

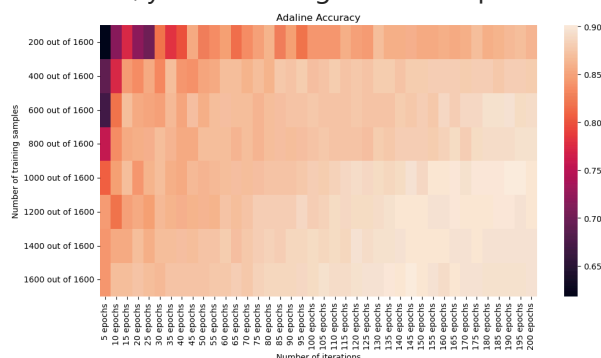
CA2 - Supervised machine learning classification pipeline - applied to medical data

Important information

- Do **not** use scikit-learn (`sklearn`) or any other high-level machine learning library for this CA
- Explain your code and reasoning in markdown cells or code comments
- Label all graphs and charts if applicable
- If you use code from the internet, make sure to reference it and explain it in your own words
- If you use additional function arguments, make sure to explain them in your own words
- Use the classes `Perceptron` , `Adaline` and `Logistic Regression` from the library `mlxtend` as classifiers (`from mlxtend.classifier import Perceptron, Adaline, LogisticRegression`). Always use the argument `minibatches=1` when instantiating an `Adaline` or `LogisticRegression` object. This makes the model use the gradient descent algorithm for training. Always use the `random_seed=42` argument when instantiating the classifiers. This will make your results reproducible.
- You can use any plotting library you want (e.g. `matplotlib` , `seaborn` , `plotly` , etc.)
- Use explanatory variable names (e.g. `X_train` and `X_train_scaled` for the training data before and after scaling, respectively)
- The dataset is provided in the file `fetal_health.csv` in the `assets` folder

Additional clues

- Use the `pandas` library for initial data inspection and preprocessing
- Before training the classifiers, convert the data to raw `numpy` arrays
- For Part IV, you are aiming to create a plot that looks similar to this:



Additional information

- Feel free to create additional code or markdown cells if you think it will help you explain your reasoning or structure your code (you don't have to).

Part I: Data loading and data exploration

Import necessary libraries/modules:

```
In [16]: # Insert your code below
# =====
from mlxtend.classifier import Perceptron, LogisticRegression, Adaline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading and exploring data

1. Load the dataset `fetal_health.csv` with `pandas` . Use the first column as the row index.
2. Check for missing data, report on your finding and remove samples with missing data, if you find any.
3. Display the raw data with appropriate plots/outputs and inspect it. Describe the distributions of the values of feature `"baseline value"` , `"accelerations"` , and the target variable `"fetal_health"` .
4. Will it be beneficial to scale the data? Why or why not?
5. Is the data linearly separable using a combination of any two pairs of features? Can we expect an accuracy close to 100% from a linear classifier?

```
In [17]: # Insert your code below
# =====
# 1
df = pd.read_csv("./assets/fetal_health.csv", index_col=0)

# 2
missing_value = df.isnull().sum().sum()
print(f"Total missing data: {missing_value}")

# 3
df.plot(x="baseline value", y="fetal_health", kind="scatter")
df.plot(x="accelerations", y="fetal_health", kind="scatter")
df.plot(x="baseline value", y="accelerations", kind="scatter")

fig, axes = plt.subplots(1, 3, figsize=(18, 5))

sns.histplot(df["baseline value"], ax=axes[0], color='pink')
axes[0].set_title("Distribution of Baseline Value")

sns.histplot(df["accelerations"], ax=axes[1], color='hotpink')
axes[1].set_title("Distribution of Accelerations")

sns.countplot(x=df["fetal_health"], ax=axes[2], palette=["lime", "red"])
axes[2].set_title("Distribution of Fetal Health Categories")
```

```
plt.show()
```

"""
We used scatter plot for both baseline value and accelerations with fetal_health
This is because we wanted a plot that could help us visualize the points and the
We also used subplots to count the features. This gives us a clearer vision of t
"""

4
"""
If we plan to use logistic regression or similar algorithms, they would perform better
scaled the accelerations and the baseline values to be more similar in scale.
"""

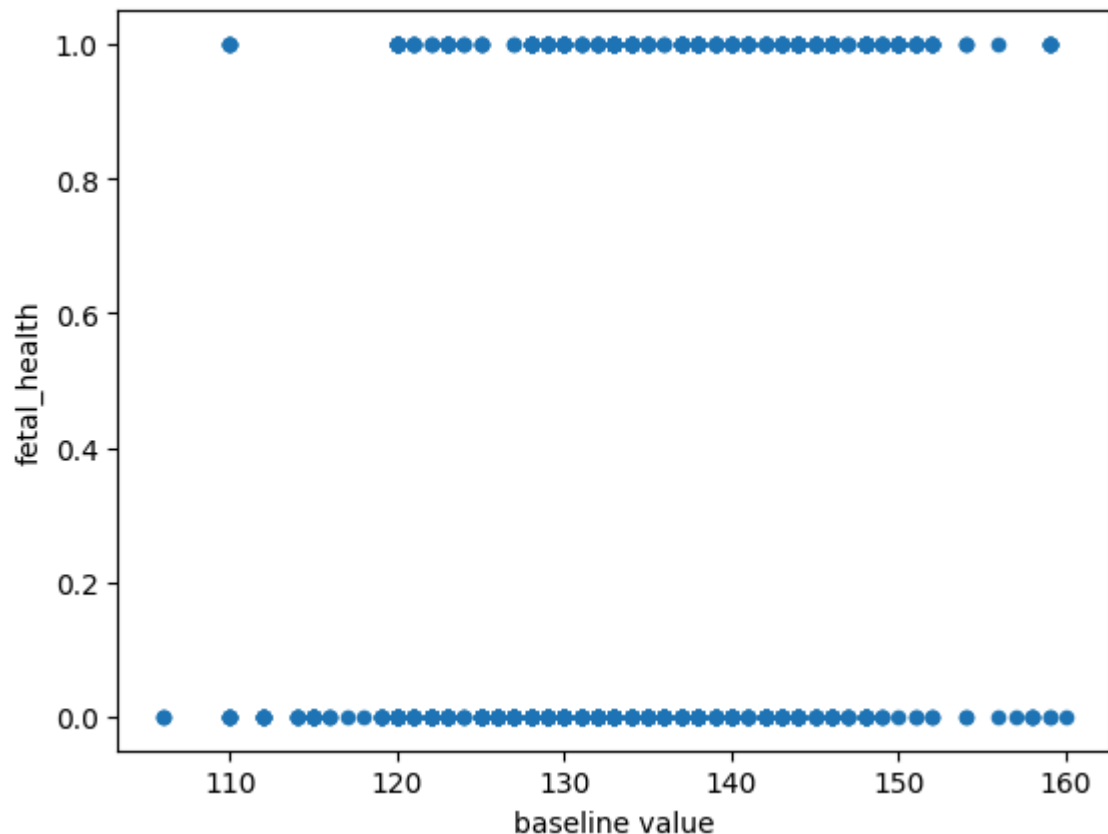
5
"""
It seems that the data is not linearly separable as we can see from the plots.
For example, we can not separate the baseline value and the accelerations with a
Therefore we can not expect anything close to 100% accuracy from a linear classifier.
"""

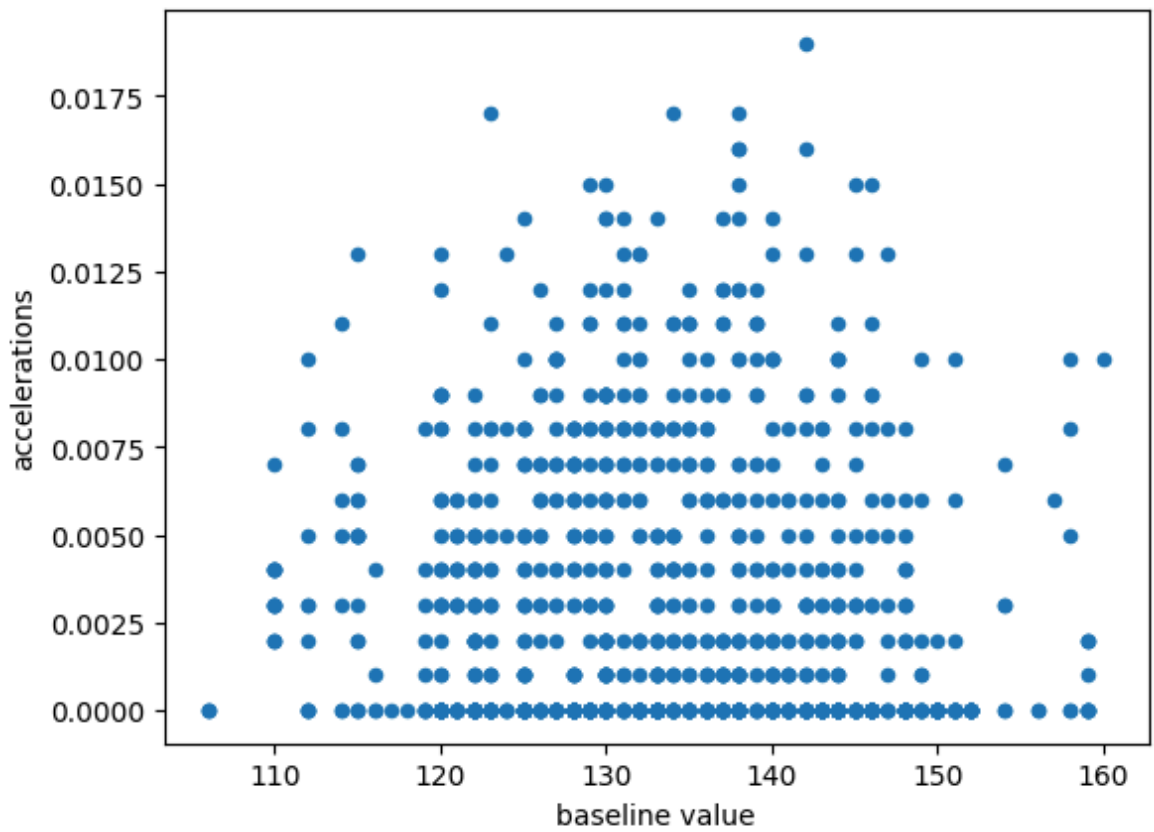
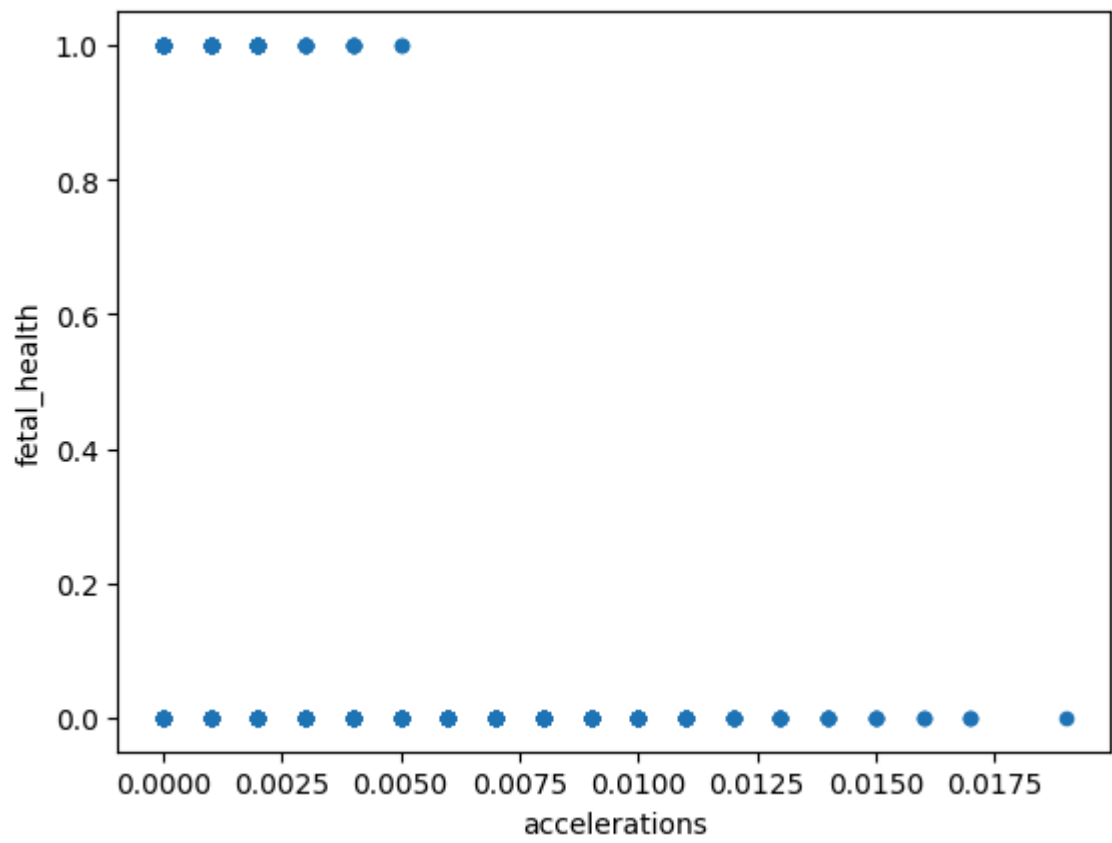
Total missing data: 0

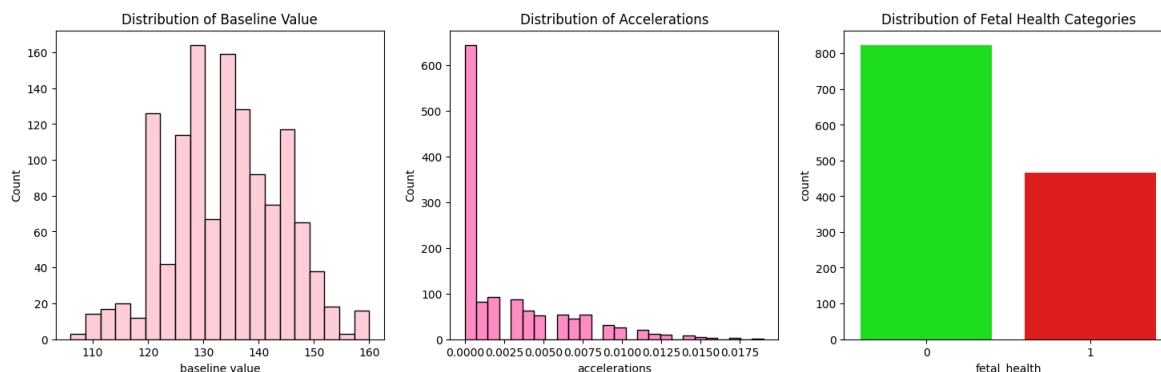
C:\Users\milad\AppData\Local\Temp\ipykernel_16760\2517102844.py:23: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(x=df["fetal_health"], ax=axes[2], palette=["lime", "red"])
```







Out[17]: '\nIt seems that the data is not linearly seperable as we can see from the plot s. \nFor example, we can not seperate the baseline value and the accelerations with a line. \nTherefore we can not expect anything close to 100% accuracy from a linear classifier.\n'

Part II: Train/Test Split

Divide your dataset into training and testing subsets. Follow these steps to create the split:

1. Divide the dataset into two data sets, each data set only contains samples of either class 0 or class 1:

- Create a DataFrame `df_0` containing all data with `"fetal_health"` equal to 0.
- Create a DataFrame `df_1` containing all data with `"fetal_health"` equal to 1.

