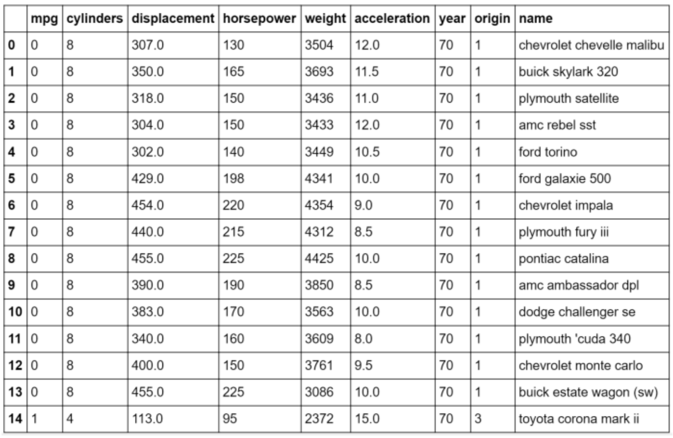
The following lab was conducted using the computer language *Python* with libraries *matplotlib, scikit-learn, pandas* and *numpy.*

Part A

The first 15 rows of the dataset are to be seen in Fig. 0.



	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
0	0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	0	8	302.0	140	3449	10.5	70	1	ford torino
5	0	8	429.0	198	4341	10.0	70	1	ford galaxie 500
6	0	8	454.0	220	4354	9.0	70	1	chevrolet impala
7	0	8	440.0	215	4312	8.5	70	1	plymouth fury iii
8	0	8	455.0	225	4425	10.0	70	1	pontiac catalina
9	0	8	390.0	190	3850	8.5	70	1	amc ambassador dpl
10	0	8	383.0	170	3563	10.0	70	1	dodge challenger se
11	0	8	340.0	160	3609	8.0	70	1	plymouth 'cuda 340
12	0	8	400.0	150	3761	9.5	70	1	chevrolet monte carlo
13	0	8	455.0	225	3086	10.0	70	1	buick estate wagon (sw)
14	1	4	113.0	95	2372	15.0	70	3	toyota corona mark ii

Fig. 0. *The first 15 rows of the dataset, out of 392.*

A few missing datapoints were noted in the *horsepower* column, which had to be removed. This was done by removing all rows with a non-integer value in that column.

a) If we were to keep the *mpg* data while training the model, the resulting model would show a strong correlation between the *mpg* data and responses *y*. Once testing the model and then removing the mpg column, the model would yield very bad results, since the (possible) correlation with other variables would have been neglected.

b) Based upon the Pearson correlation matrix, as can be seen in Fig. 1, it seems that mpg is pairwise correlated with *cylinders*, *displacement*, *weight* and *horsepower.* (Blue corresponds to a strong negative correlation and yellow to a strong positive correlation, whereas green corresponds to no correlation.) This would seem intuitive, since the previously mentioned variables are related to the engine and, to some extent, engine size. Also, a heavier vehicle must logically require more energy to move.

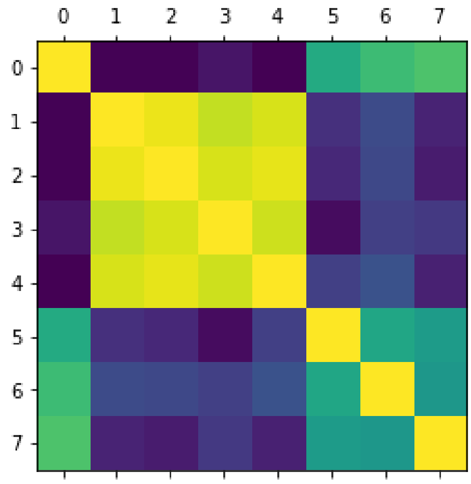


Fig. 1. *Pearson correlation matrix,* where 0 = *mpg,* 1 = *cylinders,* 2 = *displacement,* 3 = *horsepower,* 4 = *weight,* 5 = *acceleration,* 6 = *year,* 7 = *origin*

