# List of figures

[**Figure 1** Confusion matrices for the prediction of default for the credit card data using scikit-learn’s default encoding versus OneHotEncoder. 3](#_Toc24581528)

[**Figure 2** Accuracy of models for logistic regression with mini-batch sizes of 5, 10, 20, 40, 80 and 160 observations. 3](#_Toc24581529)

[**Figure 3** Accuracy of models for logistic regression using mini-batches vs. the entire training set at once per epoch (a) and difference in percentage between both (b). 4](#_Toc24581530)

[**Figure 4** Accuracy of models for logistic regression using mini-batches vs. the entire training set at once per epoch and variable vs fixed learning rate. 4](#_Toc24581531)

[**Figure 5** Accuracy score of predictions for the credit card data using a multilayer perceptron. Values for the lambda parameter are plotted on the x axis while learning rates are on the y axis. 5](#_Toc24581532)

[**Figure 6** Accuracy score of predictions for the credit card data (a) and confusion matrix (b) using a multilayer perceptron for classification. 6](#_Toc24581533)

[**Figure 7** Variation partitioning of groups of predictors for the credit card data. 6](#_Toc24581534)

[**Figure 8** Mean squared error (MSE) of predictions for the Franke function by regression using a multilayer perceptron. 7](#_Toc24581535)

[**Figure 9** Franke function predicted by regression using a multilayer perceptron (a) from generated data with added noise (b). 7](#_Toc24581536)

# Methods

## Data filtering, formatting and transformation

The credit card dataset contained the categorical variables gender, education, marital status, as well as history of past payments. Except for gender, there were erroneous values found for each of these descriptors. Accordingly, observations possessing incorrect values were removed from the dataset prior to downstream analyses. Continuous variables were scaled by dividing values by column standard deviation and substracting column mean. For the purpose of multivariate analyses performed in R, categorical variables were assigned classes of factor and ordered factor in the case of gender, marital status and default for the former while education and history of past payment were assigned the latter.

## Dimensionality reduction

Since the Gower distance metric is not Euclidean, it is not possible to use it for principal component analysis (PCA). Instead, a non-metric multidimensional scaling (nMDS) is performed to visualize the variation in the data. The relevance of using nMDS over PCA here is to perform an analysis on all variables at once, because the latter being of mixed types makes it meaningless to include in a PCA factors that would have to be treated as numeric. Accordingly, redundancy analysis was used to evaluate the contribution of the various predictors to default status. Variation partitioning was performed to examine shared contributions in explained variation.

## Logistic regression

Using the credit card data, a comparison in prediction accuracy was made with scikit-learn using its default encoding against OneHotEncoder. When encoding with OneHotEncoder, OneHotEncoder.fit\_transform() is used once on one subset of the data, after which OneHotEncoder.transform() is used on the associated subsets so that the same values among the subsets are attributed the same classes. The accuracy of the predictions were assessed with confusion matrices.

The effect of batch size on accuracy increase per epoch was investigated with the random seed set for training and test data splitting, training data shuffling and initial beta values generation. Based on the results of this comparison, a mini-batch size of 20 was arbitrarily chosen. The use of mini-batches as opposed to the full training set at once per epoch was compared in terms of accuracy and execution time.

## Classification using a neural network

Classification was also performed using a multi-layer perceptron.

When computing the hidden and output layers’ weight and bias, the learning rate is divided by batch size, so as to ensure the rate of gradient descent is independent of batch size. For example, a run with a learning rate of 0.1 and batch size of 100 yields an effective learning rate of 0.001.

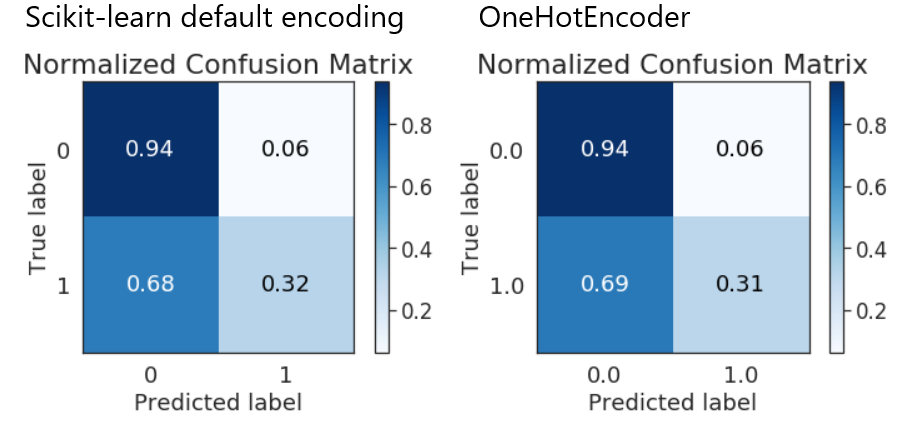
## Regression

Regression on generated Franke function data was performed using a multi-layer perceptron (MLP).