2. Split into training and test set by randomly sampling entries from the data frames:

- Create a DataFrame `df_0_train` containing by sampling 75% of the entries from `df_0` (use the `sample` method of the data frame, fix the `random_state` to 42).
- Create a DataFrame `df_1_train` using the same approach with `df_1`.
- Create a DataFrame `df_0_test` containing the remaining entries of `df_0` (use `df_0.drop(df_0_train.index)` to drop all entries except the previously extracted ones).
- Create a DataFrame `df_1_test` using the same approach with `df_1`.

3. Merge the datasets split by classes back together:

- Create a DataFrame `df_train` containing all entries from `df_0_train` and `df_1_train`. (Hint: use the `concat` method you know from CA1)
- Create a DataFrame `df_test` containing all entries from the two test sets.

4. Create the following data frames from these splits:

- `X_train` : Contains all columns of `df_train` except for the target feature `"fetal_health"`
- `X_test` : Contains all columns of `df_test` except for the target feature `"fetal_health"`

- `y_train` : Contains only the target feature `"fetal_health"` for all samples in the training set
- `y_test` : Contains only the target feature `"fetal_health"` for all samples in the test set

5. Check that your sets have the expected sizes/shape by printing number of rows and columns ("shape") of the data sets.

- (Sanity check: there should be 8 features, almost 1000 samples in the training set and slightly more than 300 samples in the test set.)

6. Explain the purpose of this slightly complicated procedure. Why did we first split into the two classes? Why did we then split into a training and a testing set?

7. What is the share (in percent) of samples with class 0 label in test and training set, and in the initial data set?

In [18]: *# Insert your code below*

=====

1

```
df_0 = df[df["fetal_health"] == 0]
```

```
df_1 = df[df["fetal_health"] == 1]
```

2

```
df_0_train = df_0.sample(frac = 0.75, random_state = 42)
```

```
df_1_train = df_1.sample(frac = 0.75, random_state = 42)
```

```
df_0_test = df_0.drop(df_0_train.index)
```

```
df_1_test = df_1.drop(df_1_train.index)
```

3

```
df_train = pd.concat([df_0_train, df_1_train])
```

```
df_test = pd.concat([df_0_test, df_1_test])
```

4

```
X_train = df_train.drop(columns = ["fetal_health"])
```

```
X_test = df_test.drop(columns = ["fetal_health"])
```

```
y_train = df_train["fetal_health"]
```

```
y_test = df_test["fetal_health"]
```

5

```
print(X_train.shape)
```

```
print(X_test.shape)
```

6

"""

First we splittet the dataset by their fetal_health value. This is so we have an and we're also making sure the sets wont loose their target value.

We are splitting them again in test and train data to have two sets to work with and for when we are testing the model.

Lastly, we put the sets together into a full training and a full testing set to the features and target value sets.

This way we will make sure the dataset we are working with is balanced and that

"""

```
# 7
full_class_0_pct = 100 * (len(df_0) / len(df))
train_class_0_pct = 100 * (len(df_0_train) / len(df_train))
test_class_0_pct = 100 * (len(df_0_test) / len(df_test))

print(f"Class 0 proportion in full dataset: {full_class_0_pct}%")
print(f"Class 0 proportion in training set: {train_class_0_pct}%")
print(f"Class 0 proportion in test set: {test_class_0_pct}%")
```

```
(967, 8)
```

```
(323, 8)
```

```
Class 0 proportion in full dataset: 63.7984496124031%
```

```
Class 0 proportion in training set: 63.805584281282314%
```

```
Class 0 proportion in test set: 63.77708978328174%
```

Convert data to numpy arrays and shuffle the training data

Many machine learning models (including those you will work with later in the assignment) will not accept DataFrames as input. Instead, they will only work if you pass numpy arrays containing the data. Here, we convert the DataFrames `X_train`, `X_test`, `y_train`, and `y_test` to numpy arrays `X_train`, `X_test`, `y_train`, and `y_test`.

Moreover we shuffle the training data. This is important because the training data is currently ordered by class. In Part IV, we use the first n samples from the training set to train the classifiers. If we did not shuffle the data, the classifiers would only be trained on samples of class 0.

Nothing to be done here, just execute the cell.

```
In [19]: # convert to numpy arrays
X_train = X_train.to_numpy()
X_test = X_test.to_numpy()
y_train = y_train.to_numpy()
y_test = y_test.to_numpy()

# shuffle training data
np.random.seed(42) # for reproducibility
shuffle_index = np.random.permutation(len(X_train)) # generate random indices
X_train, y_train = X_train[shuffle_index], y_train[shuffle_index] # shuffle data
```

Part III: Scaling the data

1. Standardize the training *and* test data so that each feature has a mean of 0 and a standard deviation of 1.
2. Check that the scaling was successful
 - by printing the mean and standard deviation of each feature in the scaled training set
 - by putting the scaled training set into a DataFrame and make a violin plot of the data

Hint: use the `axis` argument to calculate mean and standard deviation column-wise.

Important: Avoid data leakage!

More hints:

1. For each column, subtract the mean (μ) of each column from each value in the column
2. Divide the result by the standard deviation (σ) of the column

(You saw how to do both operations in the lecture. If you don't remember, you can look it up in Canvas files.)

Mathematically (in case this is useful for you), this transformation can be represented for each column as follows:

$$X_{\text{scaled}} = \frac{(X - \mu)}{\sigma}$$

where:

- (X_{scaled}) are the new, transformed column values (a column-vector)
- (X) is the original values
- (μ) is the mean of the column
- (σ) is the standard deviation of the column

```
In [20]: # Insert your code below
# =====
# 1
X_train_mean = X_train.mean(axis = 0)
X_train_std = X_train.std(axis = 0)

X_train_scaled = (X_train - X_train_mean) / X_train_std
X_test_scaled = (X_test - X_train_mean) / X_train_std

# 2
print("Mean of each feature in scaled training set:\n", X_train_scaled.mean(axis=0))
print("Standard deviation of each feature in scaled training set:\n", X_train_scaled.std(axis=0))

df_train_scaled = pd.DataFrame(X_train_scaled, columns = df.columns[:-1])

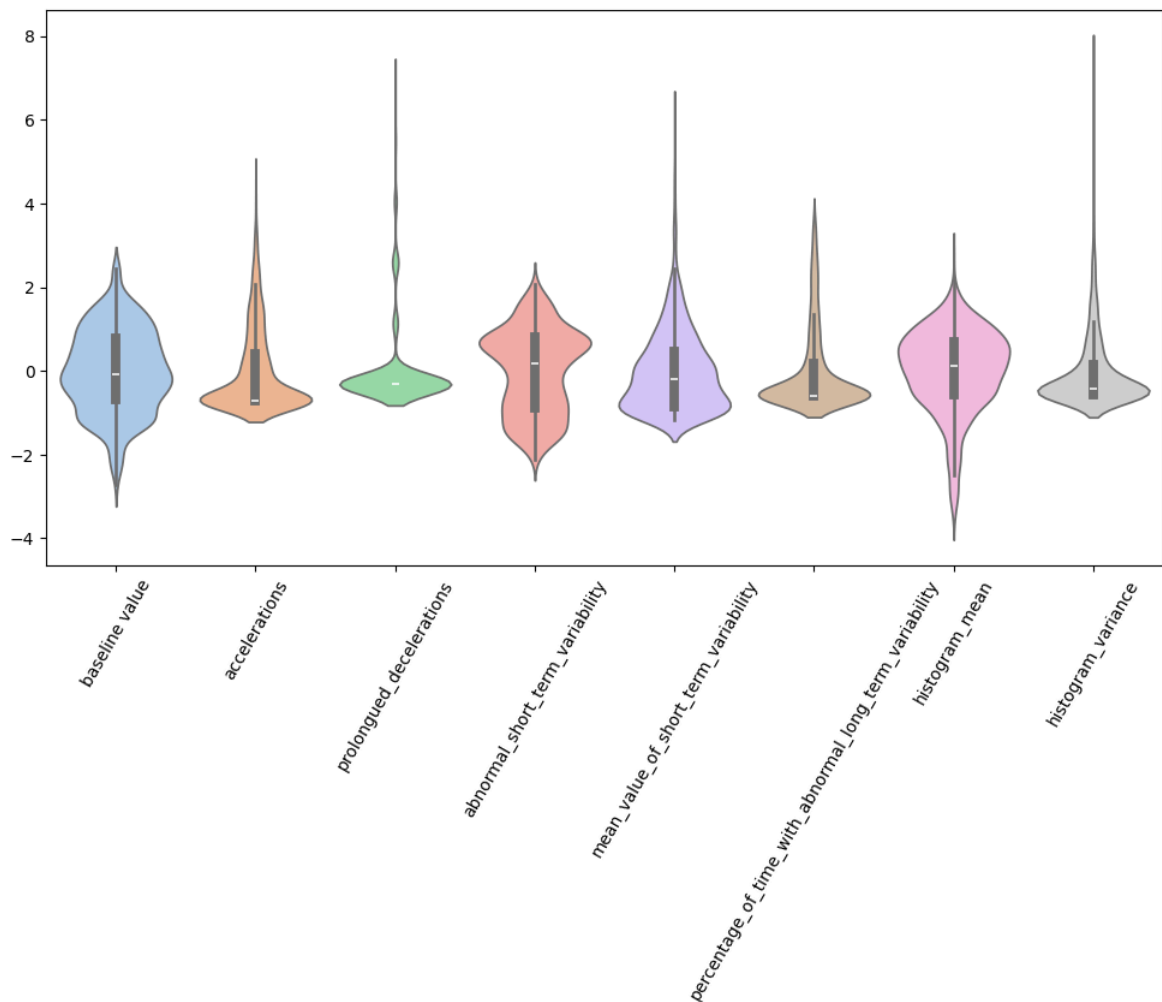
plt.figure(figsize = (12, 6))
sns.violinplot(data = df_train_scaled, palette = "pastel")
plt.xticks(rotation=60)
plt.show()
```

Mean of each feature in scaled training set:

```
[-1.31803106e-16  4.56925087e-15 -2.96097744e-16  1.33869705e-16
 -2.12543989e-17 -2.86453614e-16 -2.93342278e-16 -7.18717284e-17]
```

Standard deviation of each feature in scaled training set:

```
[1. 1. 1. 1. 1. 1. 1. 1.]
```

Part IV: Training and evaluation with different dataset sizes and training times

Often, a larger dataset size will yield better model performance. (As we will learn later, this usually prevents overfitting and increases the generalization capability of the trained model.) However, collecting data is usually rather expensive.

In this part of the exercise, you will investigate

- how the model performance changes with varying dataset size
- how the model performance changes with varying numbers of epochs/iterations of the optimizer/solver (increasing training time).

For this task (Part IV), use the `Adaline`, `Perceptron`, and `LogisticRegression` classifier from the `mlxtend` library. All use the gradient descent (GD) algorithm for training.

Important: Use a learning rate of `1e-4` (`0.0001`) for all classifiers, and use the argument `minibatches=1` when initializing `Adaline` and `LogisticRegression` classifier (this will make sure it uses GD). For all three classifiers, pass `random_seed=42` when initializing the classifier to ensure reproducibility of the results.

Model training

Train the model models using progressively larger subsets of your dataset, specifically: first 50 rows, first 100 rows, first 150 rows, ..., first 650 rows, first 700 rows (in total 14 different variants).

For each number of rows train the model with progressively larger number of epochs: 2, 7, 12, 17, ..., 87, 92, 97 (in total 20 different model variants).

