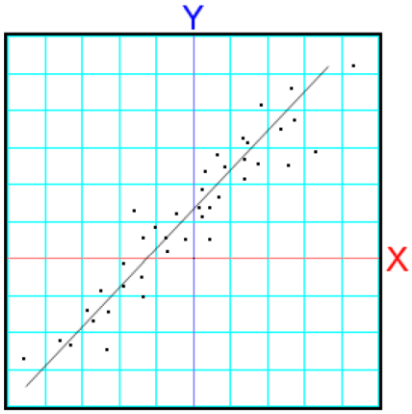
The effects of dataset pruning were examined using UCI's Auto-Mpg Data. This dataset details the following information pertaining to automobiles:

1. mpg: continuous
2. cylinders: multi-valued discrete
3. displacement: continuous
4. horsepower: continuous
5. weight: continuous
6. acceleration: continuous
7. model year: multi-valued discrete
8. origin: multi-valued discrete
9. car name: string (unique for each instance) [5]

This dataset was chosen as it as it allowed for the study of data pruning on various variable types, including integers and strings. The dataset allowed for research into the differences that occur after the dataset pruning of variables of the same type. The dataset also contained 398 instances, providing a large sample size, while also containing missing values, which provided us with the opportunity to examine the effects of dataset pruning on imperfect datasets. Three pruning

Commented [JB121]: 398 instances is not a large dataset. Especially for the given task the dataset seems not ideal given its small size. If you still have the time, it would be beneficial to have one additional large dataset for the second assignment.

0.01. This figure was chosen as opposed to the standard erro
of 0.05 due to the relatively consistent dataset.



Despite attempts to prune the dataset, we were unable to alte

Commented [JB122]: What is this? There is no figure caption, no labels of the axes, and no description of the illustration in the text.

4 RESULTS & DISCUSSION

Multiple attempts at data pruning were made, but such operations failed to have any significant impact on the results of both the OLSR and logistic regression test. The lack of outliers within the dataset made pruning with the Grubbs test difficult, as an extremely low value of error was needed to identify outliers.

Commented [JB123]: I am really sorry, but I cannot give any marks for such a "Results" section. Based on your methodology, you used multiple algorithms and multiple pruning strategies on one dataset. Hence, you should be able to provide several tables or illustrations and specific results. Even if there was no "significant" impact, you need to show that and provide specific results. What was e.g. the accuracy of the different algorithms? And if you say the impact was not "significant", you need to provide the significance level.

5 LIMITATIONS & OUTLOOK

1 INTRODUCTION

With the ever-increasing demand for electricity and for meeting environmental norms set-up by governmental agencies, the study of Energy Consumption and Pattern Usage across various appliances in domestic households take high priority. Moreover, the findings can address various issues like optimal consumption, leakage of electricity, predictive power management during the times of heavy power demands and load profiling. As per studies in Ireland, the average household consumes over 4200 kWhs amounting to excess of 1000 Euros [1]. The domestic electricity usage in the European countries stands at second place, only preceded by the industrial sector [2].

Commented [JB124]: This is not relevant for the given task. You main goal is to identify the impact of dataset pruning on machine learning performance. Using a dataset about electricity is just one way to get to your goal. Focus in your introduction on the main task.

2 RELATED WORK

Luis M. Candanedo, Véronique Feldheim and Dominique Deramaix have taken several combinations of attributes under consideration and they have discovered that GBM was most successful while predicting the usages. They identified that trivial attribute Pressure has more importance over other attributes as it directly influences the weather conditions. They

As part of the data cleaning process, missing values were imputed with the mean of the column. Min Max Scaling was used to standardize the independent variables. The main reason for implementing this method of feature scaling was to better handle and suppress the effect of the outliers. Date entries of string type were converted to ordinal format.

Table 2: Attribute Description

S.No.	Attribute	Description	Unit
1	Appliance	Total Energy Use	Wh
2	Lights	Energy used by light outlets	Wh
3	T1	Kitchen Temperature	Celsius
4	T2	Temperature – Living room	Celsius
5	T3	Laundry Area Temperature	Celsius
6	T4	Office Temperature	Celsius
7	T5	Bathroom Temperature	Celsius
8	T6	Outside (North) Temperature	Celsius
9	T7	Ironing Area Temperature	Celsius
10	T8	Teenager's Room Temperature	Celsius
11	T9	Adult Room Temperature	Celsius
12	T0	Outside (Weather Station) Temperature	Celsius
13	RH_1	Kitchen Humidity	%
14	RH_2	Living Area Humidity	%
15	RH_3	Laundry Area Humidity	%
16	RH_4	Office Humidity	%
17	RH_5	Bathroom Humidity	%
18	RH_6	Outside (North) Humidity	%
19	RH_7	Ironing Area Humidity	%
20	RH_8	Teenager's Room Humidity	%
21	RH_9	Adult Room Humidity	%
22	RH_out	Outside (Weather Station) Humidity	%
23	Wind Speed	Weather Station Data	m/s
24	Visibility	Weather Station Data	Km
25	Tdewpoint	Weather Station Data	Â°C
26	Rv1	Random Variable 1	NA
27	Rv2	Random Variable 2	NA
28	Date Time	Date Time	Yy/mm/dd hh:mm:ss