When examining scatterplots of *mpg* against each feature, the following observations can be made, see Fig. 2:

- *Cylinders*: Due to the narrow, discrete nature of the variable (3 to 8), it is hard to draw any conclusions since it is not possible to distinguish unique datapoints which coincide in each of the points. A maximum of 12 locations for a dataset containing over 300 points will not be helpful, as stated by the *pigeonhole principle.*
- *Displacement*: It seems that cars with a high mpg value correlate with low engine displacement. For displacement sizes over approximately 350, no car with a high mpg is to be found. This aligns with intuition and the correlation matrix.
- *Horsepower*: It seems that cars with a high mpg number correlate with a low horsepower figure. For engines with a horsepower figure over approximately 125, no car with a high mpg is to be found. This aligns with intuition and the correlation matrix.
- *Weight*: It seems that cars with a high mpg number correlate with a low weight. For cars with a weight over approximately 3 750, no car with a high mpg is to be found. This aligns with intuition and the correlation matrix.

When examining the remaining scatterplots, no obvious patterns or correlations could be observed. This matches the results from the *correlation matrix*, as stated above.

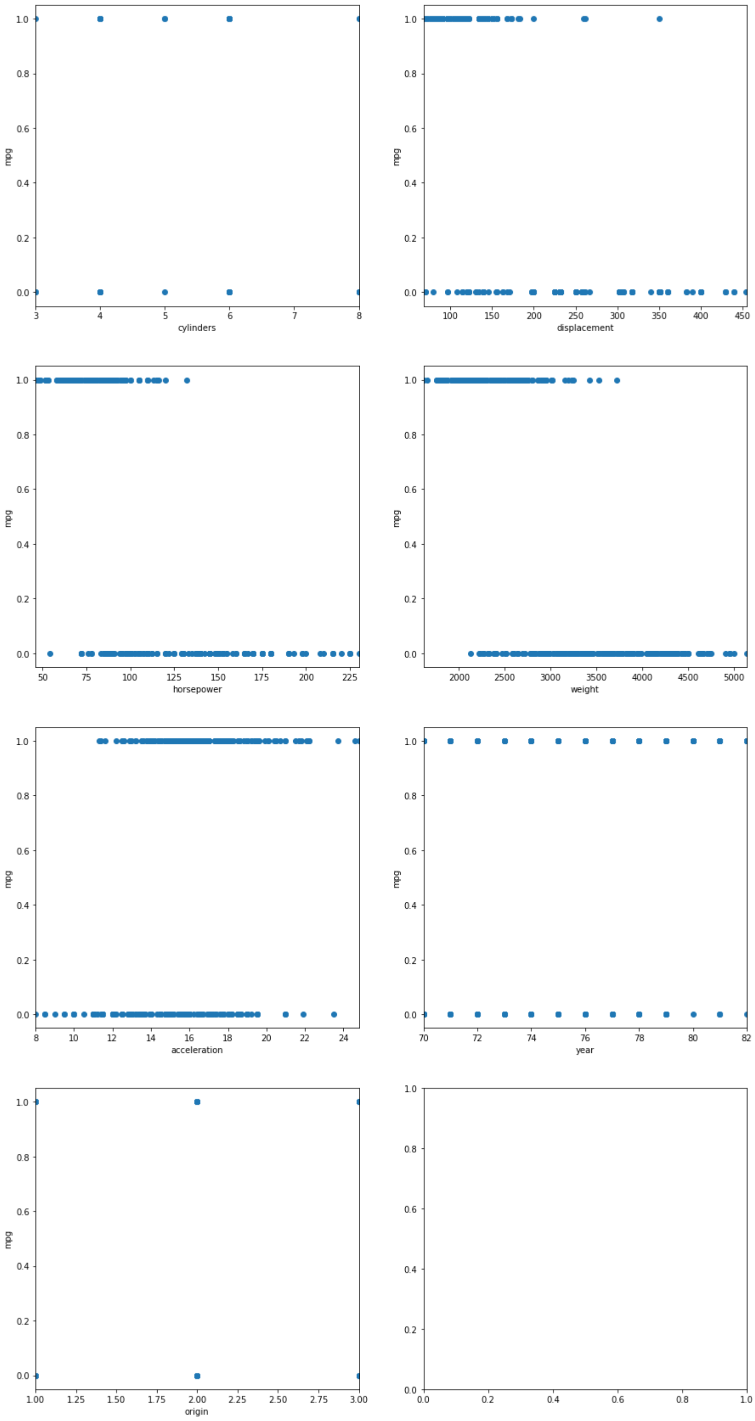


Fig. 2. *Scatterplots for mpg against each feature.*

While examining boxplots, clearly no correlation with mpg can be examined, since they are only examined independently. See Fig. 3. However, the usefulness of the horsepower data is lowered due to the presence of outliers. Secondly, spread in data is preferable to avoid high variance in the fitted model. Such spread can be observed for instance for variable *cylinders*.

