Introduction to Machine Learning and Data Mining: Project 2

Jens Leysen (s191908)
Maud Leclerc (s191975)

November 2019

1 Introduction

The goal of this report is to discuss a relevant regression and classification problem. This report follows our first report. We will shortly summarize what we learned in our first report in the next paragraph.

The Titanic dataset We are working with a titanic data set obtained from Kaggle. It contains 891 observations (distinct passengers) and 12 original attributes: Survived, Pclass, Sex, Age, SibSp, Parch, Fare, PassengerId, Name, Ticket, Embarked, Cabin.

We re-list the attributes relevant to classification and regression problems:

1. Classification:

   Relevant Attributes: Survived, Pclass, Sex, Age, SibSp, Parch, Fare

   Irrelevant Attributes: PassengerId, Name, Ticket, Embarked, Cabin

   Explanation: We are given the attributes of one passenger and have to determine whether they were likely to survive or not. The relevant attributes are attributes that can have an impact on the chance of survival of the passenger, other ones such as Name or Ticket number will not have an impact.

2. Regression:

   Relevant Attributes: Survived, Pclass, Sex, Age, SibSp, Parch, Fare

   Irrelevant Attributes: PassengerId, Name, Ticket, Cabin, Embarked

   Explanation: The fare values are described by a float between 0 and 512.3292. A regression problem would for example predict the fare of a certain passenger based on known relevant attribute values.

Data Issues We used one-out-of-K coding on the Pclass and and Embarked attributes and removed passengers that had missing/incorrect values. This left us with 712 passengers.

Summary Statistics and PCA The following table describes the summary statistics:

Attribute	Mean	Standard deviation	Median	Range
Age	29.64	14.49	28.0	79.58
SibSp	0.51	0.93	0.0	5.0
Parch	0.43	0.85	0.0	6.0
Fare	34.57	52.94	15.65	512.33

Additionally, we found out that the Age attribute can be approximated by a normal distribution while the others can't. We calculated the correlation matrix and found the strongest correlation between fare and passenger class, the passenger's class is also correlated to the age class. The last noticeable correlation is between the number of parents or children and siblings or spouse aboard. Regarding the PCA analysis, the first principal component accounts for nearly 25% of variation in the data. The first 7 principal components account for 90% of variation.

2 Regression

2.1 Part 1

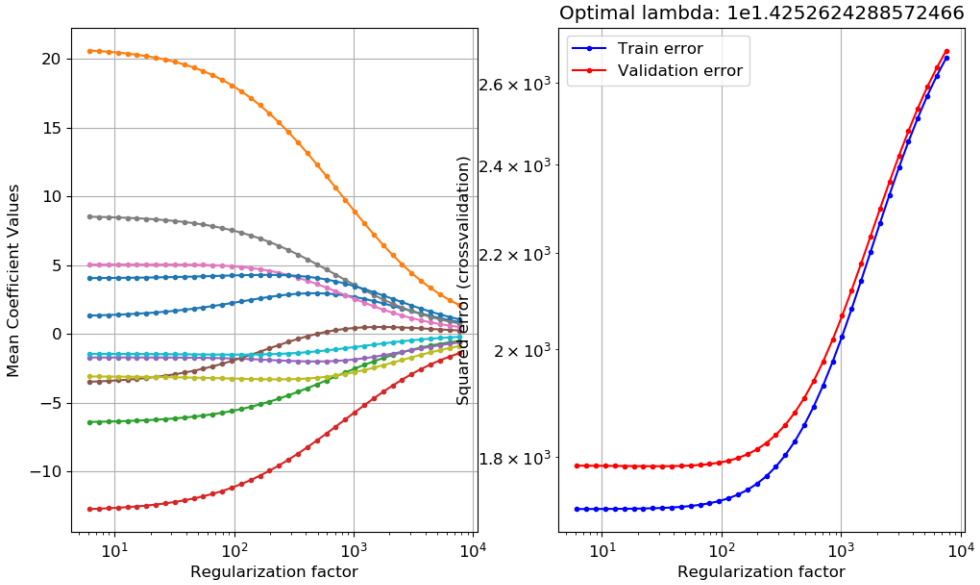
We want to predict the fare value based on the relevant attributes we found in project 1: Survived, Pclass, Sex, Age, SibSp, Parch, Embarked. As mentioned above, we applied one-out-of-K coding on the Pclass and Embarked attributes. The sex values have been changed from 'female' and 'male' to 0 and 1.