Commented [JB125]: Same problem as above: Please focus on work that relates to the main goal, i.e. dataset pruning. It is not relevant what other papers exist about predicting electricity consumption.

Commented [JB126]: Good (explanation how you pre-processed the data)

Commented [JB127]: good (table listing the features)

Table 3: Regression Model Metrics

Prediction Model	RMSE	R^2	MAE
Random Forest	76.5178	0.4649	35.5893
GBM	82.39261	0.3796	42.3337
SVM (radial kernel)	108.4752	-0.0753	47.6417
Multiple Linear Regression	96.1157	0.1557	53.4844

Commented [JB128]: Good (table with the main results). Even better would be an illustration. Also, this table seems to be for the unpruned dataset? So, how about the pruned datasets?

Word Count: 1352 words

Commented [JB129]: Really? Are you aware that there was a hard word limit? My strong advice would be the following:
- -Stick to the word limit
- -Stick to the word limit
- -Stick to the word limit
- -Stick to the word limit
- -Stick to the word limit
- -If you don't stick to the word limit
  - oDon't mention it on the cover sheet and hope that I don't pay attention to it.

I will only mark the first 1000 words of your assignment.

Data pruning is the process of identifying and removing certain training examples to improve the performance of machine learning algorithm[2].

Commented [JB130]: Good (a definition of what data pruning refers to, and even better, there is a reference). It would have been even better if you had explained what "certain training examples" refers to.

1 INTRODUCTION

Data pruning is the process of identifying and removing certain training examples to improve the performance of machine learning algorithm[2].

In our bank marketing dataset [3], data pruning involves identification of noisy examples and detection of outliers. In our dataset, we performed removal of training examples which contained more than three missing variables in bank client information as they represent incomplete information. Thereafter, we performed data imputation using random forest algorithm on training examples containing two or fewer missing values.

After performing initial data pruning, outlier detection is performed. Initially, we checked for outliers on each attributes to ensure that our data is noise free. Further, we performed outlier detection on multiple attributes using Local Outlier Factor algorithm. After outlier detection, Logistic Regression and Random Forest were used to predict the target variable. Later, evaluation was carried out using 10-fold cross validation and

Commented [JB131]: The second and third paragraph don't belong to the introduction but to the methodology (although, the two paragraphs are generally good).

3 METHODOLOGY

3.1 Feature Selection

We have removed duration feature from the dataset as it greatly influences the target variable[3] as mentioned in the dataset description.

Commented [JB132]: And what is the target variable? If I see this correctly you don't mention what you are predicting. It is really difficult to judge how well you did, if I don't know what you wanted to achieve.

… ok, I see. You mention it in section 3.3.3. . Please explain it earlier in assignment 2. It is important that the reader knows from the beginning what shall be predicted.

The number of missing values in the default feature contributes to 79% of the sample. The number of 'yes' in default

3.3 Model Fitting

3.3.1 Splitting the Data into training and test

The entire dataset of sample size 41188 is split into 80% for training and 20% percent for testing. We have chosen test dataset such that the proportion of customers opting for term deposit matches with the training dataset

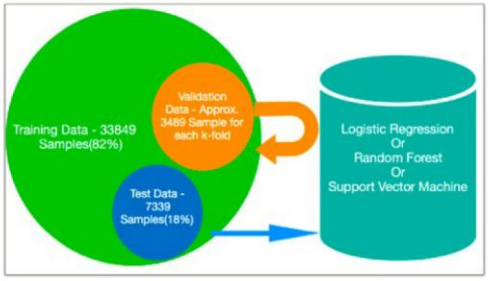


Figure 3 Model fitting Diagram

3.3.2 Performing K-fold Cross Validation

We have chosen k=10 for performing our k-fold cross-validation to maximise the amount of data while training our model.

Commented [JB133]: Good.

