COMP 551
Project write-up for
Mini-Project 1
Getting Started with Machine Learning

Riley Ballachay, Yajiv Luckheenarain, and Anna Decterov

Linear Regression and Logistic Regression are two commonly used Machine Learning (ML) models used today. In this project, our team investigated the performance of two machine learning models on two data sets. The first data set mapped the characteristics of different buildings to their respective heat and cool load. The second data set predicted the bankruptcy of companies given qualitative features. After cleaning the data and converting it to pandas data sets, an analytical and numerical linear regression solution was implemented for the first set and a logistic regression solution for the second set. The performance of each solution was evaluated using k-fold cross-validation and bootstrap validation. The resulting solution yielded weights which allowed us to infer the importance of certain features on the corresponding labels. A stochastic gradient descent class was implemented. Different characteristics of the descent were altered to optimize the convergence speed of the model such as its learning rate of the model and the size of the mini-batch used to update the gradient. In the first data set, a mini-batch of size 64 optimized the SGD performance while for the second data set, a mini-batch of size 16 was optimal. Likewise, a learning rate of 1.0 was optimal for the first data set while a learning rate of 2.0 was optimal for the second data set. The performance of an SGD Mini-batch gradient descent solution was compared to the analytical linear regression solution. Finally, we explored the accuracy of a model trained on the first data set under a linear basis against a Gaussian basis.

I. Introduction

MACHINE learning regression models are used for predicting numerical values based on input features. The goal of a regression model is to fit a mathematical equation to the data that can then be used to make predictions on unseen data. The model is trained on a labelled data set, where the target variable is numerical and the input features can be either numerical or categorical. In our project, we implemented two models learned in class: Linear regression and Logistic Regression on data sets from the UCI Machine learning repository stored in CSV (Comma Separated Values) file format. We implemented linear regression on the Energy Efficiency data set. It contains information on the physical characteristic of various buildings and their energy efficiency, as measured by two target variables: heating load and cooling load. The data contains data on 768 buildings, where each building is represented by a set of 8 input features [2]. The input features are described in *Table 1*.

We implemented logistic regression on the Qualitative Bankruptcy data set, a financial data set used for predicting bankruptcy. It contains data on companies and whether they have filed for bankruptcy or not. The data is stored in a CSV (Comma Separated Values) file format and contains data on 66 companies, where each company is represented by a set of 6 financial ratios and qualitative factors [1]. These features are represented in *Table 2*.

Preceding any model training, we cleaned up the original data sets such that they adhere to a readable format. In our case, this format was a pandas data frame. A LinearRegression, LogisticRegression and BatchGradientDescent class were developed. The LinearRegression class trains a model using either the analytical LinearRegression solution or a numerical solution making use of the BatchGradientDescent class. A CrossValidation class was written as the super class of both a KFoldValidation and BootStrapValidation classes. Their purpose is to measure the performance of any trained models by measuring how accurately they can predict the data on which the model was originally trained (training set) as well as complementary data from the same data set (testing set).

We explored the effect of choosing different testing and training subset ratios on the performance of the trained model. As our data was split randomly, we could not observe a correlation between the split ratio and the performance of our model.

An Experiment Class was developed which made use of the regression classes, the validation classes, as well as more parameters such as learning rates, mini-batch sizes and testing fractions. This class was made to compute all the different possible combinations of parameters used whether we used stochastic gradient descent to solve a regression problem or an analytical solution. Throughout the project, instances of this experiment class are called to observe the effect on the convergence speed of our SGD with different learning rates and different mini-batch sizes.

Learning rate and mini-batch size can be optimized to yield a faster convergence speed of our model towards the lowest cost solution. These values are dependent on the data set as we measured that the optimal value of these characteristics is different when it came to the models trained on the first and second data sets.

Finally, we implemented an analytical linear regression solution on the first data set, but we altered the basis of our instances from a linear basis to a Gaussian basis. The performance of this new model did poorly compared to the initial one. This suggests that the building characteristic of the first data set most likely follow a linear trend in terms of their influence on the building's energy efficiency.