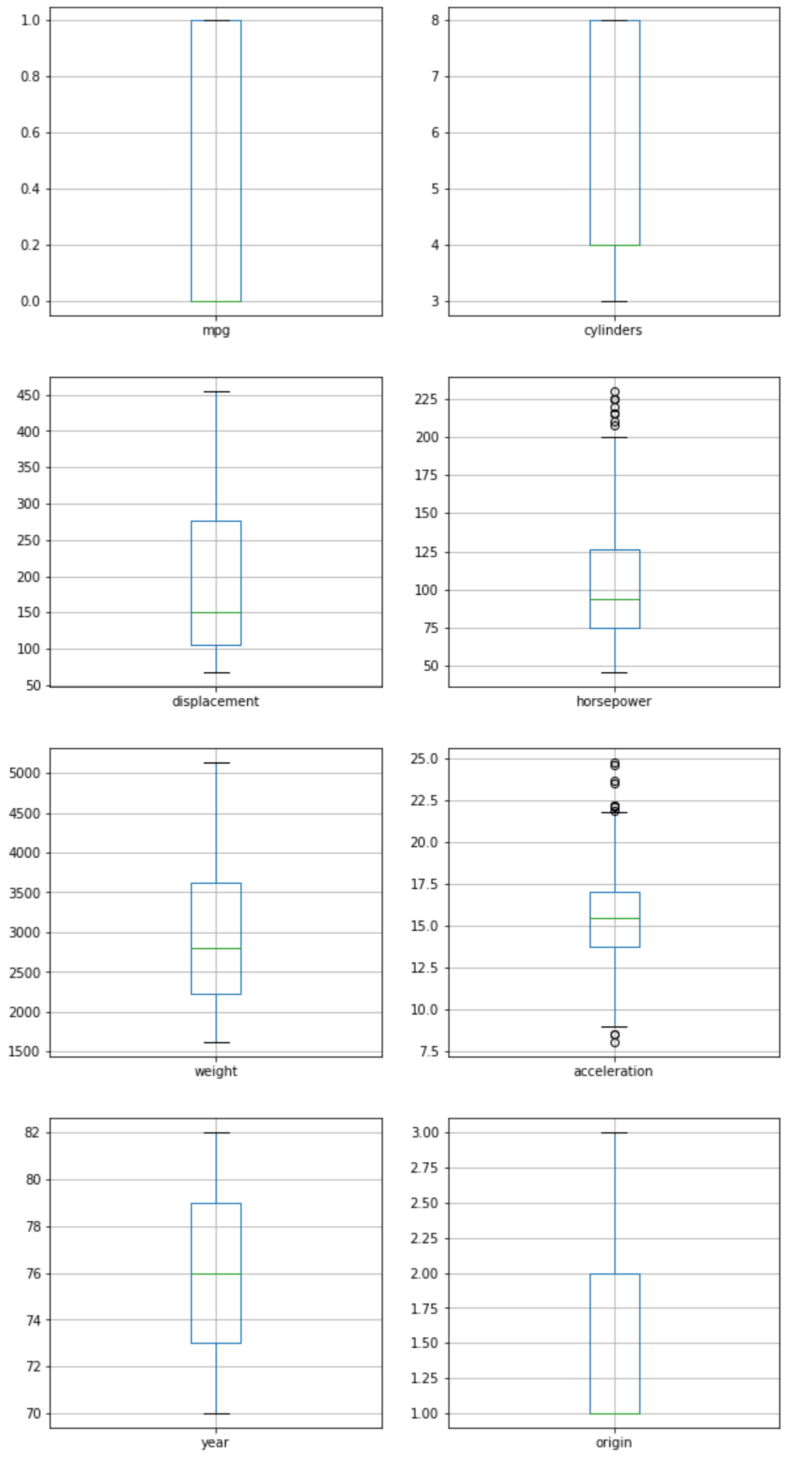


Fig. 3. *Boxplots for each feature.*

Conclusion: Variables were selected as *cylinders, weights, horsepower* and *displacement*, as per the motivation in exercises above.

c) The models for Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Logistic Regression and k-nearest neighbours were trained, tested and evaluated using *sci-kit learn* classifier functions: *LinearDiscriminantAnalysis(),* *QuadraticDiscriminantAnalysis(), LogisticRegression()* and *KNeighborsClassifier()*, respectively. The models were trained for the four features as selected in exercise b). For LDA and QDA, the *empirical distribution* was here used as prior. As a sidenote, the function for LDA uses *singular value decomposition* in the solver, which is recommended for small data sets.

For the training and test data split, a function was constructed such that the proportion between training and test data could be varied. Firstly, a proportion of 0.2 test data was set. The training and test errors could be obtained as $Error = 1 - Accuracy$, as in Tab. 1.

	LDA	QDA	Logistic Regression
Training Error	0.1086	0.1086	0.1470
Test Error	0.1139	0.1139	0.1519

Tab. 1. *Training and test errors for each model.*