Commented [JB134]: I don't follow. First, you say you did a 80/20 split. But the image shows 82/18?

	Random Forest					
	10 fold cross validation				Fitting final model on training data and evaluating using Test data	
	Unpruned		Pruned		Unpruned	Pruned
	Mean	Std. Dev	Mean	Std. Dev		
Precision	0.176	0.238	0.578	0.047	0.72	0.46
Recall	0.053	0.085	0.304	0.024	0.28	0.27
F1-score	0.065	0.092	0.398	0.026	0.41	0.34

Table 2 Evaluating Precision, Recall, F1-score metric scores of the Random Forest model using pruned and unpruned datasets, by first performing K-fold cross validation and thereafter fitting the final model and using test data

Commented [JB135]: You had a standard deviation of 0.238 for the 10 folds for precision? Sure?

1 INTRODUCTION

Commented [JB136]: This reads more like a (good) abstract than an introduction.

For the past few decades, understanding the importance of data has played a key role; how it could potentially benefit an outcome when certain direction has been provided to it. Too much data risks overfitting and a data too small might not capture important information. Data pruning defines, elimination of certain unwanted data to observe an improvement in learning performance. In this paper, we use various classification algorithms, to observe the effect of data pruning on the accuracy of these classifiers. Certain statistical measures such as Z-Score and Grubbs' Test are used to identify outliers to validate our hypothesis and reduce the anomaly in the dataset. This reduces the complexity of the given dataset resulting in an increase in accuracy.

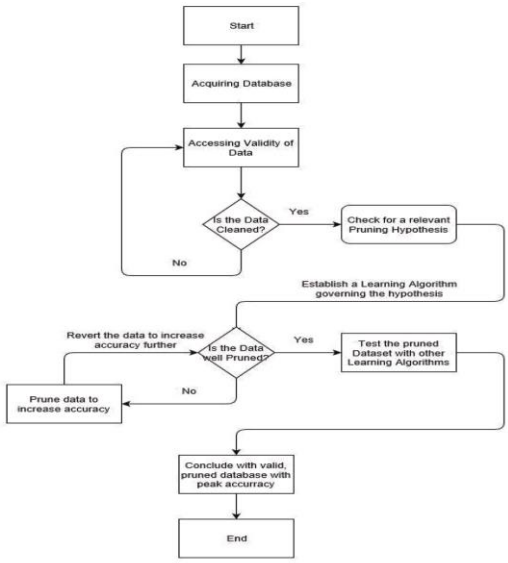


Figure 1: Flow Chart explaining the Data Pruning process.

2 Related work

Research by Anelia Angelova, DataPruning[1], dictates - "*The task is to identify if there are examples in the training set such that by eliminating them one may improve generalization performance*", which can be interpreted as, identifying outliers to prune the dataset. Removal of these "*Difficult Examples*", which are "*those which obstruct the learning process or mislead the learning algorithm*" as stated in the work would likely increase the accuracy of our models. To identify these outliers, we use the Grubbs' Test[2] to prune our dataset.

To solve our research problem, we use a dataset from Kaggle.com which provides information on various Kickstarter Projects[5] in '.csv' format containing 378661 observations and 15 variables. With the given data we are trying to predict the **success** or **failure** of a project. On analyzing the data, we explore the 'state'

Commented [JB137]: nice

Commented [JB138]: This is not a description of related work. The first part (what is data pruning) belongs to the introduction. The second part belongs to the methodology (describing what you did).

Also, please don't add hyperlinks to text.

Commented [JB139]: The dataformat (csv) is not relevant. Otherwise, good description of the dataset ´. You mention the most important things right away: the dataset you used, the number of instances, the number of features, and the target. Having a table of the 15 features would provide additional value.

variable, which describes the current state of a project. In Figure 2, the break up shows a failure rate of 52.2% and success rate of 35.4%. In Figure 3, we observe the distribution of the variable, *goal* using a linear plot, which yields an imprecise change. A logarithmic plot gives a clearer picture of the distribution, where approximately at 10^6 there is a sudden spike, which is also the