The resulting $14 \times 20 = 280$ models obtained from the different combinations of subsets and number of epochs. An output of the training process could look like this:

```
Model (1) Train a model with first 50 rows of data for 2 epochs
Model (2) Train a model with first 50 rows of data for 7 epochs
Model (3) Train a model with first 50 rows of data for 12 epochs
...
Model (21) Train a model with first 100 rows of data for 2
epochs
Model (22) Train a model with first 100 rows of data for 7
epochs
...
Model (279) Train a model with first 700 rows of data for 92
epochs
Model (280) Train a model with first 700 rows of data for 97
epochs
```

Model evaluation

For each of the 280 models, calculate the **accuracy on the test set** (do **not** use the score method but compute accuracy yourself). Store the results in the provided 2D numpy array (it has 14 rows and 20 columns). The rows of the array correspond to the different dataset sizes, and the columns correspond to the different numbers of epochs.

Tasks

1. Train the 280 Adaline classifiers as mentioned above and calculate the accuracy for each of the 280 variants.
2. Generalize your code so that is doing the same procedure for all three classifiers: `Perceptron`, `Adaline`, and `LogisticRegression` after each other. Store the result for all classifiers. You can for example use an array of shape $3 \times 14 \times 20$ to store the accuracies of the three classifiers.

Note that executing the cells will take some time (but on most systems it should not be more than 5 minutes).

```
In [30]: # Train and evaluate all model variants
# Insert your code below
# =====
dataset_sizes = np.arange(50, 751, 50)
epochs_list = np.arange(2, 102, 5)

accuracies = np.zeros((3, len(dataset_sizes), len(epochs_list)))
```

```
classifiers = {  
    "Adaline": Adaline(eta = 1e-4, minibatches = 1, random_seed = 42),  
    "Perceptron": Perceptron(eta = 1e-4, random_seed = 42),  
    "LogisticRegression": LogisticRegression(eta = 1e-4, minibatches = 1, random  
}  
  
keys = classifiers.keys()  
  
for clf_name in keys:  
    clf = classifiers[clf_name]  
    clf_index = list(keys).index(clf_name)  
  
    for size_index, size in enumerate(dataset_sizes):  
        X_train_subset, y_train_subset = X_train_scaled[:size], y_train[:size]  
  
        for epoch_index, epochs in enumerate(epochs_list):  
            clf.epochs = epochs  
            clf.fit(X_train_subset, y_train_subset)  
  
            y_pred = clf.predict(X_test_scaled)  
            accuracy = np.mean(y_pred == y_test)  
            accuracies[clf_index, size_index, epoch_index] = accuracy  
  
            print(f"{clf_name} | Size: {size} | Epochs: {epochs} | Accuracy: {ac
```

Adaline		Size: 50		Epochs: 2		Accuracy: 0.8019
Adaline		Size: 50		Epochs: 7		Accuracy: 0.8514
Adaline		Size: 50		Epochs: 12		Accuracy: 0.8669
Adaline		Size: 50		Epochs: 17		Accuracy: 0.8638
Adaline		Size: 50		Epochs: 22		Accuracy: 0.8638
Adaline		Size: 50		Epochs: 27		Accuracy: 0.8607
Adaline		Size: 50		Epochs: 32		Accuracy: 0.8607
Adaline		Size: 50		Epochs: 37		Accuracy: 0.8576
Adaline		Size: 50		Epochs: 42		Accuracy: 0.8576
Adaline		Size: 50		Epochs: 47		Accuracy: 0.8576
Adaline		Size: 50		Epochs: 52		Accuracy: 0.8607
Adaline		Size: 50		Epochs: 57		Accuracy: 0.8607
Adaline		Size: 50		Epochs: 62		Accuracy: 0.8638
Adaline		Size: 50		Epochs: 67		Accuracy: 0.8638
Adaline		Size: 50		Epochs: 72		Accuracy: 0.8638
Adaline		Size: 50		Epochs: 77		Accuracy: 0.8669
Adaline		Size: 50		Epochs: 82		Accuracy: 0.8700
Adaline		Size: 50		Epochs: 87		Accuracy: 0.8669
Adaline		Size: 50		Epochs: 92		Accuracy: 0.8669
Adaline		Size: 50		Epochs: 97		Accuracy: 0.8700
Adaline		Size: 100		Epochs: 2		Accuracy: 0.8266
Adaline		Size: 100		Epochs: 7		Accuracy: 0.8731
Adaline		Size: 100		Epochs: 12		Accuracy: 0.8700
Adaline		Size: 100		Epochs: 17		Accuracy: 0.8700
Adaline		Size: 100		Epochs: 22		Accuracy: 0.8762
Adaline		Size: 100		Epochs: 27		Accuracy: 0.8762
Adaline		Size: 100		Epochs: 32		Accuracy: 0.8793
Adaline		Size: 100		Epochs: 37		Accuracy: 0.8762
Adaline		Size: 100		Epochs: 42		Accuracy: 0.8762
Adaline		Size: 100		Epochs: 47		Accuracy: 0.8762
Adaline		Size: 100		Epochs: 52		Accuracy: 0.8762
Adaline		Size: 100		Epochs: 57		Accuracy: 0.8793
Adaline		Size: 100		Epochs: 62		Accuracy: 0.8793
Adaline		Size: 100		Epochs: 67		Accuracy: 0.8793
Adaline		Size: 100		Epochs: 72		Accuracy: 0.8793
Adaline		Size: 100		Epochs: 77		Accuracy: 0.8762
Adaline		Size: 100		Epochs: 82		Accuracy: 0.8762
Adaline		Size: 100		Epochs: 87		Accuracy: 0.8762
Adaline		Size: 100		Epochs: 92		Accuracy: 0.8762
Adaline		Size: 100		Epochs: 97		Accuracy: 0.8762
Adaline		Size: 150		Epochs: 2		Accuracy: 0.8452
Adaline		Size: 150		Epochs: 7		Accuracy: 0.8700
Adaline		Size: 150		Epochs: 12		Accuracy: 0.8669
Adaline		Size: 150		Epochs: 17		Accuracy: 0.8731
Adaline		Size: 150		Epochs: 22		Accuracy: 0.8793
Adaline		Size: 150		Epochs: 27		Accuracy: 0.8824
Adaline		Size: 150		Epochs: 32		Accuracy: 0.8793
Adaline		Size: 150		Epochs: 37		Accuracy: 0.8793
Adaline		Size: 150		Epochs: 42		Accuracy: 0.8793
Adaline		Size: 150		Epochs: 47		Accuracy: 0.8793
Adaline		Size: 150		Epochs: 52		Accuracy: 0.8762
Adaline		Size: 150		Epochs: 57		Accuracy: 0.8762
Adaline		Size: 150		Epochs: 62		Accuracy: 0.8731
Adaline		Size: 150		Epochs: 67		Accuracy: 0.8731
Adaline		Size: 150		Epochs: 72		Accuracy: 0.8731
Adaline		Size: 150		Epochs: 77		Accuracy: 0.8731
Adaline		Size: 150		Epochs: 82		Accuracy: 0.8731
Adaline		Size: 150		Epochs: 87		Accuracy: 0.8731
Adaline		Size: 150		Epochs: 92		Accuracy: 0.8731
Adaline		Size: 150		Epochs: 97		Accuracy: 0.8731

Adaline		Size: 200		Epochs: 2		Accuracy: 0.8452
Adaline		Size: 200		Epochs: 7		Accuracy: 0.8669
Adaline		Size: 200		Epochs: 12		Accuracy: 0.8669
Adaline		Size: 200		Epochs: 17		Accuracy: 0.8700
Adaline		Size: 200		Epochs: 22		Accuracy: 0.8700
Adaline		Size: 200		Epochs: 27		Accuracy: 0.8669
Adaline		Size: 200		Epochs: 32		Accuracy: 0.8731
Adaline		Size: 200		Epochs: 37		Accuracy: 0.8731
Adaline		Size: 200		Epochs: 42		Accuracy: 0.8793
Adaline		Size: 200		Epochs: 47		Accuracy: 0.8762
Adaline		Size: 200		Epochs: 52		Accuracy: 0.8762
Adaline		Size: 200		Epochs: 57		Accuracy: 0.8762
Adaline		Size: 200		Epochs: 62		Accuracy: 0.8793
Adaline		Size: 200		Epochs: 67		Accuracy: 0.8793
Adaline		Size: 200		Epochs: 72		Accuracy: 0.8762
Adaline		Size: 200		Epochs: 77		Accuracy: 0.8762
Adaline		Size: 200		Epochs: 82		Accuracy: 0.8762
Adaline		Size: 200		Epochs: 87		Accuracy: 0.8731
Adaline		Size: 200		Epochs: 92		Accuracy: 0.8731
Adaline		Size: 200		Epochs: 97		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 2		Accuracy: 0.8545
Adaline		Size: 250		Epochs: 7		Accuracy: 0.8607
Adaline		Size: 250		Epochs: 12		Accuracy: 0.8669
Adaline		Size: 250		Epochs: 17		Accuracy: 0.8669
Adaline		Size: 250		Epochs: 22		Accuracy: 0.8700
Adaline		Size: 250		Epochs: 27		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 32		Accuracy: 0.8762
Adaline		Size: 250		Epochs: 37		Accuracy: 0.8762
Adaline		Size: 250		Epochs: 42		Accuracy: 0.8762
Adaline		Size: 250		Epochs: 47		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 52		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 57		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 62		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 67		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 72		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 77		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 82		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 87		Accuracy: 0.8731
Adaline		Size: 250		Epochs: 92		Accuracy: 0.8700
Adaline		Size: 250		Epochs: 97		Accuracy: 0.8700
Adaline		Size: 300		Epochs: 2		Accuracy: 0.8638
Adaline		Size: 300		Epochs: 7		Accuracy: 0.8700
Adaline		Size: 300		Epochs: 12		Accuracy: 0.8638
Adaline		Size: 300		Epochs: 17		Accuracy: 0.8700
Adaline		Size: 300		Epochs: 22		Accuracy: 0.8793
Adaline		Size: 300		Epochs: 27		Accuracy: 0.8762
Adaline		Size: 300		Epochs: 32		Accuracy: 0.8793
Adaline		Size: 300		Epochs: 37		Accuracy: 0.8762
Adaline		Size: 300		Epochs: 42		Accuracy: 0.8762
Adaline		Size: 300		Epochs: 47		Accuracy: 0.8731
Adaline		Size: 300		Epochs: 52		Accuracy: 0.8731
Adaline		Size: 300		Epochs: 57		Accuracy: 0.8731
Adaline		Size: 300		Epochs: 62		Accuracy: 0.8731
Adaline		Size: 300		Epochs: 67		Accuracy: 0.8731
Adaline		Size: 300		Epochs: 72		Accuracy: 0.8731
Adaline		Size: 300		Epochs: 77		Accuracy: 0.8731
Adaline		Size: 300		Epochs: 82		Accuracy: 0.8731
Adaline		Size: 300		Epochs: 87		Accuracy: 0.8731
Adaline		Size: 300		Epochs: 92		Accuracy: 0.8731
Adaline		Size: 300		Epochs: 97		Accuracy: 0.8731

Adaline		Size: 350		Epochs: 2		Accuracy: 0.8700
Adaline		Size: 350		Epochs: 7		Accuracy: 0.8638
Adaline		Size: 350		Epochs: 12		Accuracy: 0.8731
Adaline		Size: 350		Epochs: 17		Accuracy: 0.8669
Adaline		Size: 350		Epochs: 22		Accuracy: 0.8700
Adaline		Size: 350		Epochs: 27		Accuracy: 0.8731
Adaline		Size: 350		Epochs: 32		Accuracy: 0.8731
Adaline		Size: 350		Epochs: 37		Accuracy: 0.8762
Adaline		Size: 350		Epochs: 42		Accuracy: 0.8762
Adaline		Size: 350		Epochs: 47		Accuracy: 0.8731
Adaline		Size: 350		Epochs: 52		Accuracy: 0.8762
Adaline		Size: 350		Epochs: 57		Accuracy: 0.8793
Adaline		Size: 350		Epochs: 62		Accuracy: 0.8793
Adaline		Size: 350		Epochs: 67		Accuracy: 0.8793
Adaline		Size: 350		Epochs: 72		Accuracy: 0.8793
Adaline		Size: 350		Epochs: 77		Accuracy: 0.8762
Adaline		Size: 350		Epochs: 82		Accuracy: 0.8762
Adaline		Size: 350		Epochs: 87		Accuracy: 0.8762
Adaline		Size: 350		Epochs: 92		Accuracy: 0.8762
Adaline		Size: 350		Epochs: 97		Accuracy: 0.8762
Adaline		Size: 400		Epochs: 2		Accuracy: 0.8638
Adaline		Size: 400		Epochs: 7		Accuracy: 0.8731
Adaline		Size: 400		Epochs: 12		Accuracy: 0.8731
Adaline		Size: 400		Epochs: 17		Accuracy: 0.8731
Adaline		Size: 400		Epochs: 22		Accuracy: 0.8700
Adaline		Size: 400		Epochs: 27		Accuracy: 0.8762
Adaline		Size: 400		Epochs: 32		Accuracy: 0.8793
Adaline		Size: 400		Epochs: 37		Accuracy: 0.8762
Adaline		Size: 400		Epochs: 42		Accuracy: 0.8793
Adaline		Size: 400		Epochs: 47		Accuracy: 0.8762
Adaline		Size: 400		Epochs: 52		Accuracy: 0.8731
Adaline		Size: 400		Epochs: 57		Accuracy: 0.8731
Adaline		Size: 400		Epochs: 62		Accuracy: 0.8731
Adaline		Size: 400		Epochs: 67		Accuracy: 0.8700
Adaline		Size: 400		Epochs: 72		Accuracy: 0.8731
Adaline		Size: 400		Epochs: 77		Accuracy: 0.8731
Adaline		Size: 400		Epochs: 82		Accuracy: 0.8731
Adaline		Size: 400		Epochs: 87		Accuracy: 0.8700
Adaline		Size: 400		Epochs: 92		Accuracy: 0.8700
Adaline		Size: 400		Epochs: 97		Accuracy: 0.8700
Adaline		Size: 450		Epochs: 2		Accuracy: 0.8638
Adaline		Size: 450		Epochs: 7		Accuracy: 0.8762
Adaline		Size: 450		Epochs: 12		Accuracy: 0.8700
Adaline		Size: 450		Epochs: 17		Accuracy: 0.8762
Adaline		Size: 450		Epochs: 22		Accuracy: 0.8793
Adaline		Size: 450		Epochs: 27		Accuracy: 0.8731
Adaline		Size: 450		Epochs: 32		Accuracy: 0.8731
Adaline		Size: 450		Epochs: 37		Accuracy: 0.8762
Adaline		Size: 450		Epochs: 42		Accuracy: 0.8793
Adaline		Size: 450		Epochs: 47		Accuracy: 0.8824
Adaline		Size: 450		Epochs: 52		Accuracy: 0.8824
Adaline		Size: 450		Epochs: 57		Accuracy: 0.8854
Adaline		Size: 450		Epochs: 62		Accuracy: 0.8854
Adaline		Size: 450		Epochs: 67		Accuracy: 0.8854
Adaline		Size: 450		Epochs: 72		Accuracy: 0.8854
Adaline		Size: 450		Epochs: 77		Accuracy: 0.8854
Adaline		Size: 450		Epochs: 82		Accuracy: 0.8885
Adaline		Size: 450		Epochs: 87		Accuracy: 0.8885
Adaline		Size: 450		Epochs: 92		Accuracy: 0.8885
Adaline		Size: 450		Epochs: 97		Accuracy: 0.8885