# Results

## Logistic regression

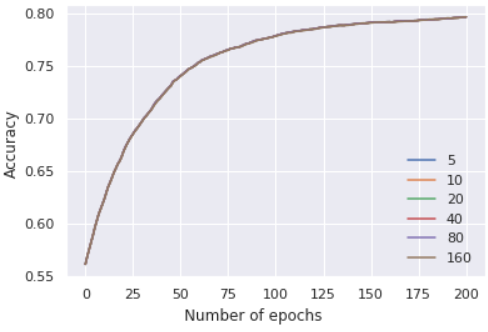
When using scikit-learn’s default encoder on the credit card data, the result is the same as with OneHotEncoder (fig. 1). It appears the y values are automatically treated as binary class rather than multiclass or continuous values, so here OneHotEncoder provides no advantage. This would be different when using TensorFlow which requires using OneHotEncoder.



**Figure 1** Confusion matrices for the prediction of default for the credit card data using scikit-learn’s default encoding versus OneHotEncoder.

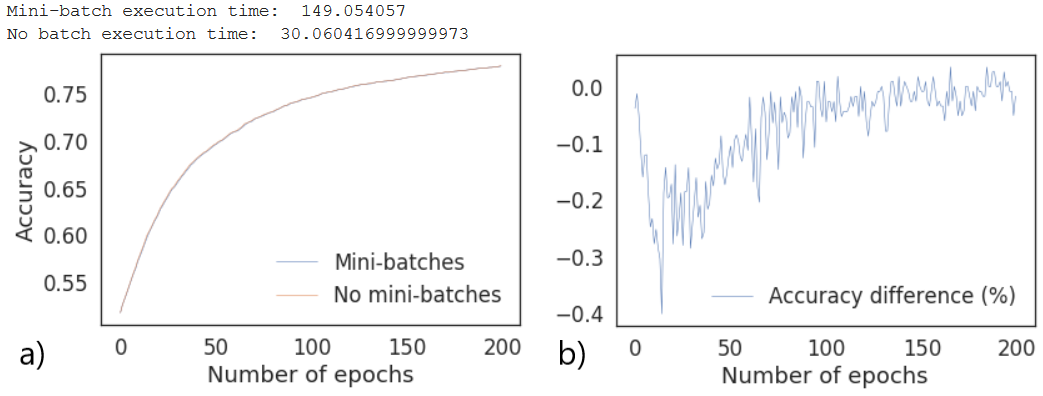
When removing rows with wrong values, there are only 4030 left with no errors. Using the dataset truncated this far results in lower accuracy for predicting default with both logistic regression and neural network. Accordingly, a dataset with unfiltered columns for past payment was used in order to have enough observations to achieve accuracies obtained in Yeh et Lien 2009. There remains 29600 observations after removal of rows with erroneous values for education and marriage.

When using a predict function for logistic regression and looking at accuracy for each epoch, it improves and tends toward an asymptote.



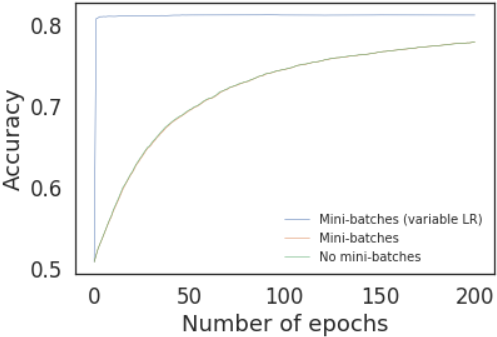
**Figure 2** Accuracy of models for logistic regression with mini-batch sizes of 5, 10, 20, 40, 80 and 160 observations.

The number of observations per mini-batch in a range from 5 to 160 resulted in no difference in terms of accuracy when each model was generated using the same random seed.



**Figure 3** Accuracy of models for logistic regression using mini-batches vs. the entire training set at once per epoch (a) and difference in percentage between both (b).