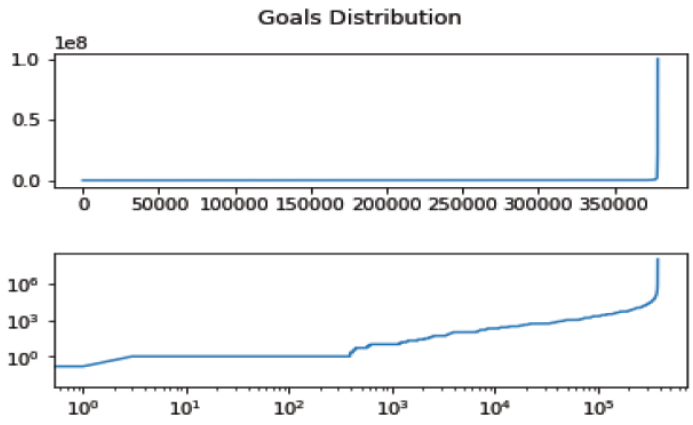


Figure 3: Distribution of the variable *'goal'*

4 RESULT

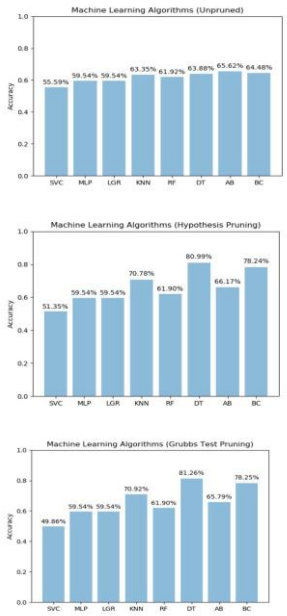
To answer our research question, we have trained Support-Vector-Cosine *SVC*, Multi-Layer-Perceptron *MLP*, Logistic-Regression *LGR*, K-Nearest-Neighbours *KNN*, Random-Forest *RF*, Decision-Tree *DT*, AdaBoost *AB* and Bagging-Classifier *BC*.

Commented [JB140]: What does the "goal variable" means? The amount of money the project is aiming for? You need to explain such variables.

Commented [JB141]: No labels on the x and y axis? How shall I know what these charts show?

Commented [JB142]: This belongs to the methodology. Generally, while it is really great that you used to many algorithms, it probably would be better to use e.g. only half the algorithms but use a second dataset instead.

On comparing the graphs representing the sklearn accuracy_score as shown in Figure 5 - Unpruned, Hypothesis-Pruned and Grubbs-Test-Pruned respectively, we see that *SVC* underperforms in our hypothesis with 51.35% accuracy and in Grubbs-Test with 49.86% but does comparatively well in the unpruned dataset with 55.59% accuracy. An increment in accuracy is noticed in KNN algorithm in both the pruned datasets, 70.78% in the Hypothesis-Pruning and 70.92% in Grubbs-Test-Pruning and a significant rise is noticed in the accuracy of the pruned datasets in Decision-Tree Classifier with 80.99% accuracy in Hypothesis-Pruning and 81.26% in Grubbs-Test-Pruning as well as Bagging Classifier, with 78.24% accuracy in Hypothesis-Pruning and 78.25% in Grubbs-Test-Pruning as compared to 64.48% in the Unpruned Dataset. Other Machine Learning algorithms show no distinct change.



Commented [JB143]: This is a good description of your results. However, as in so many other assignments the research question is not answered. You provide a long text of all the results, but so what? The value of your report would increase a lot if you had a summary of the results like e.g. "for 7 of the 10 algorithms (70%), pruning improved the accuracy for on average 12.5%". That would give a very clear idea how strong the impact of pruning was. Now, every reader needs to read through the entire paragraph and look at all the numbers him/herself to come to a conclusion.

1 INTRODUCTION

A fine line separates cleaning and pruning of a dataset. Cleaning mostly is a preprocessing step that involves removing unrequired data, data imputation, standardizing or normalizing the feature ranges and converting categorical values to numbers [2] [3]. In comparison pruning takes place after preprocessing, where certain data is strategically

Commented [JB144]: Excellent. Not only is a clear definition of pruning given, but also how it differs from other similar concepts.

3 METHODOLOGY

3.1 Dataset

The dataset chosen is from the largest publicly available movie rating database, IMDb [1]. It contains 5,043 movies with 28 attributes, with IMDb score indicating the movie ratings on a scale of 1-10. The histogram in Figure 1 shows the frequency distribution of the IMDb score indicating the rating between 6 and 7 to be the highest.