Adaline		Size: 500		Epochs: 2		Accuracy: 0.8638
Adaline		Size: 500		Epochs: 7		Accuracy: 0.8669
Adaline		Size: 500		Epochs: 12		Accuracy: 0.8669
Adaline		Size: 500		Epochs: 17		Accuracy: 0.8762
Adaline		Size: 500		Epochs: 22		Accuracy: 0.8762
Adaline		Size: 500		Epochs: 27		Accuracy: 0.8762
Adaline		Size: 500		Epochs: 32		Accuracy: 0.8762
Adaline		Size: 500		Epochs: 37		Accuracy: 0.8793
Adaline		Size: 500		Epochs: 42		Accuracy: 0.8854
Adaline		Size: 500		Epochs: 47		Accuracy: 0.8854
Adaline		Size: 500		Epochs: 52		Accuracy: 0.8854
Adaline		Size: 500		Epochs: 57		Accuracy: 0.8854
Adaline		Size: 500		Epochs: 62		Accuracy: 0.8885
Adaline		Size: 500		Epochs: 67		Accuracy: 0.8885
Adaline		Size: 500		Epochs: 72		Accuracy: 0.8916
Adaline		Size: 500		Epochs: 77		Accuracy: 0.8885
Adaline		Size: 500		Epochs: 82		Accuracy: 0.8885
Adaline		Size: 500		Epochs: 87		Accuracy: 0.8885
Adaline		Size: 500		Epochs: 92		Accuracy: 0.8885
Adaline		Size: 500		Epochs: 97		Accuracy: 0.8916
Adaline		Size: 550		Epochs: 2		Accuracy: 0.8638
Adaline		Size: 550		Epochs: 7		Accuracy: 0.8700
Adaline		Size: 550		Epochs: 12		Accuracy: 0.8762
Adaline		Size: 550		Epochs: 17		Accuracy: 0.8762
Adaline		Size: 550		Epochs: 22		Accuracy: 0.8762
Adaline		Size: 550		Epochs: 27		Accuracy: 0.8762
Adaline		Size: 550		Epochs: 32		Accuracy: 0.8793
Adaline		Size: 550		Epochs: 37		Accuracy: 0.8824
Adaline		Size: 550		Epochs: 42		Accuracy: 0.8854
Adaline		Size: 550		Epochs: 47		Accuracy: 0.8885
Adaline		Size: 550		Epochs: 52		Accuracy: 0.8885
Adaline		Size: 550		Epochs: 57		Accuracy: 0.8916
Adaline		Size: 550		Epochs: 62		Accuracy: 0.8885
Adaline		Size: 550		Epochs: 67		Accuracy: 0.8885
Adaline		Size: 550		Epochs: 72		Accuracy: 0.8885
Adaline		Size: 550		Epochs: 77		Accuracy: 0.8885
Adaline		Size: 550		Epochs: 82		Accuracy: 0.8854
Adaline		Size: 550		Epochs: 87		Accuracy: 0.8854
Adaline		Size: 550		Epochs: 92		Accuracy: 0.8854
Adaline		Size: 550		Epochs: 97		Accuracy: 0.8854
Adaline		Size: 600		Epochs: 2		Accuracy: 0.8638
Adaline		Size: 600		Epochs: 7		Accuracy: 0.8731
Adaline		Size: 600		Epochs: 12		Accuracy: 0.8762
Adaline		Size: 600		Epochs: 17		Accuracy: 0.8793
Adaline		Size: 600		Epochs: 22		Accuracy: 0.8762
Adaline		Size: 600		Epochs: 27		Accuracy: 0.8762
Adaline		Size: 600		Epochs: 32		Accuracy: 0.8762
Adaline		Size: 600		Epochs: 37		Accuracy: 0.8793
Adaline		Size: 600		Epochs: 42		Accuracy: 0.8854
Adaline		Size: 600		Epochs: 47		Accuracy: 0.8885
Adaline		Size: 600		Epochs: 52		Accuracy: 0.8885
Adaline		Size: 600		Epochs: 57		Accuracy: 0.8916
Adaline		Size: 600		Epochs: 62		Accuracy: 0.8885
Adaline		Size: 600		Epochs: 67		Accuracy: 0.8885
Adaline		Size: 600		Epochs: 72		Accuracy: 0.8916
Adaline		Size: 600		Epochs: 77		Accuracy: 0.8854
Adaline		Size: 600		Epochs: 82		Accuracy: 0.8854
Adaline		Size: 600		Epochs: 87		Accuracy: 0.8854
Adaline		Size: 600		Epochs: 92		Accuracy: 0.8854
Adaline		Size: 600		Epochs: 97		Accuracy: 0.8854

Adaline		Size: 650		Epochs: 2		Accuracy: 0.8638
Adaline		Size: 650		Epochs: 7		Accuracy: 0.8700
Adaline		Size: 650		Epochs: 12		Accuracy: 0.8762
Adaline		Size: 650		Epochs: 17		Accuracy: 0.8762
Adaline		Size: 650		Epochs: 22		Accuracy: 0.8762
Adaline		Size: 650		Epochs: 27		Accuracy: 0.8731
Adaline		Size: 650		Epochs: 32		Accuracy: 0.8762
Adaline		Size: 650		Epochs: 37		Accuracy: 0.8824
Adaline		Size: 650		Epochs: 42		Accuracy: 0.8824
Adaline		Size: 650		Epochs: 47		Accuracy: 0.8854
Adaline		Size: 650		Epochs: 52		Accuracy: 0.8854
Adaline		Size: 650		Epochs: 57		Accuracy: 0.8885
Adaline		Size: 650		Epochs: 62		Accuracy: 0.8885
Adaline		Size: 650		Epochs: 67		Accuracy: 0.8854
Adaline		Size: 650		Epochs: 72		Accuracy: 0.8885
Adaline		Size: 650		Epochs: 77		Accuracy: 0.8885
Adaline		Size: 650		Epochs: 82		Accuracy: 0.8885
Adaline		Size: 650		Epochs: 87		Accuracy: 0.8885
Adaline		Size: 650		Epochs: 92		Accuracy: 0.8885
Adaline		Size: 650		Epochs: 97		Accuracy: 0.8854
Adaline		Size: 700		Epochs: 2		Accuracy: 0.8638
Adaline		Size: 700		Epochs: 7		Accuracy: 0.8700
Adaline		Size: 700		Epochs: 12		Accuracy: 0.8762
Adaline		Size: 700		Epochs: 17		Accuracy: 0.8731
Adaline		Size: 700		Epochs: 22		Accuracy: 0.8762
Adaline		Size: 700		Epochs: 27		Accuracy: 0.8793
Adaline		Size: 700		Epochs: 32		Accuracy: 0.8793
Adaline		Size: 700		Epochs: 37		Accuracy: 0.8824
Adaline		Size: 700		Epochs: 42		Accuracy: 0.8824
Adaline		Size: 700		Epochs: 47		Accuracy: 0.8854
Adaline		Size: 700		Epochs: 52		Accuracy: 0.8885
Adaline		Size: 700		Epochs: 57		Accuracy: 0.8885
Adaline		Size: 700		Epochs: 62		Accuracy: 0.8885
Adaline		Size: 700		Epochs: 67		Accuracy: 0.8854
Adaline		Size: 700		Epochs: 72		Accuracy: 0.8885
Adaline		Size: 700		Epochs: 77		Accuracy: 0.8885
Adaline		Size: 700		Epochs: 82		Accuracy: 0.8885
Adaline		Size: 700		Epochs: 87		Accuracy: 0.8885
Adaline		Size: 700		Epochs: 92		Accuracy: 0.8885
Adaline		Size: 700		Epochs: 97		Accuracy: 0.8885
Adaline		Size: 750		Epochs: 2		Accuracy: 0.8669
Adaline		Size: 750		Epochs: 7		Accuracy: 0.8700
Adaline		Size: 750		Epochs: 12		Accuracy: 0.8731
Adaline		Size: 750		Epochs: 17		Accuracy: 0.8762
Adaline		Size: 750		Epochs: 22		Accuracy: 0.8731
Adaline		Size: 750		Epochs: 27		Accuracy: 0.8762
Adaline		Size: 750		Epochs: 32		Accuracy: 0.8731
Adaline		Size: 750		Epochs: 37		Accuracy: 0.8793
Adaline		Size: 750		Epochs: 42		Accuracy: 0.8793
Adaline		Size: 750		Epochs: 47		Accuracy: 0.8793
Adaline		Size: 750		Epochs: 52		Accuracy: 0.8793
Adaline		Size: 750		Epochs: 57		Accuracy: 0.8793
Adaline		Size: 750		Epochs: 62		Accuracy: 0.8793
Adaline		Size: 750		Epochs: 67		Accuracy: 0.8824
Adaline		Size: 750		Epochs: 72		Accuracy: 0.8824
Adaline		Size: 750		Epochs: 77		Accuracy: 0.8824
Adaline		Size: 750		Epochs: 82		Accuracy: 0.8824
Adaline		Size: 750		Epochs: 87		Accuracy: 0.8854
Adaline		Size: 750		Epochs: 92		Accuracy: 0.8854
Adaline		Size: 750		Epochs: 97		Accuracy: 0.8854

Perceptron	Size: 50	Epochs: 2	Accuracy: 0.8111
Perceptron	Size: 50	Epochs: 7	Accuracy: 0.8638
Perceptron	Size: 50	Epochs: 12	Accuracy: 0.8638
Perceptron	Size: 50	Epochs: 17	Accuracy: 0.8669
Perceptron	Size: 50	Epochs: 22	Accuracy: 0.8824
Perceptron	Size: 50	Epochs: 27	Accuracy: 0.8793
Perceptron	Size: 50	Epochs: 32	Accuracy: 0.8947
Perceptron	Size: 50	Epochs: 37	Accuracy: 0.9009
Perceptron	Size: 50	Epochs: 42	Accuracy: 0.9009
Perceptron	Size: 50	Epochs: 47	Accuracy: 0.8916
Perceptron	Size: 50	Epochs: 52	Accuracy: 0.8947
Perceptron	Size: 50	Epochs: 57	Accuracy: 0.8885
Perceptron	Size: 50	Epochs: 62	Accuracy: 0.8700
Perceptron	Size: 50	Epochs: 67	Accuracy: 0.8885
Perceptron	Size: 50	Epochs: 72	Accuracy: 0.8885
Perceptron	Size: 50	Epochs: 77	Accuracy: 0.8700
Perceptron	Size: 50	Epochs: 82	Accuracy: 0.8762
Perceptron	Size: 50	Epochs: 87	Accuracy: 0.8514
Perceptron	Size: 50	Epochs: 92	Accuracy: 0.8916
Perceptron	Size: 50	Epochs: 97	Accuracy: 0.8854
Perceptron	Size: 100	Epochs: 2	Accuracy: 0.8266
Perceptron	Size: 100	Epochs: 7	Accuracy: 0.8793
Perceptron	Size: 100	Epochs: 12	Accuracy: 0.8793
Perceptron	Size: 100	Epochs: 17	Accuracy: 0.8824
Perceptron	Size: 100	Epochs: 22	Accuracy: 0.8793
Perceptron	Size: 100	Epochs: 27	Accuracy: 0.8916
Perceptron	Size: 100	Epochs: 32	Accuracy: 0.8607
Perceptron	Size: 100	Epochs: 37	Accuracy: 0.8266
Perceptron	Size: 100	Epochs: 42	Accuracy: 0.8700
Perceptron	Size: 100	Epochs: 47	Accuracy: 0.8669
Perceptron	Size: 100	Epochs: 52	Accuracy: 0.7647
Perceptron	Size: 100	Epochs: 57	Accuracy: 0.8700
Perceptron	Size: 100	Epochs: 62	Accuracy: 0.8793
Perceptron	Size: 100	Epochs: 67	Accuracy: 0.8607
Perceptron	Size: 100	Epochs: 72	Accuracy: 0.8793
Perceptron	Size: 100	Epochs: 77	Accuracy: 0.8638
Perceptron	Size: 100	Epochs: 82	Accuracy: 0.8762
Perceptron	Size: 100	Epochs: 87	Accuracy: 0.8731
Perceptron	Size: 100	Epochs: 92	Accuracy: 0.8854
Perceptron	Size: 100	Epochs: 97	Accuracy: 0.8700
Perceptron	Size: 150	Epochs: 2	Accuracy: 0.8421
Perceptron	Size: 150	Epochs: 7	Accuracy: 0.8669
Perceptron	Size: 150	Epochs: 12	Accuracy: 0.8669
Perceptron	Size: 150	Epochs: 17	Accuracy: 0.8266
Perceptron	Size: 150	Epochs: 22	Accuracy: 0.8452
Perceptron	Size: 150	Epochs: 27	Accuracy: 0.8328
Perceptron	Size: 150	Epochs: 32	Accuracy: 0.8731
Perceptron	Size: 150	Epochs: 37	Accuracy: 0.8576
Perceptron	Size: 150	Epochs: 42	Accuracy: 0.8793
Perceptron	Size: 150	Epochs: 47	Accuracy: 0.6718
Perceptron	Size: 150	Epochs: 52	Accuracy: 0.8421
Perceptron	Size: 150	Epochs: 57	Accuracy: 0.8669
Perceptron	Size: 150	Epochs: 62	Accuracy: 0.8297
Perceptron	Size: 150	Epochs: 67	Accuracy: 0.8390
Perceptron	Size: 150	Epochs: 72	Accuracy: 0.8793
Perceptron	Size: 150	Epochs: 77	Accuracy: 0.8545
Perceptron	Size: 150	Epochs: 82	Accuracy: 0.8885
Perceptron	Size: 150	Epochs: 87	Accuracy: 0.8669
Perceptron	Size: 150	Epochs: 92	Accuracy: 0.8700
Perceptron	Size: 150	Epochs: 97	Accuracy: 0.8638

