

CA02

February 22, 2023

1 CA02 - Training Perceptron and Adaline models

Make sure you: a) describe briefly what you intend to do using markdown cells; b) comment your code properly but briefly, such that the reader can easily understand what the code is doing.

1.1 Imports

```
[ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Import the relevant classes from adaline.py and perceptron.py in the
↳ classifiers folder

from classifiers.perceptron import Perceptron
from classifiers.adaline import AdalineGD
```

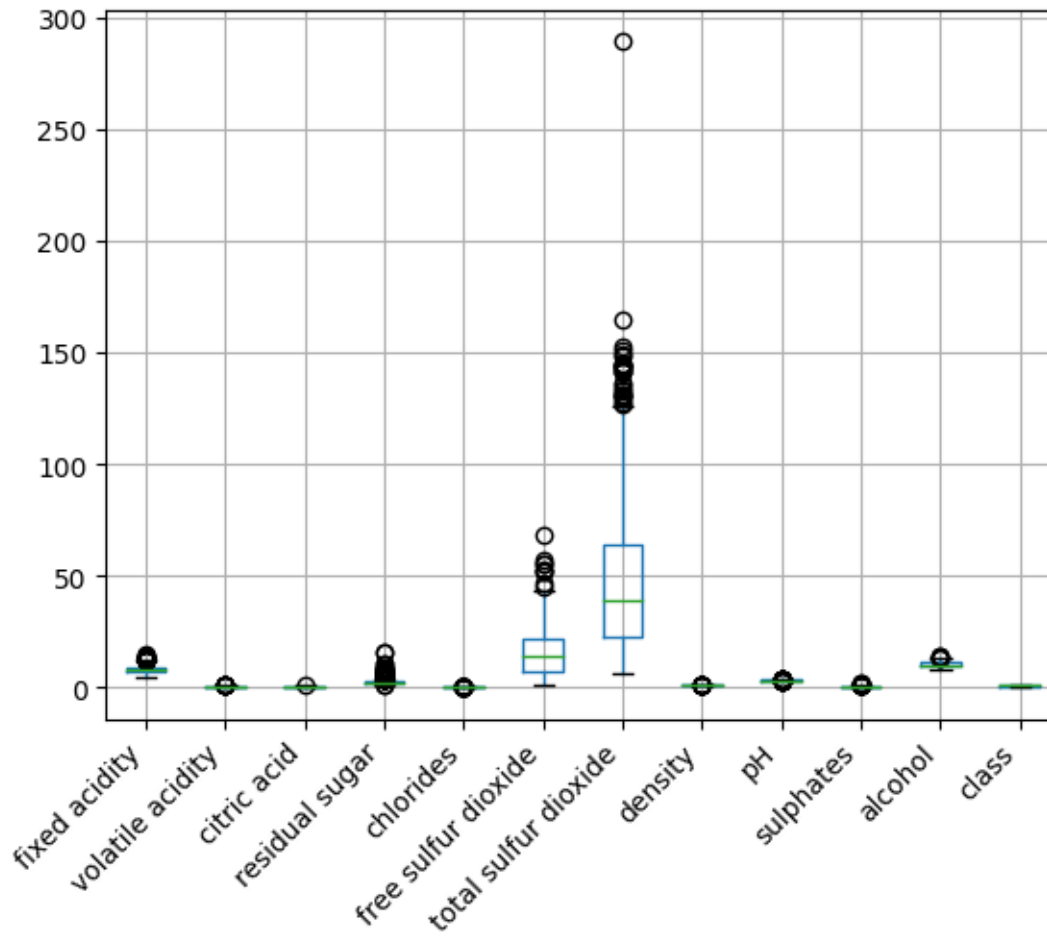
1.2 Loading and exploring data

Visualise the raw data with appropriate plots and inspect it for possible outliers or inconsistencies. Comment briefly on what you see and how this will impact the performance of the perceptron and adaline. For this use no more than three sentences.

```
[ ]: # Insert your code below
# =====

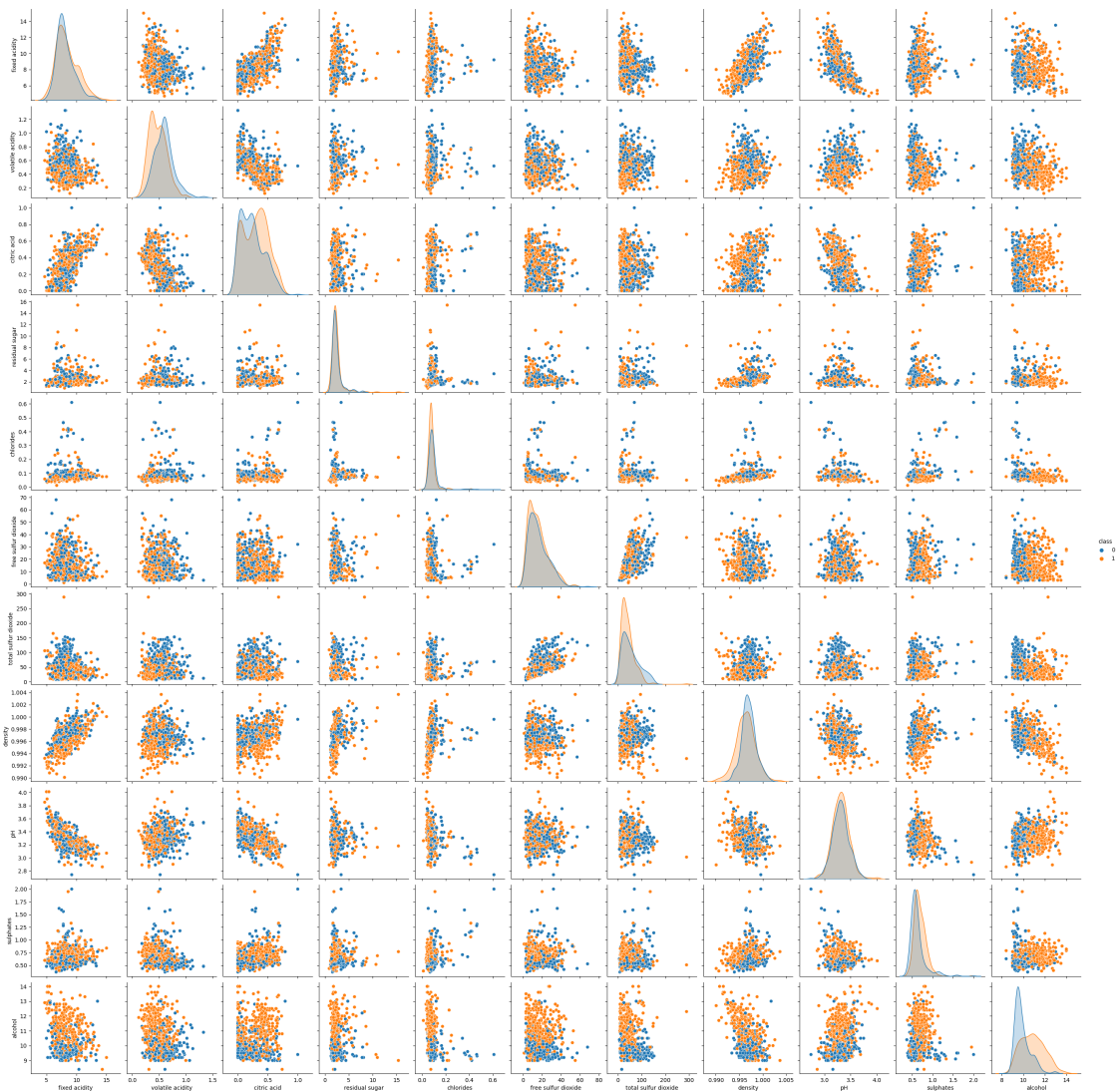
data = pd.read_csv('assets/wine.csv')

# A boxplot to see the features in one plot
data.boxplot()
plt.xticks(rotation=45, ha='right')
plt.show()
```