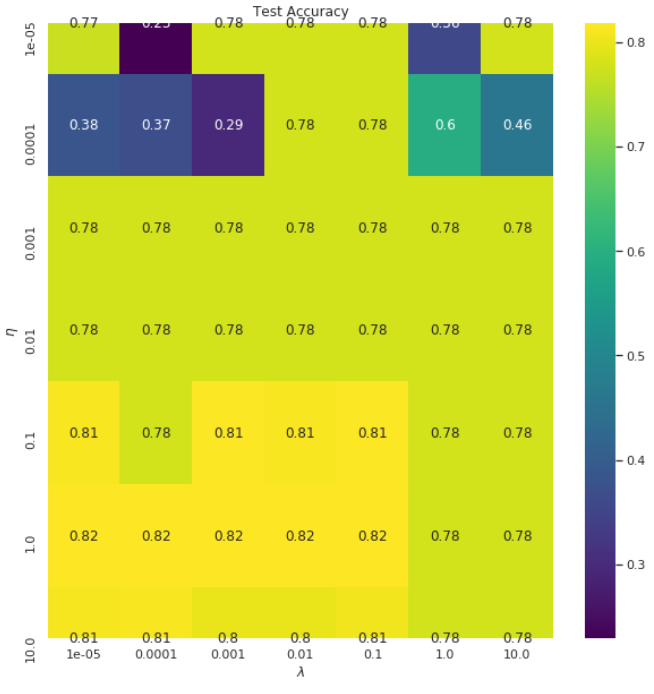
Using mini-batches with fixed learning rate yields slightly lower accuracies while execution time is greatly increased (fig.3). For the test run presented in fig. 3, the mean accuracy of the model generated with mini-batches was 0.075 % lower than without them with an execution time of 149 seconds as opposed to 30.



**Figure 4** Accuracy of models for logistic regression using mini-batches vs. the entire training set at once per epoch and variable vs fixed learning rate.

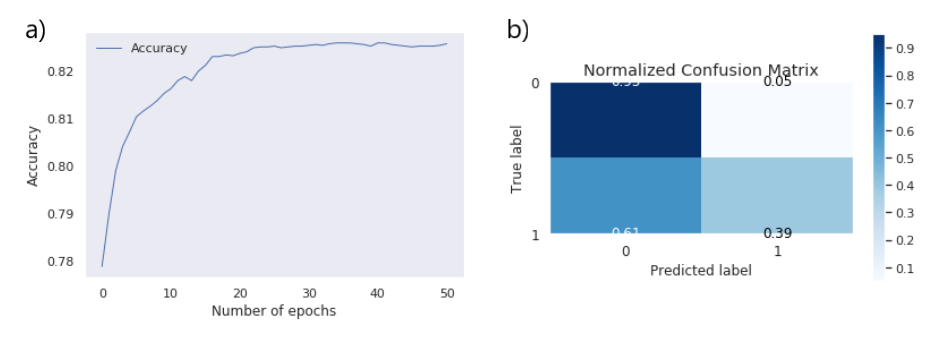
Implementing a variable learning rate for logistic regression with mini-batches yields a much faster convergence without loss of accuracy. For the test run presented in fig. 4, the accuracy reached 0.81886, which matches the result of Yeh et Lien 2009.

## Classification using a neural network



**Figure 5** Accuracy score of predictions for the credit card data using a multilayer perceptron. Values for the lambda parameter are plotted on the x axis while learning rates are on the y axis.

The grid search on learning rate and lambda hyperparameters indicated accuracy is highest with a learning rate of 1 and lambda of between 1e-05 and 0.1. These parameters were thus used for subsequent analyses. Although these parameters do no yield the smoothest accuracy curve, they return an accuracy matching the one achieved in Yeh et Lien 2009.