As expected, the test error is consistently higher than the training error.

For k-nearest neighbours, KNN, a set of k-values from 1 to 30 were evaluated, with a test data ratio of 0.2. The results in terms of test and training error, respectively, can be seen from Fig. 4 below.

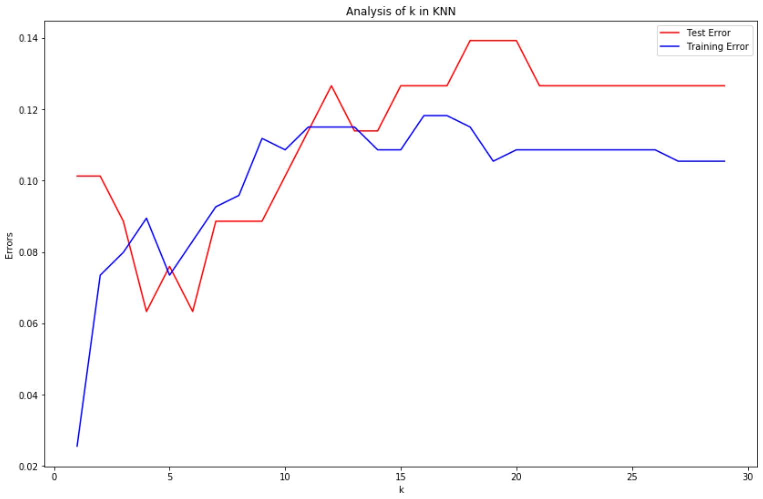


Fig. 4. *kNN test and training errors for different values on* k.

It could be observed that the training error was generally lower than the test error. It could also be seen that both the errors levelled off for $k \approx 22$, and the two errors were simultaneously minimized at $k \approx 5$.

Due to high flexibility for low k's, the training error is minimized, due to overfitting. On the contrary, the method is not sufficiently flexible for higher k's, explaining the higher test and training errors.

d) By using six different proportions of test data, regularly spread between 0.05 and 0.3, it was possible to analyse the differences that the split ratio made in evaluating the model. While keeping $k = 5$ constant for kNN, the results were obtained as can be seen in Fig. 5. The random seed in the split function was fixed, allowing for easier comparison between the different results for the same data set.