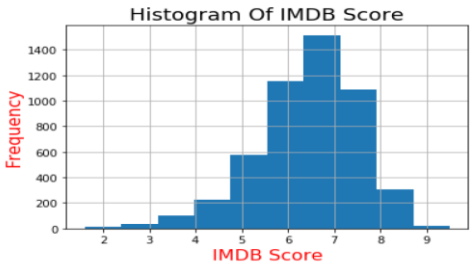


Figure 1: Frequency of IMDb Score of raw dataset

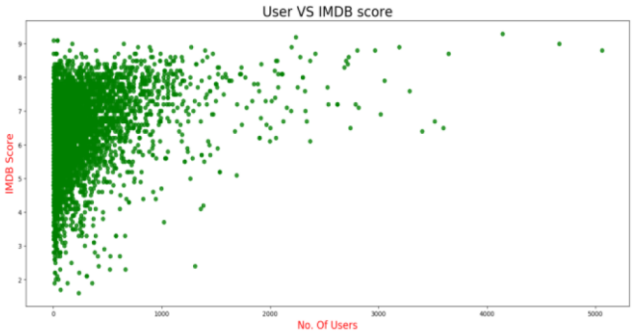


Figure 3: Plot of IMDB score v/s number of users who reviewed on a raw dataset

3.2 Pre-processing

IMDb ratings have continuous values in the range 1-10. The ratings were categorized into 5 classes: poor, average, good, very good, excellent based on the bins [0, 7, 8, 8.5, 9, 10]. Missing numeric data was imputed with the mean of the

Commented [JB145]: Good description. More details on the features would be even better. You say there are 28 features, however your heat map shows only 12.

Commented [JB146]: Wouldn't it make more sense to swap the axes?

Commented [JB147]: What is the rationale for this? I would assume you lose information by summarizing the information. Why not just use 10 "bins", one for each class, if you want to treat this as a classification problem? Generally, if you have continuous variables, treating it as a regression problem would seem to make more sense to me. For instance, with regression metrics you can easily weight errors differently (e.g. if your prediction is 3 stars off the actual start, this would be weighted differently from being only 1 star off, while in a classification problem you typically only have the binary decision "right" or "wrong".
There can be good reasons to use classification here, but the reasoning should be explained.

3.4.2 Random Forest: The n-estimator (number of decision tree classifiers) for random forest was experimented with in the range 10 to 100 in increments of 10, and it was found to be best at 40 as shown in Figure 5.

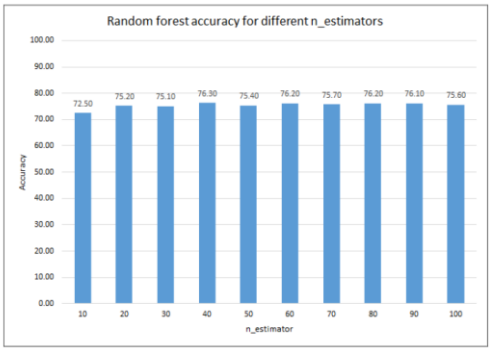


Figure 5: n-Estimator accuracy for random forest

Commented [JB148]: Good. How did you do it? K-fold cross validation? What was k?

4 RESULTS AND DISCUSSION

4.1 Metrics

The models have various predictive powers which needs proper measures to evaluate the classifier. We have used accuracy score and F1-score as the evaluation metrics [8].

4.2.1 Accuracy Score: A common metric which is the fraction of the samples correctly predicted. For a predicted value of i^{th} sample i.e. $\hat{y}_i$ and y_i being the respective true value, the fraction of right predictions over $n_{samples}$ may be defined as:

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

Commented [JB149]: The metrics you use, should also go to the methodology section. You really don't need to explain standard metrics such as accuracy, unless you believe it might not be totally clear how accuracy was calculated. However, if you explain them, then do it with a direct reference to your actual data. In this case, explain what the predicted/true value would be. Especially since you are having a multi-class classification problem, it would be good to explain how you used accuracy here and also discuss it. For instance, you have 5 classes, right? So, if the actual class is 3, and you predict a) 1 and b 4), then both cases (a and b) would be considered as misclassified, right? That is maybe not ideal, because in one case you are 1 rating off, and in the other case you are 2 off.

4.3 Results

Some related works on movie datasets were mostly centered on regression trees while some focused improving SVM accuracy [6] [9]. We ran an unbiased analysis on the three algorithms and observed that random forest performed the best followed by logistic regression and SVC as shown in Figure 8 and Figure 9. Their rankings remain unchanged on unpruned and pruned datasets across the two metrics used. However, several iterations showed some fluctuations in their performance. To conclude, pruning of datasets didn't affect the algorithm performance rankings.

Commented [JB150]: Since your goal was not to identify the best algorithm for the given dataset, the first part here ("we observed that RF performed the best followed by…") is not relevant. You can remove it for the second assignment, and use the words to describe/discuss your results in more detail.

The second part ("To conclude, pruning…") is excellent because that is a clear 1-sentence answer to the research question.