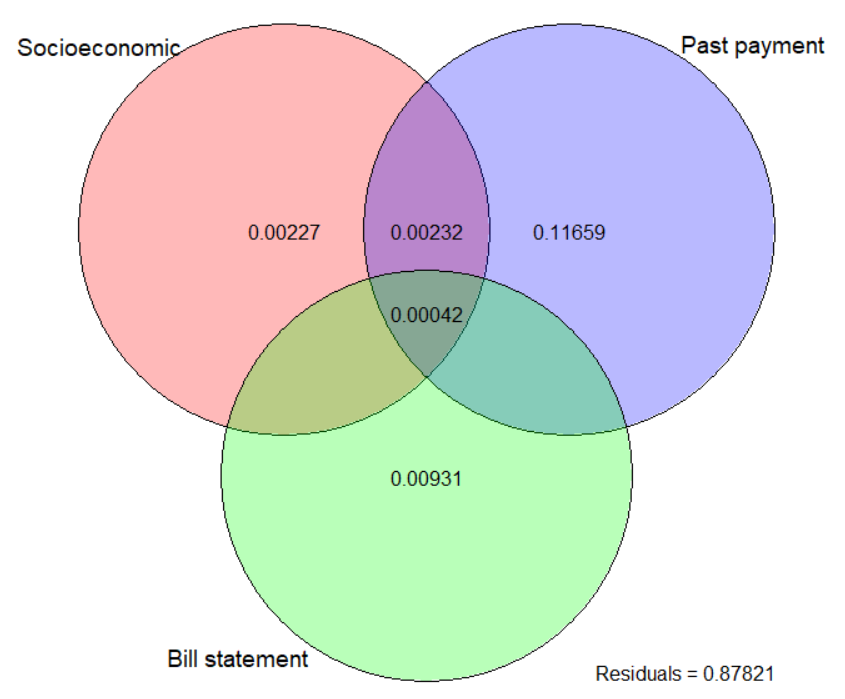


**Figure 6** Accuracy score of predictions for the credit card data (a) and confusion matrix (b) using a multilayer perceptron for classification.

The MLP achieved an accuracy of 83 % with true predictions correct 39 % of the time.

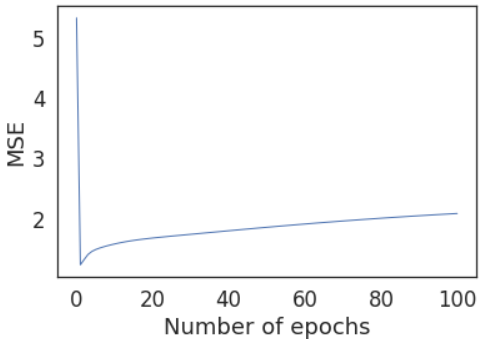
## Dimensionality reduction

Redundancy analysis of default status as response variable to all predictors yielded an adjusted R-squared value of 0.2110769. Variation partitioning of default status by socioeconomic factors, history of past payment and bill statement reveal most of the variation here can be explained by past payment and that much of the variation explained by socioeconomic and bill statement history are contained in past payment history (fig. 9). Thus, it would be arguable that only past payment be kept for further analysis if there was a need to reduce demand in computational resources without drastically affecting default prediction accuracy.



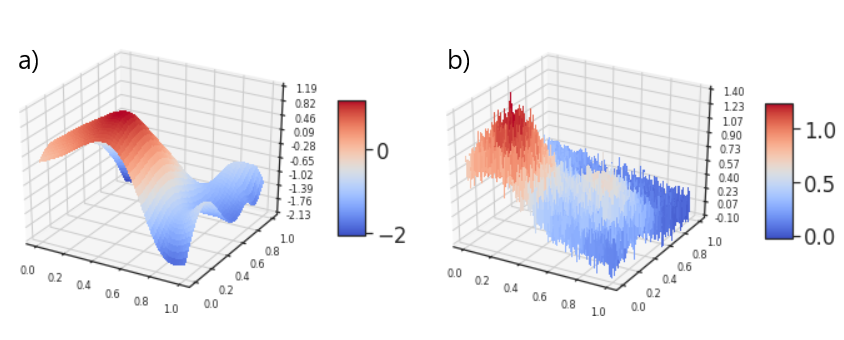
**Figure 7** Variation partitioning of groups of predictors for the credit card data.

## Regression



**Figure 8** Mean squared error (MSE) of predictions for the Franke function by regression using a multilayer perceptron.

In the test from which figures 8 and 9 were produced, regression on the Franke function noisy data using the MLP reached its lowest MSE at the first epoch. It can be assumed the increasing MSE for subsequent epochs is the result of overfitting.



**Figure 9** Franke function predicted by regression using a multilayer perceptron (a) from generated data with added noise (b).

# Discussion

Compared to the results of project 1, regression using the MLP produced in this project is less effective. For Franke function datasets of the same size and design matrix of polynomial order 5, singular value decomposition ordinary least squares and ridge regression yield MSEs of 0.013354 and 0.013302 respectively, while the MLP MSE was 0.403. There is however much improvement to be gained on the MLP MSE since the parameters were not optimized by testing factorial combinations of all of them. The latter would be of prohibitive computational cost and time. Thus, it is an advantage of linear regression methods used in project 1 to be easily optimized in comparison to a neural network.

The accuracy of the MLP for classification compared to logistic regression was a few percent better, while correct predictions of default were substantially higher with respective rates of 31 and 39 %.

# References

Yeh, I.-C., & Lien, C. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473–2480. https://doi.org/10.1016/j.eswa.2007.12.020