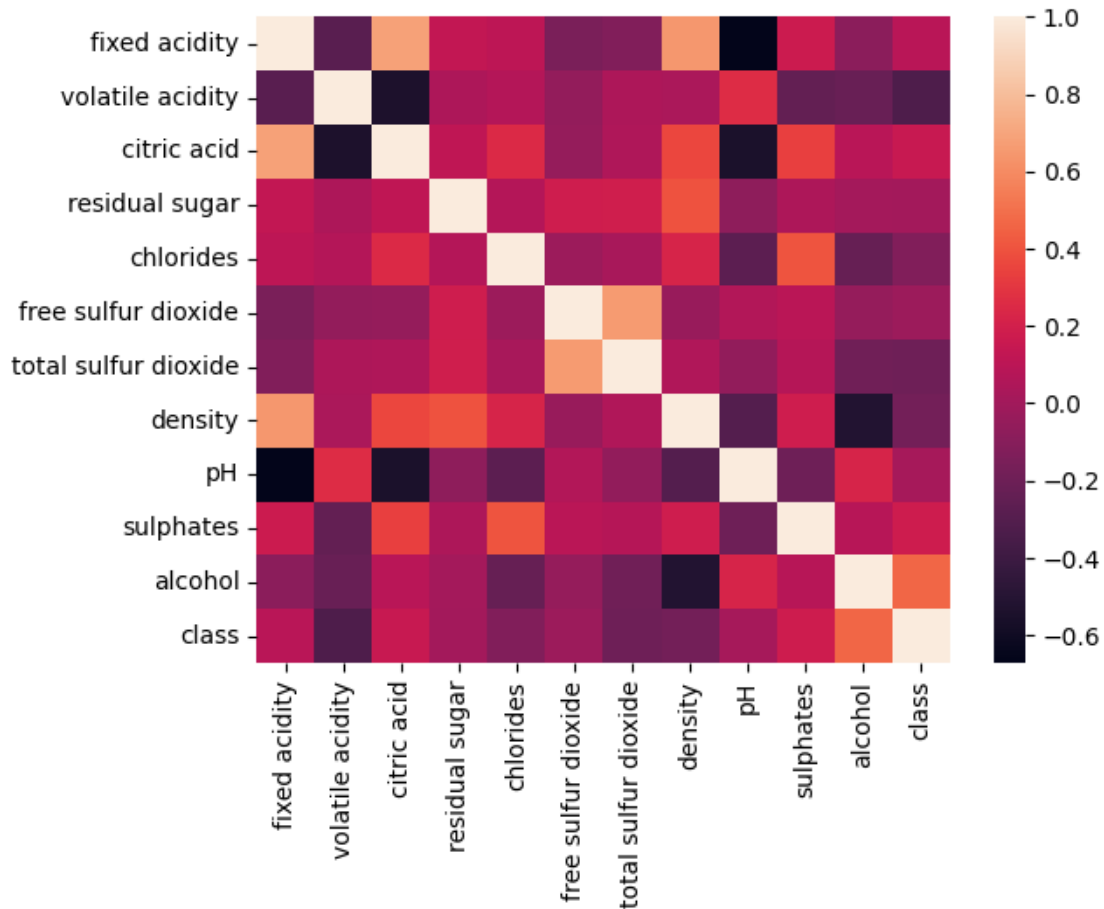


Free sulfur dioxide and total sulfur dioxide might have some outliers, but it's hard to tell before the scaling.

```
[ ]: # Plot for all features
sns.pairplot(data, hue='class')
plt.show()
```



```
[ ]: sns.heatmap(data.corr())
plt.show()
```



The correlation matrix reveals several important findings. For instance, we can note a strong negative correlation between pH levels and acidity measures, indicating that low pH levels correspond to high acidity and vice versa. Additionally, there is a significant negative correlation between density and alcohol content. Conversely, there is a positive correlation between fixed acidity and density.