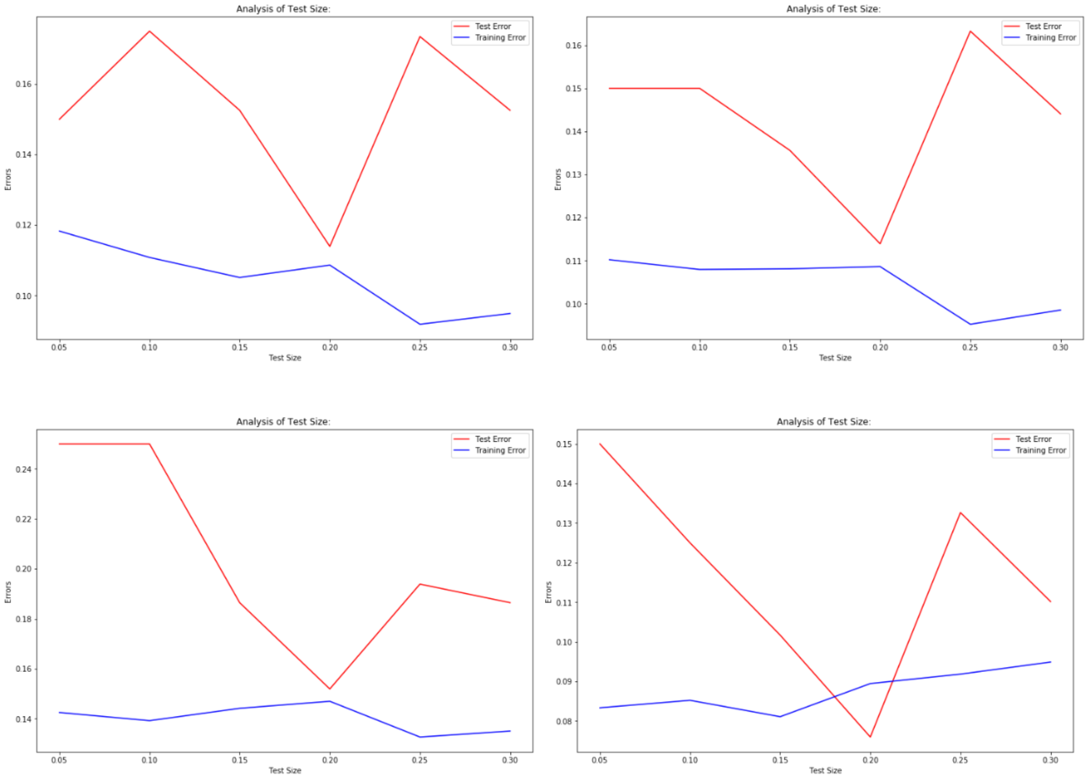


Fig. 5. *Test and training error for (from top left to right): LDA, QDA, logistic regression and kNN.*

It was observed that for all four cases, the test error varied greatly but reached a minimum around a ratio of 0.2, whereas the training error remained much more stable in each case. By making the ratio of test data significantly larger or smaller than 0.2, the test error generally increased in all cases. Particularly, the test error increased as the ratio became smaller. Moreover, the test size of approximately 0.2 matches what is most commonly used and recommended as test ratio and seems to be a suitable trade-off between training and test data sizes.

It shall be noted that the results for such a small dataset as in this case, with less than 400 data points in total, does have a high variance and depends on *which* fraction of the data the models are trained and tested on. Here, cross-validation serves a purpose to get more reliable results. However, this was omitted in the study.