Table 1: Dataset 1 parameters		
x1	Relative Compactness	A measure of the compactness of the building
x2	Surface Area	The surface area of the building
x3	Wall Area	The total area of the walls of the building
x4	Roof Area	The total area of the roof of the building
x5	Overall Height	The height of the building
x6	Orientation	The orientation of the building
x7	Glazing Area	The total area of the windows in the building
x8	Glazing Area Distribution	The distribution of the glazing area in the building
y1	Heating Load	The target variable, representing the heating load of the building
y2	Cooling Load	The target variable, representing the cooling load of the building

Table 2: Dataset 2 parameters	
x1	Industrial Risk
x2	Management Risk
x3	Financial Flexibility
x4	Credibility
x5	Competitiveness
x6	Operating Risk
y1	Class

II. Datasets

The energy efficiency dataset used 8 input features. These feature are represented in the following table:

We had to process the data sets to make sure they fit a format before using them to train our models. The first data set was an excel sheet which made it easier to import as python's pandas has a built-in "read_csv()" function that allowed us to open the data set in a data frame.

There was no missing or corrupted information in the first set. However, some of the features in the set were highly correlated. These were the features "X1 and X2" and "X4 and X5", as can be seen in the following correlation matrix.

We decided to drop features that were highly correlated as they yield similar contributions to the model's training, but increase the complexity of the model which makes it more prone to errors. We dropped features X2 and X5.

The second data set was in a "rar" extension. A bit more work was done to convert it into a data frame. There was no missing or corrupted information in the second set. There was no correlation between features that was high enough to make us drop features. The second data set was composed of qualitative data instead of quantitative one.

Features were distributed among three possibilities: N stands for Negative, A stands for Average, and P stands for Positive. The labels were either B which stands for bankrupt or NB which stands for not bankrupt.

We made the decision to map every entry to numbers so we could attribute a "weight" to each feature and a class to each of the two labels. We mapped bankrupt and not bankrupt to classes 1 and 0 respectively. We mapped Positive, Average, and Negative to 1, 0, and -1 respectively.

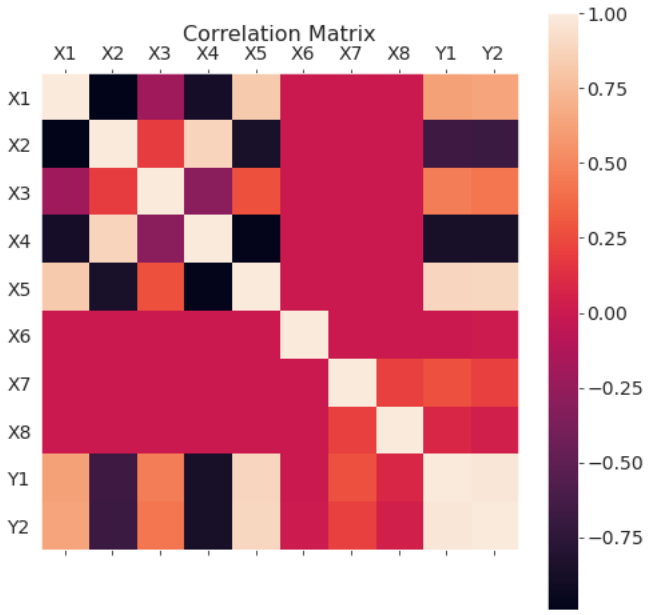

Fig. 1. Correlation matrix for Dataset 1. Color indicates pearson correlation beteewn feature row and column, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation).

These types of models are prone to ethical concerns. In a regression, the weights associated with features are only dependent on how they reduce the loss of the model on the data set. It is important to note that aside from the mathematical side of the model's training, it is possible that some features are determined to hold a higher weight by the model, but that feature might be a by-product of an entirely different reason that our data set does not consider. In other words, that feature may not intrinsically hold the weight that our model associated with it. Correlation does not equal causation. For example, with the energy efficiency model, a building could be labelled as non-energy efficient for a feature that does not in reality impact the building's energy use. This would result in the property owner getting perhaps billed more for no reason if there is an additional tax for non-energy efficient buildings.

III. Results

A. Experiment 1

We trained a model using linear regression and fully batched logistic regression on two different data sets. We used an 80%/20% train/test split and performed 5-fold cross-validation to evaluate its performance on both the training and test sets. The splits were randomized to avoid bias.

The results for linear regression showed that the accuracy of our model was quite consistent when tested against both the training and testing sets. The k-fold cross-validation method confirmed the robustness of the model, as the accuracy scores remained consistent across all five folds.