Note: The fare values are per Britain's pre-decimalised currency. The conversion formula to a decimalised value is: $\mathrm{Y} = \mathrm{P}{+}5{\cdot}(0.01)s + \frac{5}{12} \cdot (0.01)d$.

We will treat the fare value as a continuous variable and apply linear regression because the amount of distinct values is 219. In our opinion, this number is large enough to treat this problem as a regression instead of a classification problem.

We are going to compare three different models : a linear regression, a neural network and a baseline model. This baseline model is just going to be the mean of the training set. In order to compute a linear regression, the dataset has been standardized. A regularization parameter λ is introduced, with values ranging from 1.2^{10} to 1.2^{50}. A 10-fold cross-validation is then used to estimate the generalization error. The following figure displays the evolution of the generalization error for the different values of λ. The minimum value found for λ corresponds to its optimal value. After running the algorithm a few times, this optimal value seems to be between 20 and 30. This means that a λ between 20 and 30 reduces the

variance of the model, without increasing the bias to much (optimal bias-variance trade-off). We can see on the graph that a value that would be lower than 30 doesn't increase the generalization error by much.



The statistics on the linear regression confirm what can be deduced form the figure. As can be seen in the following table, for the regressions without and with feature selection, the errors (both on the training data and on the test data) are extremely close. In other words, when using the optimal lambda, there is no drastic improvement on the quality of the model.

Linear regression without feature selection:	
Training error	1583.28
Test error	1621.70
R^2 train	0.4341
R^2 test	0.4148
Regularized linear regression:	
Training error	1584.07
Test error	1621.23
R^2 train	0.4338
R^2 test	0.4150

The model with the lowest generalization error predicts new data based on the following weights :

Offset	35.13
Survived	1.57
1st class	20.03
2nd class	-6.22
3rd class	-12.38
Male	-1.73
Age	-3.1
Sibsp	5.05
Parch	8.3
S	-3.14
Q	-1.47
C	4.11

This can be interpreted as the 1st class having the biggest influence on the prediction : a passenger that is in first class is likely to purchase a more expensive ticket. Following the same logic, a 2nd class ticket is going to be less expensive, and a 3rd class ticket has quite an important negative weight, making it more likely to pay a low fare. This is logical, and was expected. The weights paired with the other attributes are less logical. For example, older people would be more likely to pay a lower fare than younger people, or big families (people travelling with their parents or siblings) would generally pay more. We have no way of checking if this was actually the case, since we don't know on what basis the ticket fares were calculated.

2.2 Part 2

In this section we will compare 3 different models.

Set-up Here we will shortly describe the set-up of our code.

The inner loop for the linear regression uses a set of lambdas that are generated using following function: np.power(1.2,range(10,50))

The rlr_validate function implements the inner loop: rlr_validate(X_train, y_train, lambdas, internal_cross_validation)

The internal_cross_validation function is set to 10. The function rlr_validate returns the optimal lambda found.

The inner loop for the ANN uses a function we programmed: neural_network(X_train,y_train,X_test,y_test) The parameters we provide may seem weird, that is because we got a strange error when we wanted to return a neural network object from this function to the calling function. We solved this problem by providing the test data (from the outer fold). The underlying function will then return the calculated mean square error for the best found neural network for the

best hidden unit count. It will also return the index of the optimal hidden unit in the array defined below.

The hidden units we test are set in the neural_network function: hidden_units_ar = [1,2,4,8,16]

The best hidden unit is the hidden unit amount that provides the best neural network e.g. has the lowest average validation error.

The outer fold will then test the optimal models returned by the functions above and calculate the mean squared error. e.g. E_i^{test} for each optimal lambda and hidden unit count.

Outer fold	ANN		Linear regression		Baseline
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	8	547.37	18.48	914.05	3592.41
2	8	574.29	12.83	3870.63	7618.75
3	8	664.00	22.18	3211.82	5504.68
4	8	657.88	22.18	872.37	2535.10
5	8	600.48	22.18	1222.35	3296.09
6	8	630.63	22.18	802.50	2856.77
7	8	746.13	7.43	2805.73	5263.51
8	8	725.32	22.18	871.27	2212.36
9	8	409.22	31.94	791.02	1862.99
10	8	737.88	26.62	880.53	3116.41

The baseline model is a constant. It is computed by taking the mean of each training set during the cross-validation process, and testing it on the test set. The value then used for the model is the one that produced the lowest generalization error. The optimal lambdas are as expected, in the previous session we did a few test runs and saw that their value was mostly in the 20-30 range. By interpreting the table, the neural networks seems to be performing better than the regression (the errors are lower), which seems to be performing better than the baseline model. The next section will try to estimate these performances statistically.