Part B

a) Firstly, a set of samples from a Gaussian distribution were generated for $M = 300$ and $N = 500$ with test ratio 0.2. The following parameters were used in the first round: $\begin{bmatrix}X_1\\X_2\end{bmatrix} \in N\left(\begin{bmatrix}-2\\-2\end{bmatrix},\begin{bmatrix}1 & 0\\0 & 100\end{bmatrix}\right)$ and $\begin{bmatrix}\tilde{X}_1\\\tilde{X}_2\end{bmatrix} \in N\left(\begin{bmatrix}7\\8\end{bmatrix},\begin{bmatrix}4 & 0\\0 & 40\end{bmatrix}\right)$, yielding samples and errors as can be seen in Fig. 6 and Tab. 2, respectively.

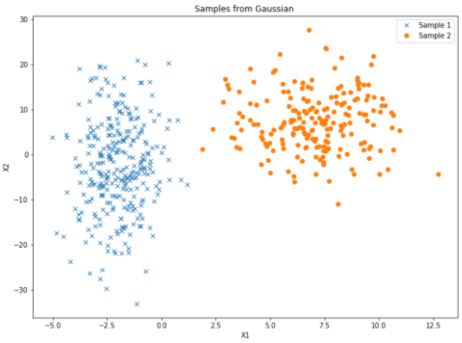


Fig. 6. *Samples from two multivariate Gaussian distributed random variables.*

	Test Error
QDA	0.00
LDA	0.01
Logistic Regression	0.00

Tab. 2. *Test error for classification of two Gaussian distributed random variables.*

Here, the parameters made the classification rather simple. With means for the two multivariate distributions spread far from each other and low variances along the X1-axis, it becomes clear from the plot how the two sets are linearly separable.

In the second round, the parameters were chosen as $\begin{bmatrix}X_1\\X_2\end{bmatrix} \in N\left(\begin{bmatrix}1\\1\end{bmatrix},\begin{bmatrix}1 & 0\\0 & 100\end{bmatrix}\right)$ and $\begin{bmatrix}\tilde{X}_1\\\tilde{X}_2\end{bmatrix} \in N\left(\begin{bmatrix}-2\\-2\end{bmatrix},\begin{bmatrix}4 & 0\\0 & 40\end{bmatrix}\right)$, yielding samples and errors as can be seen in Fig. 7 and Tab. 3, respectively.

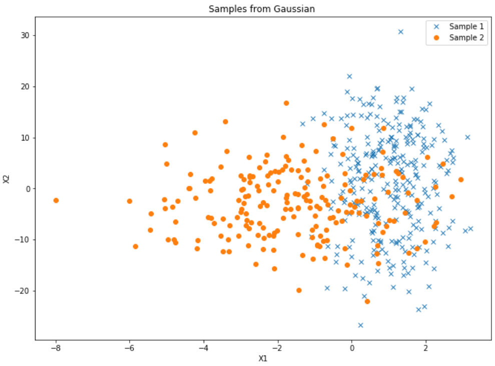


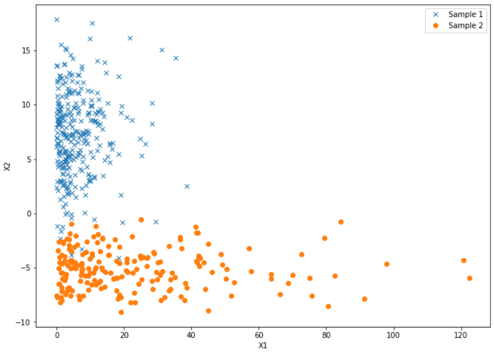
Fig. 7. *Samples from two multivariate Gaussian distributed random variables.*

	Test Error
QDA	0.14
LDA	0.14
Logistic Regression	0.14

Tab. 3. *Test error for classification of two Gaussian distributed random variables.*

In this round, the classification became much more difficult, and the two sets are clearly no longer linearly separable. Here, the three methods performed equally well and consistently worse than in the first scenario.
Generally, the larger the distance between classes, the easier classification gets.

b) In the second exercise, the first two components were drawn from a Gaussian distribution, whereas the second components were drawn from an exponential distribution. The remaining parameters were unchanged from exercise a).
$X_1 \in Exp\left(\frac{1}{6}\right), \tilde{X}_1 \in Exp\left(\frac{1}{20}\right)$ and $X_2 \in N(7,4), \tilde{X}_2 \in N(-5,2)$ (using the notation $X \in \mathrm{Exp}(\lambda)$ for $\lambda = \mathbb{E}[X]$), yielding samples and errors as can be seen in Fig. 8 and Tab. 4, respectively.