Perceptron	Size: 200	Epochs: 2	Accuracy: 0.8669
Perceptron	Size: 200	Epochs: 7	Accuracy: 0.8916
Perceptron	Size: 200	Epochs: 12	Accuracy: 0.8854
Perceptron	Size: 200	Epochs: 17	Accuracy: 0.8359
Perceptron	Size: 200	Epochs: 22	Accuracy: 0.9040
Perceptron	Size: 200	Epochs: 27	Accuracy: 0.8824
Perceptron	Size: 200	Epochs: 32	Accuracy: 0.8793
Perceptron	Size: 200	Epochs: 37	Accuracy: 0.8824
Perceptron	Size: 200	Epochs: 42	Accuracy: 0.8545
Perceptron	Size: 200	Epochs: 47	Accuracy: 0.8297
Perceptron	Size: 200	Epochs: 52	Accuracy: 0.8328
Perceptron	Size: 200	Epochs: 57	Accuracy: 0.8235
Perceptron	Size: 200	Epochs: 62	Accuracy: 0.9009
Perceptron	Size: 200	Epochs: 67	Accuracy: 0.8731
Perceptron	Size: 200	Epochs: 72	Accuracy: 0.8731
Perceptron	Size: 200	Epochs: 77	Accuracy: 0.7926
Perceptron	Size: 200	Epochs: 82	Accuracy: 0.8514
Perceptron	Size: 200	Epochs: 87	Accuracy: 0.8978
Perceptron	Size: 200	Epochs: 92	Accuracy: 0.9040
Perceptron	Size: 200	Epochs: 97	Accuracy: 0.8916
Perceptron	Size: 250	Epochs: 2	Accuracy: 0.8700
Perceptron	Size: 250	Epochs: 7	Accuracy: 0.8576
Perceptron	Size: 250	Epochs: 12	Accuracy: 0.8978
Perceptron	Size: 250	Epochs: 17	Accuracy: 0.8607
Perceptron	Size: 250	Epochs: 22	Accuracy: 0.8700
Perceptron	Size: 250	Epochs: 27	Accuracy: 0.8854
Perceptron	Size: 250	Epochs: 32	Accuracy: 0.8452
Perceptron	Size: 250	Epochs: 37	Accuracy: 0.8452
Perceptron	Size: 250	Epochs: 42	Accuracy: 0.8700
Perceptron	Size: 250	Epochs: 47	Accuracy: 0.8390
Perceptron	Size: 250	Epochs: 52	Accuracy: 0.8390
Perceptron	Size: 250	Epochs: 57	Accuracy: 0.8978
Perceptron	Size: 250	Epochs: 62	Accuracy: 0.8545
Perceptron	Size: 250	Epochs: 67	Accuracy: 0.8545
Perceptron	Size: 250	Epochs: 72	Accuracy: 0.8824
Perceptron	Size: 250	Epochs: 77	Accuracy: 0.8916
Perceptron	Size: 250	Epochs: 82	Accuracy: 0.8916
Perceptron	Size: 250	Epochs: 87	Accuracy: 0.8916
Perceptron	Size: 250	Epochs: 92	Accuracy: 0.8916
Perceptron	Size: 250	Epochs: 97	Accuracy: 0.7895
Perceptron	Size: 300	Epochs: 2	Accuracy: 0.8545
Perceptron	Size: 300	Epochs: 7	Accuracy: 0.8452
Perceptron	Size: 300	Epochs: 12	Accuracy: 0.8545
Perceptron	Size: 300	Epochs: 17	Accuracy: 0.8824
Perceptron	Size: 300	Epochs: 22	Accuracy: 0.8638
Perceptron	Size: 300	Epochs: 27	Accuracy: 0.8607
Perceptron	Size: 300	Epochs: 32	Accuracy: 0.8885
Perceptron	Size: 300	Epochs: 37	Accuracy: 0.7895
Perceptron	Size: 300	Epochs: 42	Accuracy: 0.8885
Perceptron	Size: 300	Epochs: 47	Accuracy: 0.8359
Perceptron	Size: 300	Epochs: 52	Accuracy: 0.6130
Perceptron	Size: 300	Epochs: 57	Accuracy: 0.8824
Perceptron	Size: 300	Epochs: 62	Accuracy: 0.8452
Perceptron	Size: 300	Epochs: 67	Accuracy: 0.8731
Perceptron	Size: 300	Epochs: 72	Accuracy: 0.8824
Perceptron	Size: 300	Epochs: 77	Accuracy: 0.8793
Perceptron	Size: 300	Epochs: 82	Accuracy: 0.8947
Perceptron	Size: 300	Epochs: 87	Accuracy: 0.8390
Perceptron	Size: 300	Epochs: 92	Accuracy: 0.8700
Perceptron	Size: 300	Epochs: 97	Accuracy: 0.8266

Perceptron	Size: 350	Epochs: 2	Accuracy: 0.8638
Perceptron	Size: 350	Epochs: 7	Accuracy: 0.8885
Perceptron	Size: 350	Epochs: 12	Accuracy: 0.8576
Perceptron	Size: 350	Epochs: 17	Accuracy: 0.8762
Perceptron	Size: 350	Epochs: 22	Accuracy: 0.8328
Perceptron	Size: 350	Epochs: 27	Accuracy: 0.8885
Perceptron	Size: 350	Epochs: 32	Accuracy: 0.8824
Perceptron	Size: 350	Epochs: 37	Accuracy: 0.8700
Perceptron	Size: 350	Epochs: 42	Accuracy: 0.8545
Perceptron	Size: 350	Epochs: 47	Accuracy: 0.8700
Perceptron	Size: 350	Epochs: 52	Accuracy: 0.8762
Perceptron	Size: 350	Epochs: 57	Accuracy: 0.8607
Perceptron	Size: 350	Epochs: 62	Accuracy: 0.8824
Perceptron	Size: 350	Epochs: 67	Accuracy: 0.8514
Perceptron	Size: 350	Epochs: 72	Accuracy: 0.8669
Perceptron	Size: 350	Epochs: 77	Accuracy: 0.8916
Perceptron	Size: 350	Epochs: 82	Accuracy: 0.8854
Perceptron	Size: 350	Epochs: 87	Accuracy: 0.8824
Perceptron	Size: 350	Epochs: 92	Accuracy: 0.8824
Perceptron	Size: 350	Epochs: 97	Accuracy: 0.8793
Perceptron	Size: 400	Epochs: 2	Accuracy: 0.8638
Perceptron	Size: 400	Epochs: 7	Accuracy: 0.8793
Perceptron	Size: 400	Epochs: 12	Accuracy: 0.8885
Perceptron	Size: 400	Epochs: 17	Accuracy: 0.8669
Perceptron	Size: 400	Epochs: 22	Accuracy: 0.8235
Perceptron	Size: 400	Epochs: 27	Accuracy: 0.9009
Perceptron	Size: 400	Epochs: 32	Accuracy: 0.8793
Perceptron	Size: 400	Epochs: 37	Accuracy: 0.8731
Perceptron	Size: 400	Epochs: 42	Accuracy: 0.8576
Perceptron	Size: 400	Epochs: 47	Accuracy: 0.7337
Perceptron	Size: 400	Epochs: 52	Accuracy: 0.8328
Perceptron	Size: 400	Epochs: 57	Accuracy: 0.8700
Perceptron	Size: 400	Epochs: 62	Accuracy: 0.8885
Perceptron	Size: 400	Epochs: 67	Accuracy: 0.7430
Perceptron	Size: 400	Epochs: 72	Accuracy: 0.8854
Perceptron	Size: 400	Epochs: 77	Accuracy: 0.7368
Perceptron	Size: 400	Epochs: 82	Accuracy: 0.8824
Perceptron	Size: 400	Epochs: 87	Accuracy: 0.8947
Perceptron	Size: 400	Epochs: 92	Accuracy: 0.8885
Perceptron	Size: 400	Epochs: 97	Accuracy: 0.8576
Perceptron	Size: 450	Epochs: 2	Accuracy: 0.8793
Perceptron	Size: 450	Epochs: 7	Accuracy: 0.8978
Perceptron	Size: 450	Epochs: 12	Accuracy: 0.8762
Perceptron	Size: 450	Epochs: 17	Accuracy: 0.7802
Perceptron	Size: 450	Epochs: 22	Accuracy: 0.8700
Perceptron	Size: 450	Epochs: 27	Accuracy: 0.8762
Perceptron	Size: 450	Epochs: 32	Accuracy: 0.7926
Perceptron	Size: 450	Epochs: 37	Accuracy: 0.8824
Perceptron	Size: 450	Epochs: 42	Accuracy: 0.8824
Perceptron	Size: 450	Epochs: 47	Accuracy: 0.8359
Perceptron	Size: 450	Epochs: 52	Accuracy: 0.8731
Perceptron	Size: 450	Epochs: 57	Accuracy: 0.8793
Perceptron	Size: 450	Epochs: 62	Accuracy: 0.8731
Perceptron	Size: 450	Epochs: 67	Accuracy: 0.8483
Perceptron	Size: 450	Epochs: 72	Accuracy: 0.8854
Perceptron	Size: 450	Epochs: 77	Accuracy: 0.8793
Perceptron	Size: 450	Epochs: 82	Accuracy: 0.8328
Perceptron	Size: 450	Epochs: 87	Accuracy: 0.8576
Perceptron	Size: 450	Epochs: 92	Accuracy: 0.8545
Perceptron	Size: 450	Epochs: 97	Accuracy: 0.6006

Perceptron	Size: 500	Epochs: 2	Accuracy: 0.8669
Perceptron	Size: 500	Epochs: 7	Accuracy: 0.8700
Perceptron	Size: 500	Epochs: 12	Accuracy: 0.8731
Perceptron	Size: 500	Epochs: 17	Accuracy: 0.8854
Perceptron	Size: 500	Epochs: 22	Accuracy: 0.8359
Perceptron	Size: 500	Epochs: 27	Accuracy: 0.7895
Perceptron	Size: 500	Epochs: 32	Accuracy: 0.8947
Perceptron	Size: 500	Epochs: 37	Accuracy: 0.8050
Perceptron	Size: 500	Epochs: 42	Accuracy: 0.7214
Perceptron	Size: 500	Epochs: 47	Accuracy: 0.8762
Perceptron	Size: 500	Epochs: 52	Accuracy: 0.7523
Perceptron	Size: 500	Epochs: 57	Accuracy: 0.8452
Perceptron	Size: 500	Epochs: 62	Accuracy: 0.8545
Perceptron	Size: 500	Epochs: 67	Accuracy: 0.8700
Perceptron	Size: 500	Epochs: 72	Accuracy: 0.8576
Perceptron	Size: 500	Epochs: 77	Accuracy: 0.8885
Perceptron	Size: 500	Epochs: 82	Accuracy: 0.8793
Perceptron	Size: 500	Epochs: 87	Accuracy: 0.8235
Perceptron	Size: 500	Epochs: 92	Accuracy: 0.8885
Perceptron	Size: 500	Epochs: 97	Accuracy: 0.8762
Perceptron	Size: 550	Epochs: 2	Accuracy: 0.8421
Perceptron	Size: 550	Epochs: 7	Accuracy: 0.8885
Perceptron	Size: 550	Epochs: 12	Accuracy: 0.9102
Perceptron	Size: 550	Epochs: 17	Accuracy: 0.7988
Perceptron	Size: 550	Epochs: 22	Accuracy: 0.7678
Perceptron	Size: 550	Epochs: 27	Accuracy: 0.8854
Perceptron	Size: 550	Epochs: 32	Accuracy: 0.8793
Perceptron	Size: 550	Epochs: 37	Accuracy: 0.8793
Perceptron	Size: 550	Epochs: 42	Accuracy: 0.9071
Perceptron	Size: 550	Epochs: 47	Accuracy: 0.8700
Perceptron	Size: 550	Epochs: 52	Accuracy: 0.8885
Perceptron	Size: 550	Epochs: 57	Accuracy: 0.6718
Perceptron	Size: 550	Epochs: 62	Accuracy: 0.8297
Perceptron	Size: 550	Epochs: 67	Accuracy: 0.8359
Perceptron	Size: 550	Epochs: 72	Accuracy: 0.8824
Perceptron	Size: 550	Epochs: 77	Accuracy: 0.8762
Perceptron	Size: 550	Epochs: 82	Accuracy: 0.8824
Perceptron	Size: 550	Epochs: 87	Accuracy: 0.8885
Perceptron	Size: 550	Epochs: 92	Accuracy: 0.8854
Perceptron	Size: 550	Epochs: 97	Accuracy: 0.8359
Perceptron	Size: 600	Epochs: 2	Accuracy: 0.8793
Perceptron	Size: 600	Epochs: 7	Accuracy: 0.8762
Perceptron	Size: 600	Epochs: 12	Accuracy: 0.9009
Perceptron	Size: 600	Epochs: 17	Accuracy: 0.8916
Perceptron	Size: 600	Epochs: 22	Accuracy: 0.8824
Perceptron	Size: 600	Epochs: 27	Accuracy: 0.8916
Perceptron	Size: 600	Epochs: 32	Accuracy: 0.8731
Perceptron	Size: 600	Epochs: 37	Accuracy: 0.8328
Perceptron	Size: 600	Epochs: 42	Accuracy: 0.7988
Perceptron	Size: 600	Epochs: 47	Accuracy: 0.8483
Perceptron	Size: 600	Epochs: 52	Accuracy: 0.8359
Perceptron	Size: 600	Epochs: 57	Accuracy: 0.8885
Perceptron	Size: 600	Epochs: 62	Accuracy: 0.8885
Perceptron	Size: 600	Epochs: 67	Accuracy: 0.8854
Perceptron	Size: 600	Epochs: 72	Accuracy: 0.8916
Perceptron	Size: 600	Epochs: 77	Accuracy: 0.9009
Perceptron	Size: 600	Epochs: 82	Accuracy: 0.8700
Perceptron	Size: 600	Epochs: 87	Accuracy: 0.8916
Perceptron	Size: 600	Epochs: 92	Accuracy: 0.7214
Perceptron	Size: 600	Epochs: 97	Accuracy: 0.8483