1.3 Preprocessing data and training models

- Split the data into training and test_sets, where 400 samples are used for training
- Make sure the target values are appropriate for the Adaline and Perceptron classifiers

With each of the 400 models, you should predict the classes of the unseen samples in the test data and compute the test set classification accuracy. Store the results in a (8 x 50) numpy array or a pandas dataframe.

Preprocessing:

```
[ ]: # Insert your code below
# =====
```

```

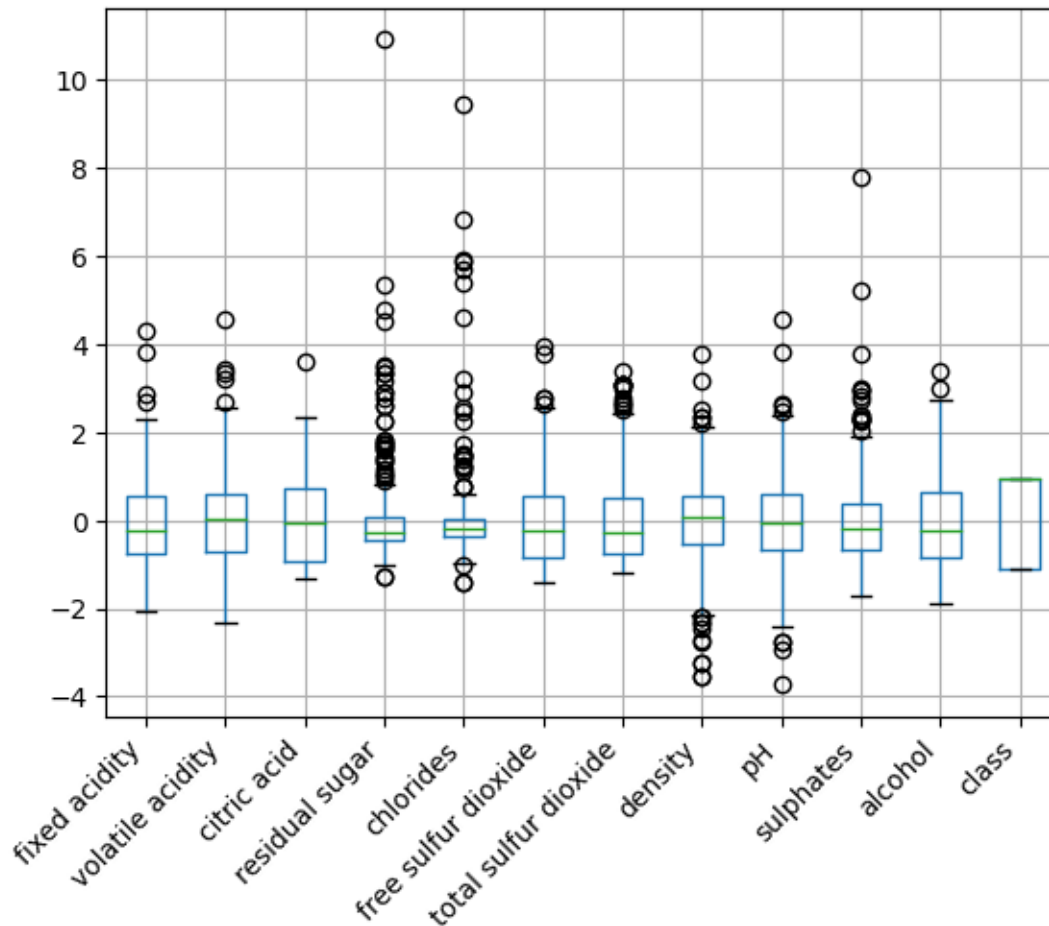
y = np.where(data['class'] == 1, 1, -1)

# Divide into test set and training set
X_train = data.iloc[:400]
X_test = data.iloc[400:]
y_train = y[:400]
y_test = y[400:]

# Scaling the data
X_train_sc = (X_train - np.mean(X_train, axis=0)) / np.std(X_train)
X_test_sc = (X_test - np.mean(X_train, axis=0)) / np.std(X_train)

# plot the boxplot to see th change the scaling did
X_train_sc.boxplot()
plt.xticks(rotation=45, ha='right')
plt.show()