The performance of logistic regression was also impressive. With the original data set containing 250 samples, a 5-fold

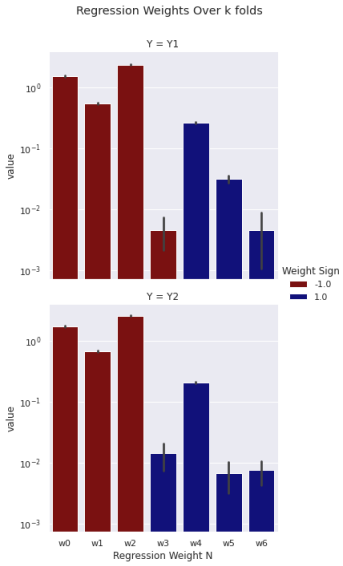

Fig. 2. Model weights for linear regression. Y-scale is log10. Top row corresponds to Y1 and bottom to Y2. Error bars show confidence interval over 5 folds. Color corresponds to sign of weight (red for negative, blue for positive).

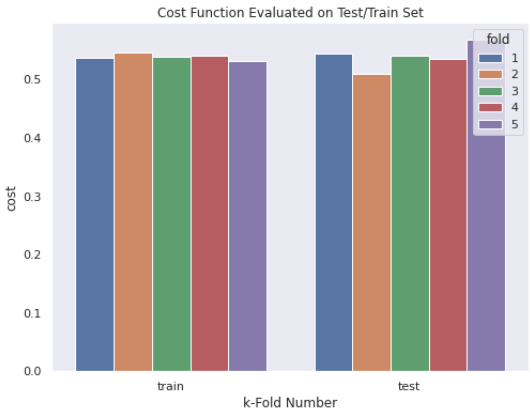

Fig. 3. Cost of logistic regression after training on test (left) and training (right) sets. Each color corresponds to a fold of L-fold cross-validation.

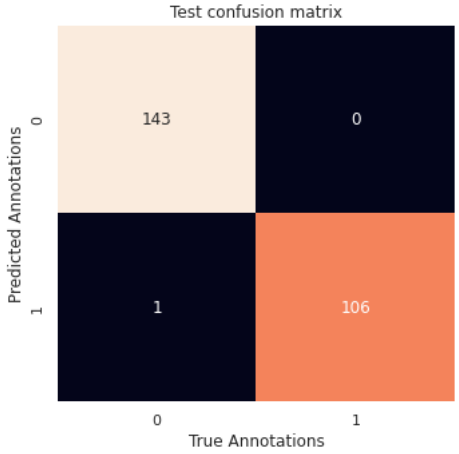

Fig. 4. Confusion matrix for logistic regression on testing set. X-axis shows true annotations + y-axis predicted annotations. Top right and bottom left refer to wrongly classified annotations.

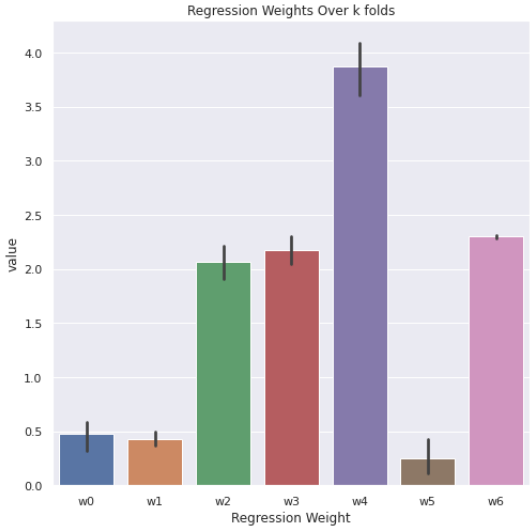

Fig. 5. Model weights for logistic regression. Color and x-label refers to weight number.

cross-validation implies that the model's accuracy was evaluated on 1000 samples for the training set, and 250 samples for the testing set. The logistic regression model could accurately predict the outcomes for approximately 99% of the samples for both sets.

B. Experiment 2

The weights of each feature in the first dataset are reported in *Fig 2*. Weights are almost identical for y1 and y2 where w2 and w0 are the highest weights, corresponding to features x1 and x3. These features are positive. This makes sense since in compact buildings (x1), the ratio of surface area to volume is lower, meaning that there is less exterior surface area through which heat can escape or enter. This can lead to lower heating and cooling costs. Having more wall area (x2) could suggest that a building is better insulated, with more material separating it from the outside. This makes this feature also very important for energy efficiency.