Perceptron	Size: 650	Epochs: 2	Accuracy: 0.8173
Perceptron	Size: 650	Epochs: 7	Accuracy: 0.8793
Perceptron	Size: 650	Epochs: 12	Accuracy: 0.9009
Perceptron	Size: 650	Epochs: 17	Accuracy: 0.8235
Perceptron	Size: 650	Epochs: 22	Accuracy: 0.6718
Perceptron	Size: 650	Epochs: 27	Accuracy: 0.8885
Perceptron	Size: 650	Epochs: 32	Accuracy: 0.7276
Perceptron	Size: 650	Epochs: 37	Accuracy: 0.8204
Perceptron	Size: 650	Epochs: 42	Accuracy: 0.8204
Perceptron	Size: 650	Epochs: 47	Accuracy: 0.8607
Perceptron	Size: 650	Epochs: 52	Accuracy: 0.7678
Perceptron	Size: 650	Epochs: 57	Accuracy: 0.8607
Perceptron	Size: 650	Epochs: 62	Accuracy: 0.8297
Perceptron	Size: 650	Epochs: 67	Accuracy: 0.8638
Perceptron	Size: 650	Epochs: 72	Accuracy: 0.8390
Perceptron	Size: 650	Epochs: 77	Accuracy: 0.8390
Perceptron	Size: 650	Epochs: 82	Accuracy: 0.7864
Perceptron	Size: 650	Epochs: 87	Accuracy: 0.8638
Perceptron	Size: 650	Epochs: 92	Accuracy: 0.8080
Perceptron	Size: 650	Epochs: 97	Accuracy: 0.8019
Perceptron	Size: 700	Epochs: 2	Accuracy: 0.8421
Perceptron	Size: 700	Epochs: 7	Accuracy: 0.8793
Perceptron	Size: 700	Epochs: 12	Accuracy: 0.7399
Perceptron	Size: 700	Epochs: 17	Accuracy: 0.6285
Perceptron	Size: 700	Epochs: 22	Accuracy: 0.8452
Perceptron	Size: 700	Epochs: 27	Accuracy: 0.8050
Perceptron	Size: 700	Epochs: 32	Accuracy: 0.8576
Perceptron	Size: 700	Epochs: 37	Accuracy: 0.8297
Perceptron	Size: 700	Epochs: 42	Accuracy: 0.8669
Perceptron	Size: 700	Epochs: 47	Accuracy: 0.8638
Perceptron	Size: 700	Epochs: 52	Accuracy: 0.8607
Perceptron	Size: 700	Epochs: 57	Accuracy: 0.8885
Perceptron	Size: 700	Epochs: 62	Accuracy: 0.8731
Perceptron	Size: 700	Epochs: 67	Accuracy: 0.8545
Perceptron	Size: 700	Epochs: 72	Accuracy: 0.6594
Perceptron	Size: 700	Epochs: 77	Accuracy: 0.8019
Perceptron	Size: 700	Epochs: 82	Accuracy: 0.7771
Perceptron	Size: 700	Epochs: 87	Accuracy: 0.8669
Perceptron	Size: 700	Epochs: 92	Accuracy: 0.7368
Perceptron	Size: 700	Epochs: 97	Accuracy: 0.8576
Perceptron	Size: 750	Epochs: 2	Accuracy: 0.8545
Perceptron	Size: 750	Epochs: 7	Accuracy: 0.8762
Perceptron	Size: 750	Epochs: 12	Accuracy: 0.8916
Perceptron	Size: 750	Epochs: 17	Accuracy: 0.9009
Perceptron	Size: 750	Epochs: 22	Accuracy: 0.8452
Perceptron	Size: 750	Epochs: 27	Accuracy: 0.8638
Perceptron	Size: 750	Epochs: 32	Accuracy: 0.8607
Perceptron	Size: 750	Epochs: 37	Accuracy: 0.8916
Perceptron	Size: 750	Epochs: 42	Accuracy: 0.8669
Perceptron	Size: 750	Epochs: 47	Accuracy: 0.8854
Perceptron	Size: 750	Epochs: 52	Accuracy: 0.8824
Perceptron	Size: 750	Epochs: 57	Accuracy: 0.7988
Perceptron	Size: 750	Epochs: 62	Accuracy: 0.8762
Perceptron	Size: 750	Epochs: 67	Accuracy: 0.8669
Perceptron	Size: 750	Epochs: 72	Accuracy: 0.8514
Perceptron	Size: 750	Epochs: 77	Accuracy: 0.8854
Perceptron	Size: 750	Epochs: 82	Accuracy: 0.8978
Perceptron	Size: 750	Epochs: 87	Accuracy: 0.8916
Perceptron	Size: 750	Epochs: 92	Accuracy: 0.8700
Perceptron	Size: 750	Epochs: 97	Accuracy: 0.8854

LogisticRegression	Size: 50	Epochs: 2	Accuracy: 0.7833
LogisticRegression	Size: 50	Epochs: 7	Accuracy: 0.8235
LogisticRegression	Size: 50	Epochs: 12	Accuracy: 0.8452
LogisticRegression	Size: 50	Epochs: 17	Accuracy: 0.8514
LogisticRegression	Size: 50	Epochs: 22	Accuracy: 0.8607
LogisticRegression	Size: 50	Epochs: 27	Accuracy: 0.8669
LogisticRegression	Size: 50	Epochs: 32	Accuracy: 0.8669
LogisticRegression	Size: 50	Epochs: 37	Accuracy: 0.8638
LogisticRegression	Size: 50	Epochs: 42	Accuracy: 0.8638
LogisticRegression	Size: 50	Epochs: 47	Accuracy: 0.8607
LogisticRegression	Size: 50	Epochs: 52	Accuracy: 0.8638
LogisticRegression	Size: 50	Epochs: 57	Accuracy: 0.8576
LogisticRegression	Size: 50	Epochs: 62	Accuracy: 0.8545
LogisticRegression	Size: 50	Epochs: 67	Accuracy: 0.8545
LogisticRegression	Size: 50	Epochs: 72	Accuracy: 0.8545
LogisticRegression	Size: 50	Epochs: 77	Accuracy: 0.8545
LogisticRegression	Size: 50	Epochs: 82	Accuracy: 0.8576
LogisticRegression	Size: 50	Epochs: 87	Accuracy: 0.8545
LogisticRegression	Size: 50	Epochs: 92	Accuracy: 0.8576
LogisticRegression	Size: 50	Epochs: 97	Accuracy: 0.8576
LogisticRegression	Size: 100	Epochs: 2	Accuracy: 0.8019
LogisticRegression	Size: 100	Epochs: 7	Accuracy: 0.8576
LogisticRegression	Size: 100	Epochs: 12	Accuracy: 0.8793
LogisticRegression	Size: 100	Epochs: 17	Accuracy: 0.8731
LogisticRegression	Size: 100	Epochs: 22	Accuracy: 0.8731
LogisticRegression	Size: 100	Epochs: 27	Accuracy: 0.8731
LogisticRegression	Size: 100	Epochs: 32	Accuracy: 0.8731
LogisticRegression	Size: 100	Epochs: 37	Accuracy: 0.8700
LogisticRegression	Size: 100	Epochs: 42	Accuracy: 0.8700
LogisticRegression	Size: 100	Epochs: 47	Accuracy: 0.8700
LogisticRegression	Size: 100	Epochs: 52	Accuracy: 0.8731
LogisticRegression	Size: 100	Epochs: 57	Accuracy: 0.8731
LogisticRegression	Size: 100	Epochs: 62	Accuracy: 0.8762
LogisticRegression	Size: 100	Epochs: 67	Accuracy: 0.8762
LogisticRegression	Size: 100	Epochs: 72	Accuracy: 0.8762
LogisticRegression	Size: 100	Epochs: 77	Accuracy: 0.8762
LogisticRegression	Size: 100	Epochs: 82	Accuracy: 0.8762
LogisticRegression	Size: 100	Epochs: 87	Accuracy: 0.8793
LogisticRegression	Size: 100	Epochs: 92	Accuracy: 0.8793
LogisticRegression	Size: 100	Epochs: 97	Accuracy: 0.8793
LogisticRegression	Size: 150	Epochs: 2	Accuracy: 0.8080
LogisticRegression	Size: 150	Epochs: 7	Accuracy: 0.8638
LogisticRegression	Size: 150	Epochs: 12	Accuracy: 0.8669
LogisticRegression	Size: 150	Epochs: 17	Accuracy: 0.8700
LogisticRegression	Size: 150	Epochs: 22	Accuracy: 0.8731
LogisticRegression	Size: 150	Epochs: 27	Accuracy: 0.8731
LogisticRegression	Size: 150	Epochs: 32	Accuracy: 0.8700
LogisticRegression	Size: 150	Epochs: 37	Accuracy: 0.8700
LogisticRegression	Size: 150	Epochs: 42	Accuracy: 0.8762
LogisticRegression	Size: 150	Epochs: 47	Accuracy: 0.8731
LogisticRegression	Size: 150	Epochs: 52	Accuracy: 0.8762
LogisticRegression	Size: 150	Epochs: 57	Accuracy: 0.8793
LogisticRegression	Size: 150	Epochs: 62	Accuracy: 0.8793
LogisticRegression	Size: 150	Epochs: 67	Accuracy: 0.8793
LogisticRegression	Size: 150	Epochs: 72	Accuracy: 0.8824
LogisticRegression	Size: 150	Epochs: 77	Accuracy: 0.8824
LogisticRegression	Size: 150	Epochs: 82	Accuracy: 0.8824
LogisticRegression	Size: 150	Epochs: 87	Accuracy: 0.8824
LogisticRegression	Size: 150	Epochs: 92	Accuracy: 0.8824
LogisticRegression	Size: 150	Epochs: 97	Accuracy: 0.8824

LogisticRegression	Size: 200	Epochs: 2	Accuracy: 0.8142
LogisticRegression	Size: 200	Epochs: 7	Accuracy: 0.8607
LogisticRegression	Size: 200	Epochs: 12	Accuracy: 0.8669
LogisticRegression	Size: 200	Epochs: 17	Accuracy: 0.8669
LogisticRegression	Size: 200	Epochs: 22	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 27	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 32	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 37	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 42	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 47	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 52	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 57	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 62	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 67	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 72	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 77	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 82	Accuracy: 0.8700
LogisticRegression	Size: 200	Epochs: 87	Accuracy: 0.8669
LogisticRegression	Size: 200	Epochs: 92	Accuracy: 0.8669
LogisticRegression	Size: 200	Epochs: 97	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 2	Accuracy: 0.8204
LogisticRegression	Size: 250	Epochs: 7	Accuracy: 0.8731
LogisticRegression	Size: 250	Epochs: 12	Accuracy: 0.8638
LogisticRegression	Size: 250	Epochs: 17	Accuracy: 0.8607
LogisticRegression	Size: 250	Epochs: 22	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 27	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 32	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 37	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 42	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 47	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 52	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 57	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 62	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 67	Accuracy: 0.8638
LogisticRegression	Size: 250	Epochs: 72	Accuracy: 0.8638
LogisticRegression	Size: 250	Epochs: 77	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 82	Accuracy: 0.8669
LogisticRegression	Size: 250	Epochs: 87	Accuracy: 0.8700
LogisticRegression	Size: 250	Epochs: 92	Accuracy: 0.8700
LogisticRegression	Size: 250	Epochs: 97	Accuracy: 0.8700
LogisticRegression	Size: 300	Epochs: 2	Accuracy: 0.8452
LogisticRegression	Size: 300	Epochs: 7	Accuracy: 0.8638
LogisticRegression	Size: 300	Epochs: 12	Accuracy: 0.8669
LogisticRegression	Size: 300	Epochs: 17	Accuracy: 0.8700
LogisticRegression	Size: 300	Epochs: 22	Accuracy: 0.8700
LogisticRegression	Size: 300	Epochs: 27	Accuracy: 0.8700
LogisticRegression	Size: 300	Epochs: 32	Accuracy: 0.8669
LogisticRegression	Size: 300	Epochs: 37	Accuracy: 0.8669
LogisticRegression	Size: 300	Epochs: 42	Accuracy: 0.8669
LogisticRegression	Size: 300	Epochs: 47	Accuracy: 0.8731
LogisticRegression	Size: 300	Epochs: 52	Accuracy: 0.8731
LogisticRegression	Size: 300	Epochs: 57	Accuracy: 0.8731
LogisticRegression	Size: 300	Epochs: 62	Accuracy: 0.8700
LogisticRegression	Size: 300	Epochs: 67	Accuracy: 0.8700
LogisticRegression	Size: 300	Epochs: 72	Accuracy: 0.8731
LogisticRegression	Size: 300	Epochs: 77	Accuracy: 0.8731
LogisticRegression	Size: 300	Epochs: 82	Accuracy: 0.8731
LogisticRegression	Size: 300	Epochs: 87	Accuracy: 0.8762
LogisticRegression	Size: 300	Epochs: 92	Accuracy: 0.8793
LogisticRegression	Size: 300	Epochs: 97	Accuracy: 0.8793