Summary of generalization errors We will estimate the generalization error of each model by averaging the test errors obtained in the table above. The result for each model is given in this section. Additionally, we will calculate the standard deviation of the test errors. (STD = Standard deviation, SEM = Standard error of mean) Next, we will apply a paired difference t-test, following the procedure outlined in: [1].

Note: We understood the procedure outlined in the book as follows: the models we train in the outer loop of the 2-fold cross-validation are not independent since there is considerable overlap between the used training sets. Therefore, it is suggested that the K-fold cross-validation procedure is repeated one or more times to get a total of J = 20/30 estimates. Given these results the procedure outlined below is followed to determine the p-value and confidence intervals. Due to time constraints, we were not able to re-run the K-fold cross-validation procedure and will only use J=K estimates.

Method	ANN	Linear Regression
Mean	629.3200	1624.2270
STD	103.1104	1187.1240
SEM	32.6064	375.4016

Method	ANN	Baseline
Mean	629.3200	3785.9070
STD	103.1104	1799.8530
SEM	32.6064	569.1635

Method	Baseline	Linear Regression
Mean	3785.9070	1624.2270
STD	1799.8530	1187.1240
SEM	569.1635	375.401

P-values We calculated the p-values as follows:

1. Calculate difference of mean values.
2. Calculate the standard error of difference. (= square root of the sum of the the squares of the two SEMs)
3. df = 9.
4. Calculate t-value as step1/step2.
5. Use a t-distribution table to find the p-value.

ANN-Linear Regression	p=0.0256
ANN-Baseline	p less than 0.0001
Baseline-Linear Regression	p=0.0003

Our significance level α was chosen to be 0.05. The above p-values (for the null hypothesis that the two models have the same performance) is statistically significant for all pairs. The p-value captures how unlikely it is to observe a value at least as extreme as we found (given H0 is true), hence the smaller the p-value, the more reason we have to doubt H0. So, since the p-values are all smaller than α, we consider them statistically significant and reject the null hypothesis.

Confidence intervals Here we calculate the 95% confidence intervals for the difference between model pairs.

ANN-Linear Regression	[-1837.4552,-152.3588]
ANN-Baseline	[-4436.7735,-1876.4005]
Baseline-Linear Regression	[1627.2490,2696.1110]

We see that the 95% confidence intervals never contain zero (the intervals for the difference between the two means (per pair) does not include zero). All values outside the interval are rejected as plausible values for the mean difference e.g. we reject the null hypothesis.

Discussion Since the above p-values are all statistically significant, we can conclude that there is a performance difference between the 3 methods. Both models are better than the baseline. We would recommend using the ANN model since it clearly has the best performance according to the discussion above.

We verified these results through: [2].

3 Classification

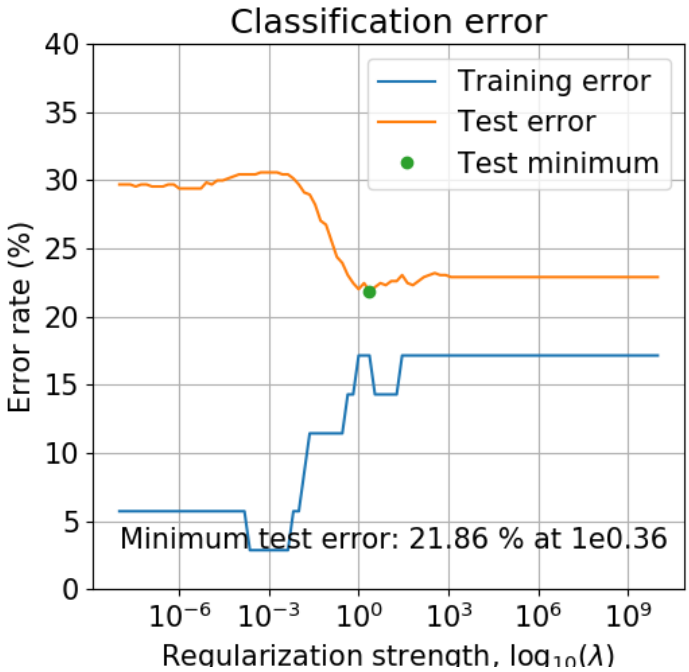
We have chosen to discuss a binary classification problem. We are given the attributes of a passenger and want to determine whether they were likely to survive or not. The relevant attributes are attributes that can have an impact on the chance of survival of the passenger, other ones such as Name or Ticket number will not have an impact. We are then basing our models on the following attributes (as stated in the summary) : Survived, Pclass, Sex, Age, SibSp, Parch, Fare.