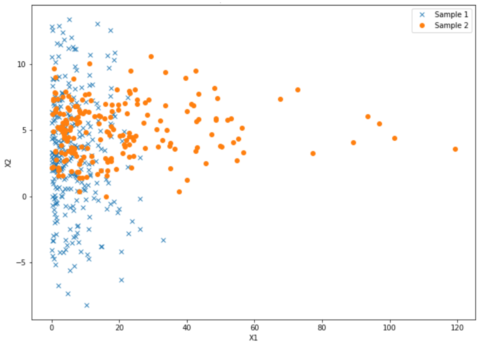
	Test Error
QDA	0.04
LDA	0.04
Linear Regression	0.04

Tab. 4. *Test error for classification of two Gaussian and exponentially distributed random variables.*

Fig. 8. *Samples from two Gaussian and exponentially distributed random variables.*

Here, the two datasets were not linearly separable, and the methods performed equally well.

In the following round, new parameters were selected as $X_1 \in Exp\left(\frac{1}{6}\right), \tilde{X}_1 \in Exp\left(\frac{1}{20}\right)$ and $X_2 \in N(3,4), \tilde{X}_2 \in N(5,2)$, yielding samples and errors as can be seen in Fig. 9 and Tab. 5, respectively.



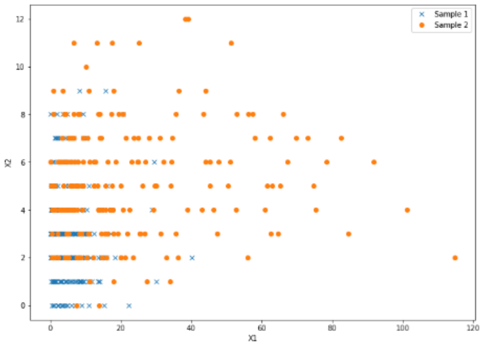
	Test Error
QDA	0.24
LDA	0.28
Linear Regression	0.27

Tab. 5. *Test error for classification of two Gaussian and exponentially distributed random variables.*

Fig. 9. *Samples from two Gaussian and exponentially distributed random variables.*

Here, it became clear how classification was very difficult, as the two datasets were far from linearly separable. The methods yielded approximately equal results in terms of test error.

c) In the third exercise, the first two components were drawn from an exponential distribution whereas the second two components were drawn from a Poisson distribution. The remaining parameters were still unchanged from exercise a). $X_1 \in Exp\left(\frac{1}{6}\right), \tilde{X}_1 \in Exp\left(\frac{1}{20}\right)$ and $X_2 \in Po(3), \tilde{X}_2 \in Po(5)$, yielding samples and errors as can be seen in Fig. 10 and Tab. 6, respectively.



	Test Error
QDA	0.23
LDA	0.26
Linear Regression	0.22

Tab. 6. *Test error for classification of two Poisson and exponentially distributed random variables.*

Fig. 10. *Samples from two Poisson and exponentially distributed random variables.*

In the following round, new parameters were selected as $X_1 \in Exp\left(\frac{1}{6}\right), \tilde{X}_1 \in Exp\left(\frac{1}{20}\right)$ and $X_2 \in Po(3), \tilde{X}_2 \in Po(5)$, yielded samples and errors as can be seen in Fig. 11 and Tab. 5, respectively.