LogisticRegression	Size: 350	Epochs: 2	Accuracy: 0.8452
LogisticRegression	Size: 350	Epochs: 7	Accuracy: 0.8638
LogisticRegression	Size: 350	Epochs: 12	Accuracy: 0.8669
LogisticRegression	Size: 350	Epochs: 17	Accuracy: 0.8638
LogisticRegression	Size: 350	Epochs: 22	Accuracy: 0.8700
LogisticRegression	Size: 350	Epochs: 27	Accuracy: 0.8731
LogisticRegression	Size: 350	Epochs: 32	Accuracy: 0.8700
LogisticRegression	Size: 350	Epochs: 37	Accuracy: 0.8700
LogisticRegression	Size: 350	Epochs: 42	Accuracy: 0.8731
LogisticRegression	Size: 350	Epochs: 47	Accuracy: 0.8731
LogisticRegression	Size: 350	Epochs: 52	Accuracy: 0.8731
LogisticRegression	Size: 350	Epochs: 57	Accuracy: 0.8731
LogisticRegression	Size: 350	Epochs: 62	Accuracy: 0.8700
LogisticRegression	Size: 350	Epochs: 67	Accuracy: 0.8700
LogisticRegression	Size: 350	Epochs: 72	Accuracy: 0.8669
LogisticRegression	Size: 350	Epochs: 77	Accuracy: 0.8669
LogisticRegression	Size: 350	Epochs: 82	Accuracy: 0.8700
LogisticRegression	Size: 350	Epochs: 87	Accuracy: 0.8700
LogisticRegression	Size: 350	Epochs: 92	Accuracy: 0.8700
LogisticRegression	Size: 350	Epochs: 97	Accuracy: 0.8700
LogisticRegression	Size: 400	Epochs: 2	Accuracy: 0.8607
LogisticRegression	Size: 400	Epochs: 7	Accuracy: 0.8669
LogisticRegression	Size: 400	Epochs: 12	Accuracy: 0.8669
LogisticRegression	Size: 400	Epochs: 17	Accuracy: 0.8793
LogisticRegression	Size: 400	Epochs: 22	Accuracy: 0.8762
LogisticRegression	Size: 400	Epochs: 27	Accuracy: 0.8762
LogisticRegression	Size: 400	Epochs: 32	Accuracy: 0.8762
LogisticRegression	Size: 400	Epochs: 37	Accuracy: 0.8762
LogisticRegression	Size: 400	Epochs: 42	Accuracy: 0.8731
LogisticRegression	Size: 400	Epochs: 47	Accuracy: 0.8731
LogisticRegression	Size: 400	Epochs: 52	Accuracy: 0.8731
LogisticRegression	Size: 400	Epochs: 57	Accuracy: 0.8731
LogisticRegression	Size: 400	Epochs: 62	Accuracy: 0.8731
LogisticRegression	Size: 400	Epochs: 67	Accuracy: 0.8731
LogisticRegression	Size: 400	Epochs: 72	Accuracy: 0.8731
LogisticRegression	Size: 400	Epochs: 77	Accuracy: 0.8731
LogisticRegression	Size: 400	Epochs: 82	Accuracy: 0.8700
LogisticRegression	Size: 400	Epochs: 87	Accuracy: 0.8700
LogisticRegression	Size: 400	Epochs: 92	Accuracy: 0.8700
LogisticRegression	Size: 400	Epochs: 97	Accuracy: 0.8731
LogisticRegression	Size: 450	Epochs: 2	Accuracy: 0.8669
LogisticRegression	Size: 450	Epochs: 7	Accuracy: 0.8638
LogisticRegression	Size: 450	Epochs: 12	Accuracy: 0.8762
LogisticRegression	Size: 450	Epochs: 17	Accuracy: 0.8762
LogisticRegression	Size: 450	Epochs: 22	Accuracy: 0.8762
LogisticRegression	Size: 450	Epochs: 27	Accuracy: 0.8731
LogisticRegression	Size: 450	Epochs: 32	Accuracy: 0.8731
LogisticRegression	Size: 450	Epochs: 37	Accuracy: 0.8700
LogisticRegression	Size: 450	Epochs: 42	Accuracy: 0.8731
LogisticRegression	Size: 450	Epochs: 47	Accuracy: 0.8700
LogisticRegression	Size: 450	Epochs: 52	Accuracy: 0.8700
LogisticRegression	Size: 450	Epochs: 57	Accuracy: 0.8700
LogisticRegression	Size: 450	Epochs: 62	Accuracy: 0.8700
LogisticRegression	Size: 450	Epochs: 67	Accuracy: 0.8700
LogisticRegression	Size: 450	Epochs: 72	Accuracy: 0.8700
LogisticRegression	Size: 450	Epochs: 77	Accuracy: 0.8731
LogisticRegression	Size: 450	Epochs: 82	Accuracy: 0.8762
LogisticRegression	Size: 450	Epochs: 87	Accuracy: 0.8762
LogisticRegression	Size: 450	Epochs: 92	Accuracy: 0.8793
LogisticRegression	Size: 450	Epochs: 97	Accuracy: 0.8824

LogisticRegression	Size: 500	Epochs: 2	Accuracy: 0.8638
LogisticRegression	Size: 500	Epochs: 7	Accuracy: 0.8638
LogisticRegression	Size: 500	Epochs: 12	Accuracy: 0.8731
LogisticRegression	Size: 500	Epochs: 17	Accuracy: 0.8731
LogisticRegression	Size: 500	Epochs: 22	Accuracy: 0.8700
LogisticRegression	Size: 500	Epochs: 27	Accuracy: 0.8700
LogisticRegression	Size: 500	Epochs: 32	Accuracy: 0.8700
LogisticRegression	Size: 500	Epochs: 37	Accuracy: 0.8669
LogisticRegression	Size: 500	Epochs: 42	Accuracy: 0.8669
LogisticRegression	Size: 500	Epochs: 47	Accuracy: 0.8669
LogisticRegression	Size: 500	Epochs: 52	Accuracy: 0.8669
LogisticRegression	Size: 500	Epochs: 57	Accuracy: 0.8700
LogisticRegression	Size: 500	Epochs: 62	Accuracy: 0.8700
LogisticRegression	Size: 500	Epochs: 67	Accuracy: 0.8731
LogisticRegression	Size: 500	Epochs: 72	Accuracy: 0.8731
LogisticRegression	Size: 500	Epochs: 77	Accuracy: 0.8793
LogisticRegression	Size: 500	Epochs: 82	Accuracy: 0.8762
LogisticRegression	Size: 500	Epochs: 87	Accuracy: 0.8762
LogisticRegression	Size: 500	Epochs: 92	Accuracy: 0.8762
LogisticRegression	Size: 500	Epochs: 97	Accuracy: 0.8824
LogisticRegression	Size: 550	Epochs: 2	Accuracy: 0.8638
LogisticRegression	Size: 550	Epochs: 7	Accuracy: 0.8638
LogisticRegression	Size: 550	Epochs: 12	Accuracy: 0.8762
LogisticRegression	Size: 550	Epochs: 17	Accuracy: 0.8731
LogisticRegression	Size: 550	Epochs: 22	Accuracy: 0.8700
LogisticRegression	Size: 550	Epochs: 27	Accuracy: 0.8731
LogisticRegression	Size: 550	Epochs: 32	Accuracy: 0.8700
LogisticRegression	Size: 550	Epochs: 37	Accuracy: 0.8700
LogisticRegression	Size: 550	Epochs: 42	Accuracy: 0.8700
LogisticRegression	Size: 550	Epochs: 47	Accuracy: 0.8700
LogisticRegression	Size: 550	Epochs: 52	Accuracy: 0.8731
LogisticRegression	Size: 550	Epochs: 57	Accuracy: 0.8762
LogisticRegression	Size: 550	Epochs: 62	Accuracy: 0.8824
LogisticRegression	Size: 550	Epochs: 67	Accuracy: 0.8793
LogisticRegression	Size: 550	Epochs: 72	Accuracy: 0.8793
LogisticRegression	Size: 550	Epochs: 77	Accuracy: 0.8793
LogisticRegression	Size: 550	Epochs: 82	Accuracy: 0.8824
LogisticRegression	Size: 550	Epochs: 87	Accuracy: 0.8762
LogisticRegression	Size: 550	Epochs: 92	Accuracy: 0.8762
LogisticRegression	Size: 550	Epochs: 97	Accuracy: 0.8731
LogisticRegression	Size: 600	Epochs: 2	Accuracy: 0.8700
LogisticRegression	Size: 600	Epochs: 7	Accuracy: 0.8638
LogisticRegression	Size: 600	Epochs: 12	Accuracy: 0.8762
LogisticRegression	Size: 600	Epochs: 17	Accuracy: 0.8731
LogisticRegression	Size: 600	Epochs: 22	Accuracy: 0.8731
LogisticRegression	Size: 600	Epochs: 27	Accuracy: 0.8731
LogisticRegression	Size: 600	Epochs: 32	Accuracy: 0.8700
LogisticRegression	Size: 600	Epochs: 37	Accuracy: 0.8700
LogisticRegression	Size: 600	Epochs: 42	Accuracy: 0.8700
LogisticRegression	Size: 600	Epochs: 47	Accuracy: 0.8700
LogisticRegression	Size: 600	Epochs: 52	Accuracy: 0.8700
LogisticRegression	Size: 600	Epochs: 57	Accuracy: 0.8731
LogisticRegression	Size: 600	Epochs: 62	Accuracy: 0.8793
LogisticRegression	Size: 600	Epochs: 67	Accuracy: 0.8824
LogisticRegression	Size: 600	Epochs: 72	Accuracy: 0.8793
LogisticRegression	Size: 600	Epochs: 77	Accuracy: 0.8793
LogisticRegression	Size: 600	Epochs: 82	Accuracy: 0.8793
LogisticRegression	Size: 600	Epochs: 87	Accuracy: 0.8824
LogisticRegression	Size: 600	Epochs: 92	Accuracy: 0.8793
LogisticRegression	Size: 600	Epochs: 97	Accuracy: 0.8793

LogisticRegression	Size: 650	Epochs: 2	Accuracy: 0.8638
LogisticRegression	Size: 650	Epochs: 7	Accuracy: 0.8669
LogisticRegression	Size: 650	Epochs: 12	Accuracy: 0.8731
LogisticRegression	Size: 650	Epochs: 17	Accuracy: 0.8700
LogisticRegression	Size: 650	Epochs: 22	Accuracy: 0.8700
LogisticRegression	Size: 650	Epochs: 27	Accuracy: 0.8700
LogisticRegression	Size: 650	Epochs: 32	Accuracy: 0.8669
LogisticRegression	Size: 650	Epochs: 37	Accuracy: 0.8669
LogisticRegression	Size: 650	Epochs: 42	Accuracy: 0.8700
LogisticRegression	Size: 650	Epochs: 47	Accuracy: 0.8731
LogisticRegression	Size: 650	Epochs: 52	Accuracy: 0.8731
LogisticRegression	Size: 650	Epochs: 57	Accuracy: 0.8731
LogisticRegression	Size: 650	Epochs: 62	Accuracy: 0.8762
LogisticRegression	Size: 650	Epochs: 67	Accuracy: 0.8793
LogisticRegression	Size: 650	Epochs: 72	Accuracy: 0.8793
LogisticRegression	Size: 650	Epochs: 77	Accuracy: 0.8762
LogisticRegression	Size: 650	Epochs: 82	Accuracy: 0.8762
LogisticRegression	Size: 650	Epochs: 87	Accuracy: 0.8793
LogisticRegression	Size: 650	Epochs: 92	Accuracy: 0.8793
LogisticRegression	Size: 650	Epochs: 97	Accuracy: 0.8762
LogisticRegression	Size: 700	Epochs: 2	Accuracy: 0.8638
LogisticRegression	Size: 700	Epochs: 7	Accuracy: 0.8638
LogisticRegression	Size: 700	Epochs: 12	Accuracy: 0.8700
LogisticRegression	Size: 700	Epochs: 17	Accuracy: 0.8731
LogisticRegression	Size: 700	Epochs: 22	Accuracy: 0.8669
LogisticRegression	Size: 700	Epochs: 27	Accuracy: 0.8700
LogisticRegression	Size: 700	Epochs: 32	Accuracy: 0.8669
LogisticRegression	Size: 700	Epochs: 37	Accuracy: 0.8669
LogisticRegression	Size: 700	Epochs: 42	Accuracy: 0.8700
LogisticRegression	Size: 700	Epochs: 47	Accuracy: 0.8731
LogisticRegression	Size: 700	Epochs: 52	Accuracy: 0.8731
LogisticRegression	Size: 700	Epochs: 57	Accuracy: 0.8762
LogisticRegression	Size: 700	Epochs: 62	Accuracy: 0.8793
LogisticRegression	Size: 700	Epochs: 67	Accuracy: 0.8793
LogisticRegression	Size: 700	Epochs: 72	Accuracy: 0.8793
LogisticRegression	Size: 700	Epochs: 77	Accuracy: 0.8762
LogisticRegression	Size: 700	Epochs: 82	Accuracy: 0.8762
LogisticRegression	Size: 700	Epochs: 87	Accuracy: 0.8793
LogisticRegression	Size: 700	Epochs: 92	Accuracy: 0.8793
LogisticRegression	Size: 700	Epochs: 97	Accuracy: 0.8762
LogisticRegression	Size: 750	Epochs: 2	Accuracy: 0.8700
LogisticRegression	Size: 750	Epochs: 7	Accuracy: 0.8638
LogisticRegression	Size: 750	Epochs: 12	Accuracy: 0.8669
LogisticRegression	Size: 750	Epochs: 17	Accuracy: 0.8700
LogisticRegression	Size: 750	Epochs: 22	Accuracy: 0.8700
LogisticRegression	Size: 750	Epochs: 27	Accuracy: 0.8669
LogisticRegression	Size: 750	Epochs: 32	Accuracy: 0.8700
LogisticRegression	Size: 750	Epochs: 37	Accuracy: 0.8700
LogisticRegression	Size: 750	Epochs: 42	Accuracy: 0.8669
LogisticRegression	Size: 750	Epochs: 47	Accuracy: 0.8731
LogisticRegression	Size: 750	Epochs: 52	Accuracy: 0.8731
LogisticRegression	Size: 750	Epochs: 57	Accuracy: 0.8731
LogisticRegression	Size: 750	Epochs: 62	Accuracy: 0.8731
LogisticRegression	Size: 750	Epochs: 67	Accuracy: 0.8731
LogisticRegression	Size: 750	Epochs: 72	Accuracy: 0.8731
LogisticRegression	Size: 750	Epochs: 77	Accuracy: 0.8731
LogisticRegression	Size: 750	Epochs: 82	Accuracy: 0.8731
LogisticRegression	Size: 750	Epochs: 87	Accuracy: 0.8731
LogisticRegression	Size: 750	Epochs: 92	Accuracy: 0.8762
LogisticRegression	Size: 750	Epochs: 97	Accuracy: 0.8762