In this part, we are going to compare a logistic regression, a neural network and a baseline function. The baseline is going to compute which one of the two classes is the largest on the dataset, and predict that all the data then belongs to this class.

3.1 Logistic regression

The logistic regression function models the density of probability of an observation to belong to one class or the other by considering that the output is a Bernouilli variable.

The regularization parameter λ varies between 10^{-8} and 10^{100} at a logarithmic rhythm. The final model is selected through a 10-fold cross-validation. The lowest generalization error is found for a λ around 15. For instance, for the model that the following graph has been issued from, the lowest generalization error is 21.86%, for λ=14.33.



According to the same graph, introducing a regularization parameter has an impact on the quality of the prediction, because the test error lowers (to a certain point) when we increase the value of λ. This means that a model with a regularization parameter lying between 2 and 3 is generally going to perform better than without any regularization. Without this λ, the model is probably overfitting the data, and reducing its variance enables it to make better predictions.

The weights computed by the logistic regression are the following:

1st class	0.06
2nd class	0.25
3rd class	0.11
Male	-0.28
Age	-0.38
Sibsp	-0.41
Parch	-0.47
Fare	-0.32
S	0.3
Q	0.02
C	-0.11

From this weights, we can estimate that the logistic regression model gives more importance to the number of relatives of a passenger, their age, their sex and the fare they paid than to the other attributes. For instance, a young female passenger travelling alone would be more likely to survive than an old man travelling with a big family.

3.2 ANN

The neural network uses an inner loop that we programmed in a similar manner to what we did for the regression part of the report, which is a function named neural_network_classification. It will return the error for the best model found for the best hidden unit count, as well as the index of the optimal hidden unit in this array : hidden_units_ar = [1,2,4,8,16] , that is defined inside of the function.

3.3 Model comparison

Setup The three models (regression, ANN and baseline) have been computed on a cross-validation algorithm, using the same training and testing sets for each of the 10 outer folds. The logistic regression is then trained on 10 more inner folds, while, to speed up the process, the ANN uses only 2. The optimal optimization parameters (λ and h) have been recorded for each fold, and displayed in the following table, as well as the error associated with each one of the folds E_i^{test}. This error is determined by dividing the number of misclassified observations by the number of observations of the test set.

Outer fold	ANN		Linear regression		Baseline
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	2	0.1656	3.511	0.2222	0.3888
2	2	0.1781	151.99	0.1944	0.4027
3	2	0.1937	18.73	0.1971	0.4647
4	2	0.1687	1e-08	0.2394	0.4929
5	2	0.2000	1e-08	0.2394	0.3943
6	2	0.1843	1e-08	0.1267	0.4366
7	2	0.1593	1e-08	0.2112	0.3802
8	2	0.1843	811.13	0.2535	0.3802
9	2	0.1906	1e-08	0.1971	0.2957
10	2	0.2031	5.3366	0.1549	0.4084

Statistics As said before, the error is determined by dividing the number of misclassified observations by the number of observations of the test set. The estimated differrence in accuracy can be found by substracting these two values. The optimal lambdas seem to be varying in a large range, but most of them are quite small, which was expected from the test runs. It seems like the ANN is performing better than the logistic regression, which is better than the baseline. Note: Due to time constraints, we were unable to do a thorough statistical evaluation of the classification part.

4 Discussion

First of all, for the regression part, we wanted to predict the ticket fare paid by a passenger according to their attributes. We compared a linear regression, a neural network and a baseline function. We trained and tested them on the same splits. We found in our statistical evaluation that the neural network model performed the best, followed by the linear regression and then the baseline. But even the neural network has a hard time predicting the fare. This isn't really surprising, because when we look at the data, the passengers paid prices that could vary a lot. Some outliers values are quite likely to confuse the model, even if they are right. As we discussed in the previous report, there is for example a passenger that paid for his employees : his fare is going to be very high, while his employees will have a fare attribute of 0. This kind of data could explain why the model doesn't perform well.

