Kasper Søballe Poulsen, 23T
Machine Learning
30/03/2025

Mandatory Assignment. Spring 2025.

Exercise 1

A)

I have chosen to use the python code from Andreas C. Mueller (moons_dataset_nn.py). Beneath is a schema showing the results of my adjustments to the number of hidden layers and number of neurons as well as how the alpha parameter can impact the results.

Hidden layers			Accuracy					
			alpha = 0,0001		alpha = 0,01		alpha = 1	
Layer 1	Layer 2	Layer 3	Training data	Test data	Training data	Test data	Training data	Test data
3	0	0	89%	84%	83%	92%	88%	80%
8	0	0	87%	92%	88%	94%	89%	84%
3	3	0	89%	92%	93%	88%	89%	76%
8	8	0	85%	84%	100%	96%	87%	84%
3	3	3	87%	88%	91%	84%	92%	72%
8	8	8	91%	88%	97%	96%	91%	80%
10	10	10	100%	96%	100%	100%	91%	84%

After reviewing the result it is clear that using multiple layers with a good amount of neurons leads to higher acurracy. For example having only one layer with 3 neurons is not enough for the model to correctly categorize data points. However having 3 layers of 10 neurons makes the model able to classify data points much more accurately.

Regarding the alpha parameter, 0.01 appears to be the best choice as it minimizes the variation between training and test accuracy, leading to better generalization. With alpha = 0.0001, the model might be overfitting and when using alpha = 1, the model might be underfitting. Both overfitting and underfitting will have a negative impact on the accuracy of the test data.

My recommendation is to use 3 hidden layers where each layer contains 10 neurons and to have an alpha value of 0,01 because this leads to the highest accuracy for both training data and test data.

B)

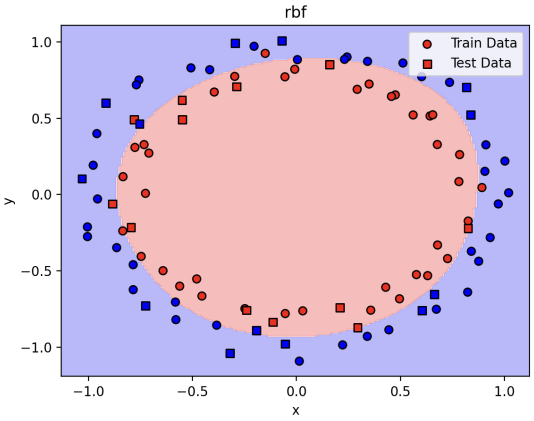
Kernel	Training data accuracy	Testing data accuracy
rbf	93%	88%
poly	60%	52%
linear	57%	36%
sigmoid	53%	44%

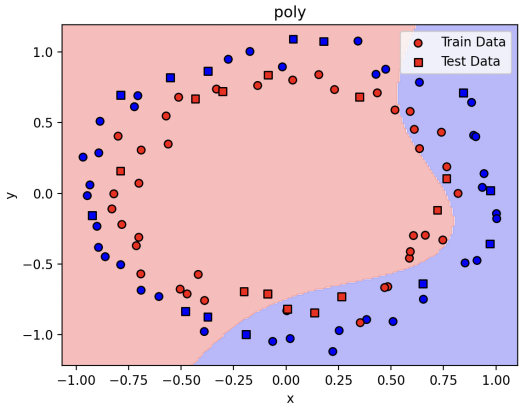
Using an rbf kernel gives us the best generalization model, according to the results of accuracy for testing data, with 88% correctly classified data points.

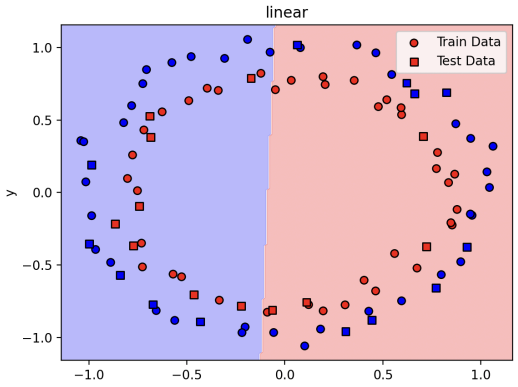
The other kernels all perform poorly. The poly kernel is not able to separate the data, even though this kernel sometimes works on nonlinear data. However, the complex circular pattern in this dataset seems too difficult for the polynomial kernel to model effectively.

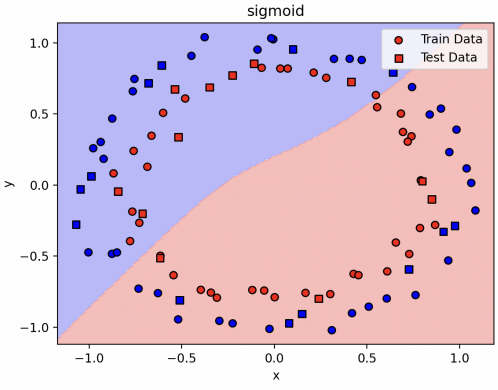
A linear kernel assumes that our data points are linearly separable, which they are not since they form concentric circles. As a result, the linear model fails to classify the data correctly, leading to very low accuracy.