```



Training:

```
[ ]: # Insert your code below
# =====

# Epochs from 1 to 50 and subsets of the training set from 50 to 400 in
↳ intervals of 50
epochs = [i for i in range(1,51)]
subsets = [50*i for i in range(1,9)]

# initialising a dataframe for th perceptron accuracy score
df_perceptron = pd.DataFrame(columns=subsets, index=epochs)

# train the data
for sub in subsets:
    acc = []
    for e in epochs:
        X_train_subset = X_train_sc.iloc[0:sub, 0:10].values

        ppn = Perceptron(eta=0.0001, n_iter=e)
        ppn.fit(X_train_subset, y_train[:sub])

        X_test_subset = X_test_sc.iloc[:, 0:10].values

        y_pred = ppn.predict(X_test_subset)
        acc = (y_pred == y_test).sum()/len(y_pred)
        df_perceptron[sub][e] = acc*100
```

```
[ ]: # Epochs from 1 to 50 and subsets of the training set from 50 to 400 in
↳ intervals of 50
epochs = [i for i in range(1,51)]
subsets = [50*i for i in range(1,9)]

# initialising a dataframe for th adaline accuracy score
df_adaline = pd.DataFrame(columns=subsets, index = epochs)

# train the data
for sub in subsets:
    for e in epochs:
        X_train_subset = X_train_sc.iloc[0:sub, 0:10].values

        ada1 = AdalineGD(n_iter=e, eta=0.0001)
        ada1.fit(X_train_subset, y_train[:sub])

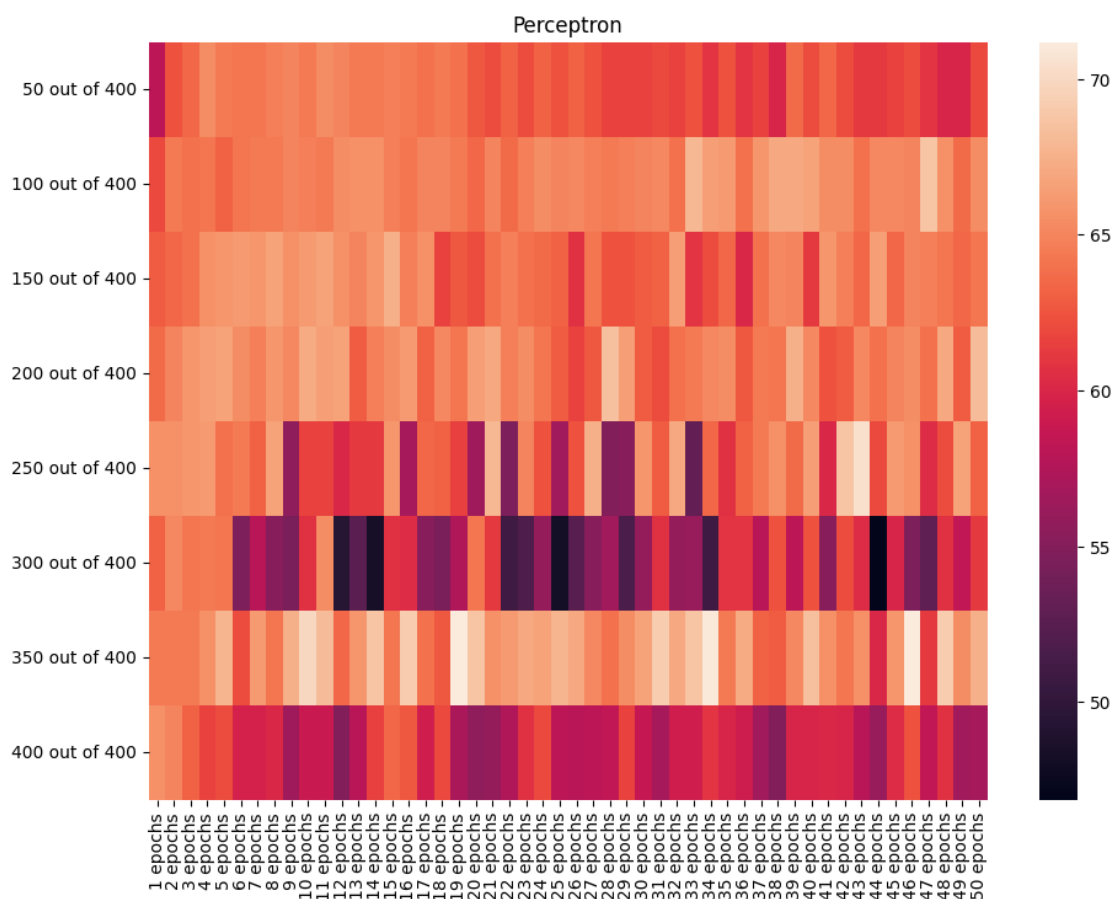
        X_test_subset = X_test_sc.iloc[:, 0:10].values

        y_pred = ada1.predict(X_test_subset)
        acc = (y_pred == y_test).sum()/len(y_pred)
        df_adaline[sub][e] = abs(acc)*100
```