Secondly, for the classification part, we wished to predict whether or not a passenger would survive (binary classification problem). We created three models: a logistic regression, a neural network and a baseline. We then trained them on the same splits and compared them on the number of missclassified observations. It seems that for this part as well, the neural network is the best performing model of the three. The errors are quite low, the model has between 80% and 85% accuracy on the different trials. It would probably be interesting to train an other type of classifier, such as a classification tree or a nearest neighbor model, and compare them, maybe they would perform differently. Note: Due to time constraints, we were unable to do a thorough statistical evaluation of the classification part.

Trying different machine learning models to compare them, and maybe combine them (in a voting system for example) seems to be what other people did with this same Titanic dataset. We chose to look at some of the analysis displayed on Kaggle (where we downloaded the dataset from). People are mostly interested in studying the survival chances of an individual, so this is just going to be relevant in comparison to the classification part of this report.

The best (as in more popular) analyses are going a step further than we did when it comes to preparing the dataset before establishing a model. For example, this study [3] analysed the names of the passengers (sorting people by their titles, such as Mr, Miss, Sir, Lady,...). They also grouped the number of parents, spouses, siblings and children in the same "family size" attribute. This analysis ([4]) did the same thing for the family, and then split it in 4 classes of different family sizes.

When it comes to computing models, the first analysis [3] did a 10-split K-fold, with a proportion of 70% - 30% for the train and test dataset sizes. Different classifiers (SVC, Decision trees, Nearset neighbors,...) are then trained on the same splits. Most of the models achieve around 80% accuracy.

The second analysis [4] also does a 10-split K-fold to test a range of classifiers. The hyperparameters for some of the models (for example SVC, AdaBoost or RandomForest) are then tuned. The classifiers are compared, and combined, to obtain better accuracy.

In both cases, it is quite hard to compare the results obtained, since the feature engineering

is way more advanced that we did. There is in general a lot more pre-processing on the data, so it would make sense that the classifiers from these analysis perform better than ours. They also used logistic regression and neural networks, but since other types of models end up being more accurate, these two are left out when combining classifiers at the end. But on a larger scale, the process is very similar to what we did, in terms of which attributes were considered relevant and used for the analysis, and the amount of training for the models.

	X	MASHvstRap	MASHvsBEEML	tRapvsBEEML	frequency	Mash_mean	BEEML_mean	tRap_mean
1	ETS	8.95e-04	7.35e-04	4.78e-06	10	0.52	0.67	0.30
11	ZnF_C2H2	7.08e-21	2.09e-02	1.70e-26	54	0.55	0.64	0.25
10	Zn2Cys6	4.94e-04	5.50e-02	3.52e-06	17	0.38	0.61	0.13
8	IRF	1.16e-06	6.65e-02	5.54e-08	10	0.52	0.62	0.28
2	FH	1.27e-05	8.61e-02	5.20e-07	10	0.53	0.66	0.27
3	HLH	2.49e-05	1.31e+00	4.27e-05	13	0.61	0.74	0.26
4	HMG	8.73e-33	1.41e+00	3.49e-08	44	0.55	0.48	0.12
12	ZnF_C4	2.92e-06	1.92e+00	1.03e-07	10	0.66	0.73	0.27
9	unknown	3.15e-27	1.96e+00	5.38e-21	121	0.44	0.49	0.16
5	Homeo	1.69e-164	6.26e+00	2.35e-75	158	0.72	0.73	0.17
7	Homeo, POU	3.12e-12	7.36e+00	5.21e-12	11	0.69	0.70	0.18
6	Homeo	9.82e-13	9.73e+00	1.21e-05	19	0.67	0.65	0.14

Table 1: Paired t-test of most common TF families for Pearson Correlations

5 Attribution Table

Section	Responsibility
Introduction and Summary of project 1	Jens (100%)
Regression part 1	Jens (60%) / Maud (40%)
Regression part 2	Jens (80%) / Maud (20%)
Classification	Maud (70%) / Jens (30%)
Discussion	Maud (100%)

References

[1] McClave, J. T., & Sincich, T. (2013). Statistics. Hoboken, NJ: Pearson Education, Inc.

[2] `https://www.graphpad.com/quickcalcs/ttest1/`

[3] Beginners Notebook to achieve 80% accuracy
`https://www.kaggle.com/loyashoib/beginners-notebook-to-achieve-80-accuracy`

[4] Titanic Top 4% with ensemble modeling
`https://www.kaggle.com/yassineghouzam/titanic-top-4-with-ensemble-modeling`