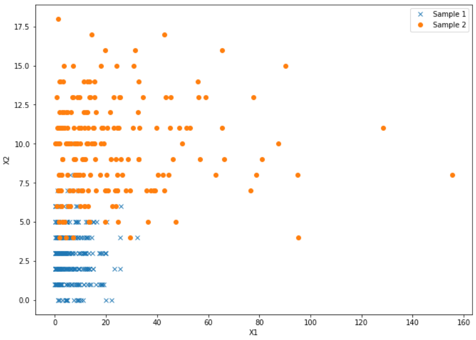


Fig. 11. *Samples from two Poisson and exponentially distributed random variables.*

	Test Error
QDA	0.23
LDA	0.17
Linear Regression	0.18

Tab. 5. *Test error for classification of two Poisson and exponentially distributed random variables.*

It shall be noted that both QDA and LDA assumes data sampled from a Gaussian distribution. This assumption is violated in exercises b) and c). However, the results yielded depends largely on the parameters chosen and the randomness in the sampling. The higher test errors in b) and c) may thus not entirely depend on this specific violation.

Part C

a)

I) For $p = 1$, we have that X ∈ U(0,1) on $\mathbb{R}$. It is clear that for x ∈ [0.05, 0.95], the observations to be used are on the interval [x - 0.05, x + 0.05], corresponding to a fraction of 0.1. For x ∈ [0, 0.05], however, observations on the interval [0, x + 0.05] would be used, corresponding to a fraction of (x + 0.05). In a similar fashion, for x ∈ [0.95, 1], observations corresponding to a fraction of (1.05 - x) would be used.

Hence, the average fraction to be used is

$$\int_{0.05}^{0.95} 0.1dx + \int_{0}^{0.05} (x + 0.05)dx + \int_{0.95}^{1} (1.05 - x)dx = 0.0975$$

This corresponds to a fraction of <u>9.75%</u> of the available points to be used, on average, for $p = 1$.

II) For $p = 2$, we have that $(X_1, X_2) \in U(0,1) \times U(0,1)$ on $\mathbb{R}^2$. If we now assume X_1 and X_2 to be independent, we have that the fraction of available points to be used is simply $0.0975^2 \approx 0.009506$.

Hence, a fraction of approximately <u>0.95%</u> of the available points are to be used, on average, for $p = 2$.

III) Using the same logic as in II), we have for $p = 100$ and independent $X_1, X_2, \ldots, X_{100}$, that the fraction of available points to be used is simply $0.0975^{100} \approx 8{\cdot}10^{-102}$, or particularly $\lim\limits_{p\to\infty} 0.0975^p = 0$.

As can be seen in exercises I) – III), the fraction of available points within a certain interval decreases with increased dimensionality, implying that points tend to be more spread apart in higher dimensions. This means that the k nearest neighbours may not be so "near" anymore, which breaks down the principle of kNN - as it is based on the assumption that nearby points behave similarly and are to be classified similarly. Classifying a test observation based upon its k nearest neighbours is therefore not as suitable in high dimensions. This phenomenon is known as the *curse of dimensionality.*

As for LDA and QDA, the same logic as for kNN is applied. LDA and QDA both utilises the assumption that nearby points are to be classified similarly. Instead of taking the k nearest neighbours, LDA and QDA separates clusters using a decision boundary. For higher dimensions, the distances between points within each cluster will increase. Hence, points on each side of the boundary are more spread out from each other and are thus not necessarily as "similar" as in lower dimensions.

b)

i) If the Bayes decision boundary is linear, we would expect QDA to perform better on the training set, but LDA to perform better on the test set. This is since QDA is non-linear and allows for higher flexibility and therefore can be better fitted to the training data. QDA does on the other hand, for the same reason, run the risk of overfitting the data, which is why LDA is expected to perform better on the test data.

(2) QDA will perform better on the test data when Bayes decision boundary is non-linear. This is because the Bayes decision boundary per definition minimizes the test error and is more flexible. QDA will most likely also perform better on the training data, with the same motivation.

3) As the sample size n increases, the performance of both QDA and LDA will improve, since more data allows for a more reliable, close to real world, fit. However, the performance of QDA relative to LDA will increase due to its flexibility. Additionally, more data will lower the influence of the higher variance property of QDA.