1.4 Visualizing results

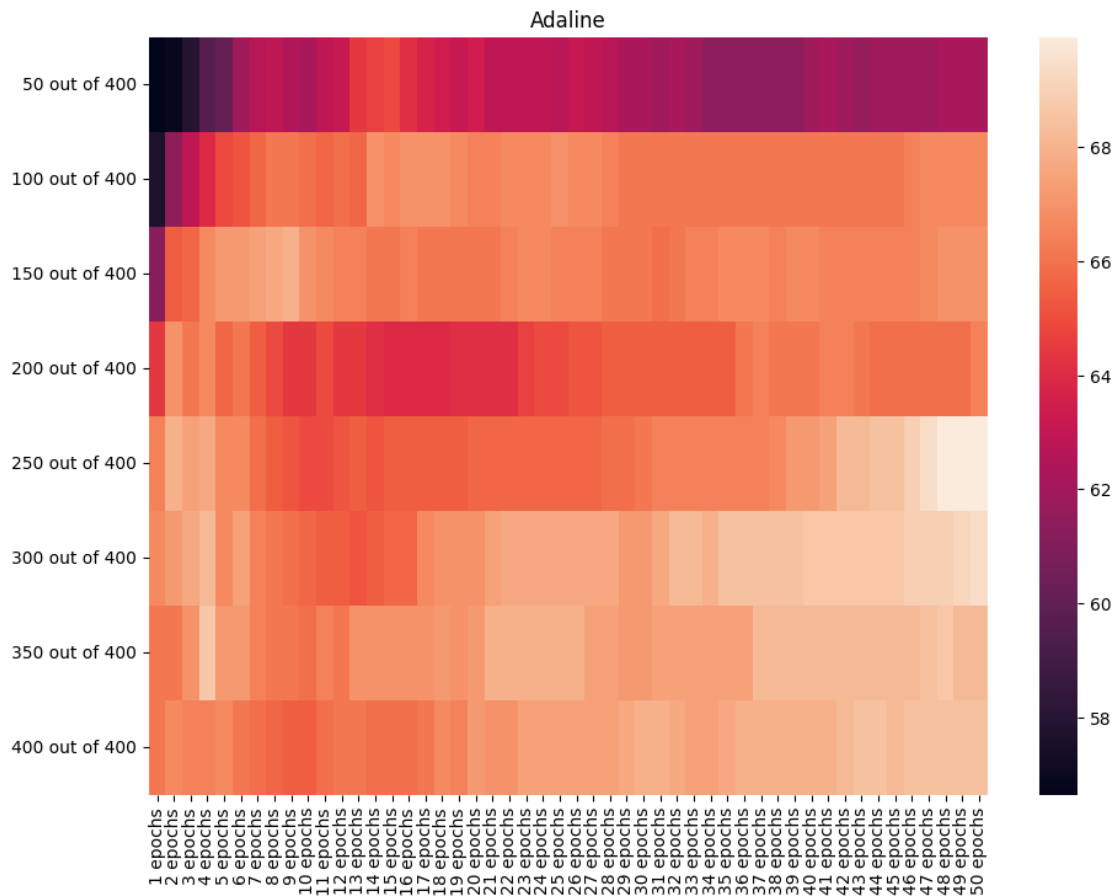
Plot a heatmap of the results (test set classification accuracy) using Python plotting packages matplotlib or seaborn (Lenker til en ekstern side.). See below what the heatmaps should look like for the two classification algorithms.

```
[ ]: y_label = [f"{sub} out of 400" for sub in subsets]
x_label = [f"{e} epochs" for e in epochs]
fig = plt.subplots(figsize=(11,8))
plt.title('Perceptron')
df_perceptron = df_perceptron.infer_objects()
sns.heatmap(df_perceptron.transpose(), xticklabels=x_label, yticklabels=y_label)
plt.show()
```



```
[ ]: y_label = [f"{sub} out of 400" for sub in subsets]
x_label = [f"{e} epochs" for e in epochs]
fig = plt.subplots(figsize=(11,8))
plt.title('Adaline')
df_adaline = df_adaline.infer_objects()
```

```
sns.heatmap(df_adaline.transpose(), xticklabels=x_label, yticklabels=y_label)
plt.show()
```