The sigmoid kernel also performs poorly, as it is not well suited for this type of problem. It's better suited for data that can be separated using an "s" shape.









Kasper Søballe Poulsen, 23T
Machine Learning
30/03/2025

Exercise 2

A)

In this exercise we have a classification problem since we want to predict whether a passenger died or not. Since the predictions only have two possible outcomes this is called binary classification. We are going to train our model using supervised learning because we have a csv file containing features for passengers we know for sure died or survived.

```python
import pandas as pd
df = pd.read_csv("titanic_800.csv")
print(df.columns)
print(f"Number of columns: {len(df.columns)}")
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
Number of columns: 12
```

As shown in the picture above there are 12 columns in total. 11 of these columns are our features and the last column (Survived column) is our Y-data.

B)

I investigated the data using the count() and describe() methods for dataframes.
I chose to remove the Cabin column since this value was only known for 185 of the 800 passengers which I don't believe is enough data to use Cabin as a reliable feature for our model.
There was only 1 passenger row that didn't include an Embarked value so I decided to remove this passenger row since it would have no significant impact on our amount of data.
About 20% of the passenger rows were missing values for the Age column. Therefore I decided to insert the average age for these rows in order to maintain a good amount of training and test data.
I converted the values in the Sex column into numerical values (0 for men and 1 for women).
I removed the Name column because it's different for almost every person and doesn't help the model predict survival.
I used a scatter plot or a bar plot function to check all columns that have not been removed if they have a relation to the survival rate. I used this visualisation to decide whether or not to remove the specific column.
I turned categorical variables into multiple dummy variables to make the data more suitable for machine learning algorithms.
Lastly, I split the data into a training dataset and a test dataset, and applied a scaler that was fitted only on the training data. I used 80% for training and 20% for testing, because I want an accurate model but also a sufficient amount of testing data.

C)

I chose to use both a neural network and a random forest model because I wanted to compare these two models. The neural network is good at finding complex patterns in the data, especially after scaling. I also used a random forest because it performs well with small datasets like this one.

D)

```
Classification Report NN:
              precision    recall

           0       0.80      0.89
           1       0.82      0.69
```

```
Classification Report Random Forest:
              precision    recall  f

           0       0.81      0.85
           1       0.78      0.74
```

Neural network:

Precision: Of all predicted survivors, 82% actually survived.
Recall: Of all actual survivors, 69% were correctly predicted.

Random forest:

Precision: Of all predicted survivors, 78% actually survived.
Recall: Of all actual survivors, 74% were correctly predicted.

```
Confusion Matrix NN:
[[82 10]
 [21 47]]
```

```
Confusion Matrix Random Forest:
[[78 14]
 [18 50]]
```

Neural network:

Accuracy: (82 + 47) / (82 + 10 + 21 + 47) = 0,80625
Best at: Predicting passengers who did not survive (True negatives)
Worst at: It missed 21 actual survivors (False negatives)

Random forest:

Accuracy: (78 + 50) / (78 + 14 + 18 + 50) = 0,8
Best at: Predicting passengers who did not survive (True negatives)
Worst at: It missed 18 actual survivors (False negatives)

E)

1. `mlp = MLPClassifier(hidden_layer_sizes=(10, 10, 10), max_iter=500, random_state=42)`
2. `mlp = MLPClassifier(hidden_layer_sizes=(50, 50, 50), max_iter=1000, random_state=42)`

Line 1 gives a better accuracy (80%) than line 2 (77%). This might be due to overfitting.

Kasper Søballe Poulsen, 23T
Machine Learning
30/03/2025

Exercise 3

A)

The iris dataset contains 4 features which means it is a 4 dimensional dataset. 4 dimensions is very difficult to visualize so to solve this problem we use PCA. PCA is able to convert our 4 dimensional data into 2 dimensional data so that we can easily plot it.
The 2d scatter plot shows three clear groupings and when using KMeans with k=3 to identify these 3 clusters it also aligns pretty well with the real classifications. This suggests that three natural clusters can reasonably be found in the dimensionality-reduced Iris dataset.

B)

PCA is even more useful with larger datasets containing many dimensions because it figures out which of the many dimensions are most important and uses these to make principal components. When we get rid of these less important dimensions we get faster training time for our models since there is less data to process.
There was no significant improvement of the accuracy when setting c=10000. This only led to a worse recall for the classes. This is probably because a lower c value gives a better generalization model while a high C makes it more focused on perfectly fitting the training data.
Adjusting the gamma from 0,001 to 0,01 and 0,00001 both led to a significant decrease in accuracy which might be due to the model either overfitting (gamma = 0.01) or underfitting (gamma = 0.00001).
The PCA dimensionality reduction is necessary because it makes training the model faster and prevents the model from overfitting because of the otherwise many dimensions.
Other classifiers like KNN, Random Forest, or Neural Network could also be tested to see if they can outperform the SVM model.
If we are only looking for one specific person we would label all persons in the dataset with the number 0 if they are not the correct person or 1 if they are the correct person. That way we have a binary classification problem that can be solved with any binary classification model.