For the second dataset, the highest weight by far is w4, corresponding to feature x5. This makes sense since competition (x5) directly impacts the financial state of a company by reducing sales. The lowest weights are for features x1, x2 and x6. These are all risk-related features, and I believe that their weights may be smaller since we would see their effects only further in the future. It is important to note that a lot of the features in data sets are interrelated.

C. Experiment 3

Similarly to Experiment 1, we trained a model through linear and logistic regression. The difference lies in the train/test split of the data set. Growing subsets of the training data were used to train our model ranging from 20% to 80% in increments of 10%.

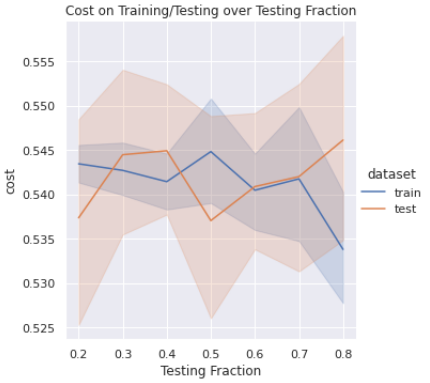

Fig. 6. Final cost as a function of testing fraction for logistic regression, repeated over multiple bootstrap samples. Orange showing testing cost, blue training

The performance of each of the models trained with a different subset of training data was tested on both the training and testing set. This performance was then plotted as a function of the fraction of testing data used in the model's training for both sets.

This experiment was run multiple times and yielded different results in each iteration as the subsets of our data were always chosen randomly. The plots that resulted from these iterations did not demonstrate any distinguishable correlation between the size of the training data and the performance of our model for both linear and logistic regression.

D. Experiment 4

Returning to an 80%/20% train/test split, we once again trained our models using linear and logistic regression. This time, stochastic gradient descent with a mini-batch was used to train our models.

Different mini-batch of size 2^k, with k taking values between 3 and 6, were used to train our models. The model's convergence speed and performance were plotted as a function of the mini-batch size and the number of gradient descent steps necessary to converge to a solution.

As can be seen from our convergence plots, our model trained through linear regression converged faster with a mini-batch of size 64, while our model trained through logistic regression performed better with a mini-batch size of 16.

E. Experiment 5

The convergence speed and performance of our models were measured as a function of said learning rates. The corresponding plots demonstrate that the model trained through linear regression converged faster with a learning rate of 1.0 while the model trained through logistic regression converged faster with a learning rate of 2.0.

F. Experiment 6

One of our models was trained using the analytical linear regression solution and the other was trained through stochastic gradient descent.

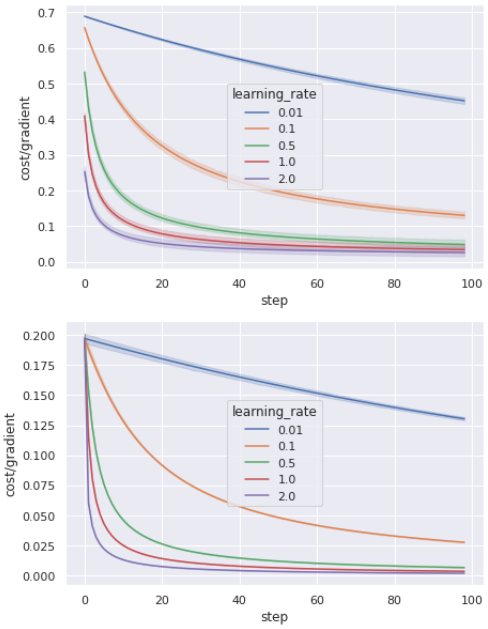

Fig. 7. Cost and gradient for logistic regression as a function of iteration. Top plot shows the cost, bottom plot shows the gradient. Each color represents a different learning rate.

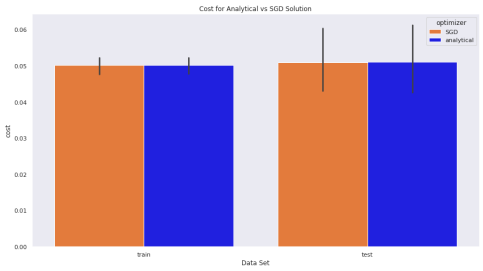

Fig. 8. Cost for analytical vs support gradient descent solution. SGD used with epsilon 1e-100, maxiters 1e5, learning rate 0.1. Left is test set, right is train.