1.5 Metrics

Provide the maximum test set classification accuracy for each, the perceptron classifier and the adaline classifier and information on with which combination of number training data samples and number of epochs the best classification accuracy was achieved.

```
[ ]: perceptron_max_accuracy = df_perceptron.max().max()
print(f'The max accuracy score for Perceptron is {perceptron_max_accuracy}')
adaline_max_accuracy = df_adaline.max().max()
print(f'The max accuracy score for Adaline is {adaline_max_accuracy}')

print()

# Finding the location of the highest accuracy
max_acc_ind_perceptron = df_perceptron[df_perceptron.
    ↪isin([perceptron_max_accuracy])==True].last_valid_index()
```



```

max_acc_col_perceptron = df_perceptron.max()[df_perceptron.max() ==
    ↳perceptron_max_accuracy].index[0]

max_acc_ind_adaline = df_adaline[df_adaline.isin([adaline_max_accuracy])==True].
    ↳last_valid_index()
max_acc_col_adaline = df_adaline.max()[df_adaline.max() ==
    ↳adaline_max_accuracy].index[0]

print(f'The highest accuracy for Perceptron occurred at {max_acc_col_perceptron}
    ↳rows and {max_acc_ind_perceptron} epochs. ')
print(f'The highest accuracy for Adaline occurred at {max_acc_col_adaline} rows
    ↳and {max_acc_ind_adaline} epochs. ')

```

The max accuracy score for Perceptron is 71.17794486215539

The max accuracy score for Adaline is 69.92481203007519

The highest accuracy for Perceptron occurred at 350 rows and 46 epochs.

The highest accuracy for Adaline occurred at 250 rows and 50 epochs.

1.6 Discussion

The training time of the simpler perceptron algorithm is quite a bit longer than the training time of the adaline algorithm. What might be the reason for this?

The perceptron updates the weight everytime a sample is misclassified, while the adaline only updates one time.