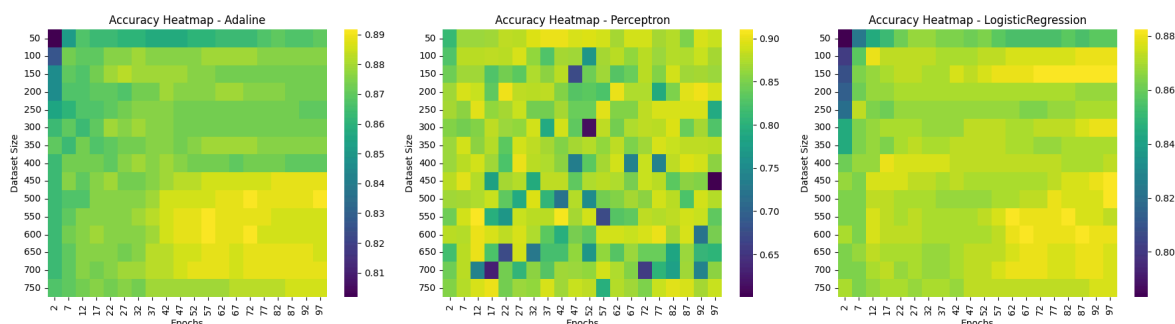
Performance visualization

Plot the performance measure for all classifiers (accuracy on the test set; use the result array from above) of all the 280 variants for each classifier in a total of three heatmaps using, for example `seaborn` or `matplotlib` directly.

The color should represent the accuracy on the test set, and the x and y axes should represent the number of epochs and the dataset size, respectively. Which one is x and which one is y is up to you to decide. Look in the example output at the top of the assignment for inspiration for how the plot could look like and how it could be labeled nicely. (But use the correct numbers corresponding to your dataset sizes and number of epochs.)

```
In [31]: # Insert your code below
# =====

fig, axes = plt.subplots(1, 3, figsize=(18, 5))
for idx, clf_name in enumerate(classifiers.keys()):
    sns.heatmap(accuracies[idx], ax=axes[idx], cmap="viridis", xticklabels=epoch
    axes[idx].set_xlabel("Epochs")
    axes[idx].set_ylabel("Dataset Size")
    axes[idx].set_title(f"Accuracy Heatmap - {clf_name}")
plt.tight_layout()
plt.show()
```



Part V: Some more plotting

For the following cell to execute you need to have the variable `X_test_scaled` with all samples of the test set and the variable `y_test` with the corresponding labels. Complete at least up until Part III. Executing the cell will plot something.

1. Add code comments explaining what the lines are doing
2. What is the purpose of the plot?
3. Describe all components of the subplot and then comment in general on the entire plot. What does it show? What does it not show?

```
In [32]: # Train and a Logistic regression model with 300 epochs and Learning rate 0.0001
clf = LogisticRegression(eta = 0.0001, epochs = 300, minibatches=1, random_seed=
clf.fit(X_test_scaled, y_test)

# Making an 8x8 grid with subplots with a figure size 30 height and 30 Length
```

```

fig, axes = plt.subplots(8, 8, figsize=(30, 30))

# Loop over all of features
for i in range(0, 8):
    for j in range(0, 8):
        feature_1 = i      # choosing the first feature
        feature_2 = j      # choosing the second feature
        ax = axes[i, j]    # getting the subplot

        ax.set_xlabel(f"Feature {feature_1}")      # labeling the x-axis to fea
        ax.set_ylabel(f"Feature {feature_2}")      # labeling the y-axis to fea

        # Getting min and max values for both features
        mins = X_test_scaled.min(axis=0)
        maxs = X_test_scaled.max(axis=0)

        # Generating 100 points for feature dimensions
        x0 = np.linspace(mins[feature_1], maxs[feature_1], 100)
        x1 = np.linspace(mins[feature_2], maxs[feature_2], 100)

        # Making a mesh grid using the features
        X0, X1 = np.meshgrid(x0, x1)
        X_two_features = np.c_[X0.ravel(), X1.ravel()]      # making the grid in

        # Creating a new dataset where features are set to zero except for the t
        X_plot = np.zeros(shape=(X_two_features.shape[0], X_test_scaled.shape[1])
        X_plot[:, feature_1] = X_two_features[:, 0]      # assigning first choose
        X_plot[:, feature_2] = X_two_features[:, 1]      # assigning second choos

        # Predicting probabilities for the grid points
        y_pred = clf.predict_proba(X_plot)
        Z = y_pred.reshape(X0.shape)      # reshaping predictions to be more L

        # Making a color plot for the decision boundary
        ax.pcolor(X0, X1, Z)
        ax.contour(X0, X1, Z, levels=[0.5], colors='k') # drawing a contour line

        # making scatter plots
        ax.scatter(X_test_scaled[y_test == 0, feature_1], X_test_scaled[y_test ==
        ax.scatter(X_test_scaled[y_test == 1, feature_1], X_test_scaled[y_test ==

# Fixing and showing plots
fig.tight_layout()
plt.show()

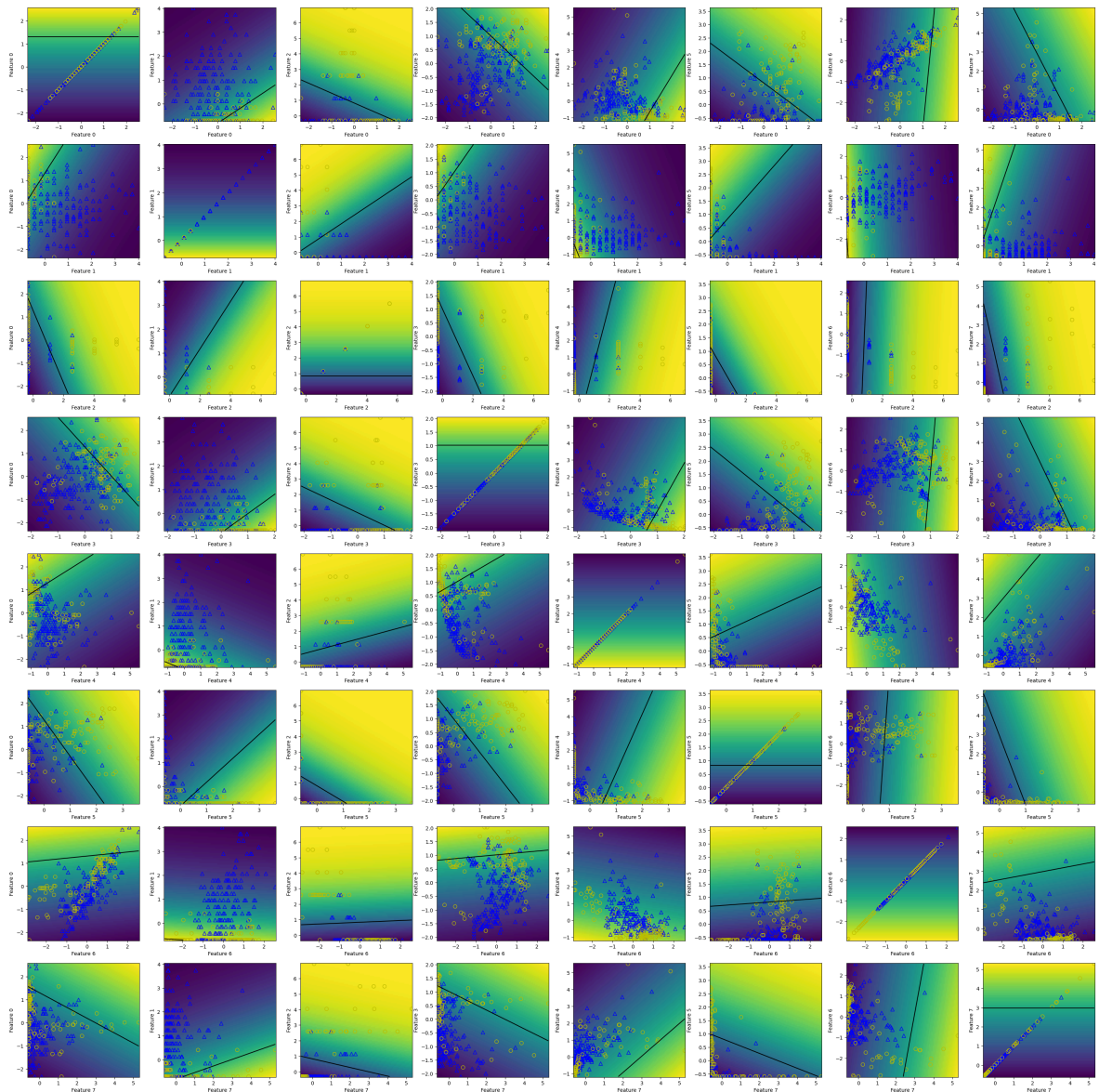
# 2
"""
The purpose of the plot is to visualize the decision boundaries of the logistic
on the testdata for all possible pairs,
and show how it separates the two classes 0 and 1 for fetal health.
"""

# 3
"""
The background color shows how confident the model is.
The black line is the decision boundary.
The blue and yellow points are test data.
The plot in general shows how good the logistic regression model separates
the two classes with the pair of features.

```

However, it does not show us the accuracy and the classifiers or which feature is the most important feature.

""



Out[32]: '\n\nThe background color shows how confident the model is.\n\nThe black line is the decision boundary. \n\nThe blue and yellow points are test data.\n\nThe plot in general shows how good the logistic regression model separates \n\nthe two classes with the pair of features.\n\nHowever, it does not show us the accuracy and the classifiers or \n\nwhich feature is the most important feature.\n'

Part VI: Additional discussion

Part I:

1. What kind of plots did you use to visualize the raw data, and why did you choose these types of plots?

Part II:

1. What happens if we don't shuffle the training data before training the classifiers like in Part IV?
2. How could you do the same train/test split (Point 1.-4.) using scikit-learn?

Part IV:

1. How does increasing the dataset size affect the performance of the logistic regression model? Provide a summary of your findings.
2. Describe the relationship between the number of epochs and model accuracy
3. Which classifier is much slower to train and why do you think that is?
4. One classifier shows strong fluctuations in accuracy for different dataset sizes and number of epochs. Which one is it and why do you think this happens?

Answers: Additional discussion

Part I:

1.
 - We used histograms to analyze the distribution of numerical features like baseline value and accelerations. This helps us understand the spread of these features.
 - Violin plots were used to compare the distributions of different features across the target classes. These plots show both the density and the other statistical components like the mean, making it easier to analyse or find differences in the features.
 - Scatter plots were used to check for linear separability between pairs of features, which is useful for seeing the effectiveness of linear classifiers like logistic regression.

Part II:

1. If we don't shuffle the training data then it is more likely that the model will learn from only one class at the start. This will lead to poor results.
2. We could use the `train_test_split` function from `sklearn.model_selection`

Part IV

1.
 - If the dataset size increases, then we will have more training data and therefore our model's accuracy improves.
 - On the other side, too much data can make the model more generalizing. Even though too little data makes the model overfitted.
2. Too many epochs makes the model overfit because it memorizes the training data without learning useful patterns. However, it improves accuracy.
3. The Adeline classifier is the slowest to train, because it updates weights for every sample. This makes the training computation heavier.

4. The Perceptron classifier shows stronger fluctuations in accuracy. Perceptron updates weights only when a misclassification happens, and therefore more sensitive to small changes.