The performance of both models ended up being the same. The stochastic gradient descent approach converges to the analytical answer given enough iterations and an appropriate learning rate.

In terms of convergence speed, however, the analytical solution is unmatched as it requires no stepping. The stochastic gradient descent falls short of the answer given only 10^3 iterations but converges given 10^5 iterations. The processing time of both solutions is approximately respectively 0.13 seconds against 50 seconds, which demonstrates the superiority of the analytical solution.

G. Extra Experiment 7

The same experiment as experiment 1 was run on the first data set, but instead of using a linear basis in the design matrix, a Gaussian base was used. As the data was already standardized, simply applying a Gaussian function to each feature was sufficient.

The model was trained using the analytical linear regression solution. Its performance was evaluated using a 5-fold cross-

validation. The cost of the trained model was evaluated on its training and testing set and compared to the performance of the linear base analytical regression solution's performance.

The performance of the Gaussian base linear regression was poor compared to its linear base counterpart. The cost of each model evaluated on both their training and testing set shows a large disparity in costs, where the Gaussian base cost far exceeds the linear base ones.

IV. Discussion

The first experiment of this mini-project investigated the performance of a model that was trained through linear regression using the analytical solution and logistic regression. All of the following experiments investigated the impact that the different parameters of a stochastic gradient descent solution could yield on the training of a model.

We separated the data set into a training and testing subset and we validated our model using k-fold cross-validation. Varying splits were used such as 20% testing to 80% testing. The splits were randomized to avoid bias and we did not see any correlation between the testing fraction that the model was trained on and the accuracy of the model.

Our results showed that full-batch gradient descent can be outclassed by mini-batch gradient descent in terms of convergence speed depending on the batch size. It is important to note that choosing a batch that is too small in size considerably slows down the gradient descent's converging speed.

Varying the learning rate of the stochastic gradient descent also had a large influence on the convergence speed of the descent. Choosing a learning rate that is too small would sometimes not converge to the optimal solution in the number of iterations that we allowed the descent to have. Choosing a learning rate that is too high is not optimal as the descent oscillates around the solution before settling to it.

The cost function for linear regression is convex. The stochastic gradient descent solution using linear regression converges to the analytical linear regression solution given enough steps in the descent. However, the computing time to reach that same solution is larger, sometimes by a factor of 500. It is possible to optimize SGD for it to converge to a solution faster, but it will never be faster than the analytical solution of the regression when it comes to linear regression.

Another extra experiment was run to compare the performance of a model that was trained through the analytical solution of linear regression and another model which was trained in the exact same way.

The difference between both models was that instead of using a linear basis for our instances in the second model's training, we used a Gaussian base. Unfortunately, the performance of that second model was poor compared to the original. A change of basis for our instance could make it more accurate as well as it could make it worse. The issue here is that the features of the first data set are more likely to follow a linear basis than a Gaussian one as they are characteristic of a house mapped to the heat and cool load of a building.

V. Conclusion

There are multiple ways in machine learning to train models. Each of these possibilities is dependent on a multitude of characteristics such as learning rate and mini-batch size. Analytical solutions sometimes exist, but most of the time a numerical gradient descent solution needs to be implemented. It is important to understand how to optimize the training of a model to allow it to converge toward a solution fast, but also accurately. We have investigated those different characteristics, but only a limited set of data. The next step would be to train our models against different sets of data which are potentially better fitted using a non-linear basis and explore new implementations of the stochastic gradient descent that may optimize the convergence speed of the model's training.

VI. Statement of Contributions

Riley worked on the setting up the workbooks and python objects to store results and plotted results. Yajiv designed, ran and interpreted experiments and wrote report. Anna interpreted results and wrote report.

References

[1] Martin Aruldoss, T. Lakshmi, and V. Venkatesan. "An Analysis on Qualitative Bankruptcy Prediction Rules using Ant-Miner". In: *International Journal of Intelligent Systems and Applications* 6 (Jan. 2014), pp. 36–44. DOI: 10.5815/ijisa.2014.01.05.

[2] Athanasios Tsanas and Angeliki Xifara. "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools". In: *Energy and Buildings* 49 (2012), pp. 560–567. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2012.03.003. URL: https://www.sciencedirect.com/science/article/pii/S037877881200151X.