Report of the 1st Project

Submission by 21st November 2023

1 Introduction

In this assignment, we trained and evaluated our first models with basic classification methods, namely the **kNN Classifier** and **Logistic Regression**. We also examined the importance of the **train-test-split** in model evaluation and analyzed the effect of **unbalanced classes** on our classifiers.

2 Data Sets

For our experiments, we used three datasets in total.
Dataset 1, the first dataset we trained on, is a balanced dataset with two classes, 0 and 1. Each class has 400 samples. The input has two features. The shape of the dataset can be described as a spiral with the inner part being mostly class 1 while the outer part is mostly class 0.
Dataset 2 is also balanced, has two classes named 0 and 1 with 400 samples each and two-dimensional input vectors. The shape of this dataset, however, is different from Dataset 1. All the datapoints are much closer to each other and the shape of each class can be described as a crescent. The classes are clustered in close proximity of each other, with the crescents interlinking in the middle.
Dataset 3, which we examined for the last part of our project, is an unbalanced dataset with two classes, 0 and 1. Out of 1000 samples, 85% belong to class 1 and 15% to class 0. The samples from class 1 are distributed like a crescent again, with the class 0 samples clustered in the shape of an ellipsis at the hollow center of the crescent.

	Dataset 1	Dataset 2	Dataset 3
Mean of Features	(3.071, 0.425)	(0.215, 0.242)	(5.450, 5.687)
Standard Deviation of Features	(6.364, 6.794)	(0.893, 0.604)	(7.419, 7.433)
Variance of Features	(40.505, 46.159)	(0.798, 0.365)	(55.040, 55.243)

Table 1: Statistical Information about the Datasets' Features

3 Foundations

3.1 Methods

3.1.1 kNN Classifier

The kNN Classifier is a very simple classifier that simply stores all instances of training data and computes the class of a new datapoint via a majority vote. The model is non-parametric, with only one hyper-parameter named k which indicates the number of neighbours used for classification and can be determined via hyperparameter-tuning. The classifier works by computing the distances of a query point to all training points, sorting by those distances to determine the k nearest neighbours and assigning the query point the majority class of those neighbours.

3.1.2 Logistic Regression

Logistic regression is a linear model for classification which computes the probabilities of possible classes using the logistic function. Optimization is done via **Gradient Descent** in our experiments, although the solver is a hyperparameter of the general method. Other hyperparamters are the penalty, which describes the norm used during penalization (e.g. L1 or L2) and C, which denotes the inverse of the regularization strength.

To allow the linear classifier to perform better on non-linear data, it is possible to compute **polynomial features** from the input. This additional **preprocessing step** has the degree of the polynomial as its main hyperparameter.

3.2 Evaluation

For our actual evaluation, we only use one measure: **Accuracy**. It is simply the percentage of correctly classified datapoints out of all datapoints. We examine both training and test accuracy. Cross-validation is introduced but not consistently applied.

4 Experiments

4.1 Set-Up

Our experiments were extremely limited in their design as we simply trained the classifiers on the datasets and then analyzed the accuracy and some hyperparameters (i.e. k for kNN and the degree of the polynomial for polynomial preprocessing).

4.2 Analysis and Results

Overall, we found that the kNN classifier yields good results for both Dataset 1 and Dataset 2, both much better than the majority class accuracy of 0.5. Our analysis showed that an ideal k for Dataset 1 is k = 6. We obtained this result by analyzing the effect of all possible k - from 1 to the number of datapoints in the training set - on the classification accuracy on Dataset 1 4.2.

Linear Logistic Regression yields good results for Dataset 2 but quite bad results for Dataset 1, particularly compared with kNN. 4.2 This can be explained by the linear decision boundary seen in 4.2, which is unable to seperate the datapoints of the classes as the dataset itself is highly non-linear in its composition. Applying polynomial preprocessing before the actual Logistic Regression step yields much better results 4.2. The ideal polynomial degree for Dataset 1 is somewhere between three and five, with every degree greater than one leading to an improvement, although extremely high polynomial degrees can be prone to overfitting. Which exact degree is best depends on the data.

In general, it can be said that both kNN and Logistic Regression obtain good results on both datasets, with only Linear Regression on Dataset 1 yielding particularly bad results due to the non-linear nature of the data. For both datasets, kNN is a good choice, although Polynomial Regression with degree 3 performs a little better on Dataset 1. For both classifiers and and datasets, the training accuracy was slightly higher than the test accuracy.

Classification on Dataset 3 obtains also good results in terms of accuracy, with kNN outperforming Logistic Regression 4.2, however, further analysis shows that samples from class 1 are classified more accurately than those from class 0 for both classifiers, with a particularly significant difference in accuracy observeable for Logistic Regression.4.2

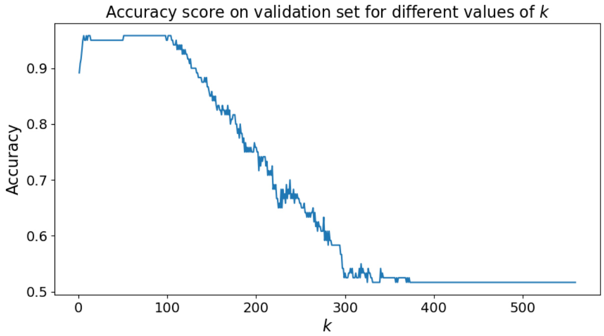


Figure 1: Accuracy on Dataset 1 for different k in kNN

Data	kNN	Linear Regression	Polinomial Regression
Dataset 1	0.908	0.55	0.933
Dataset 2	0.929	0.925	0.925
Dataset 3	0.985	0.865	-

Table 2: Test Accuracies of our Methods

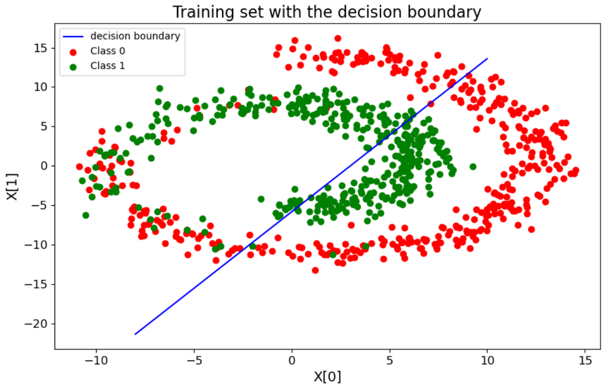


Figure 2: Training Set (Dataset 1) with Decision Boundary for Logistic Regression

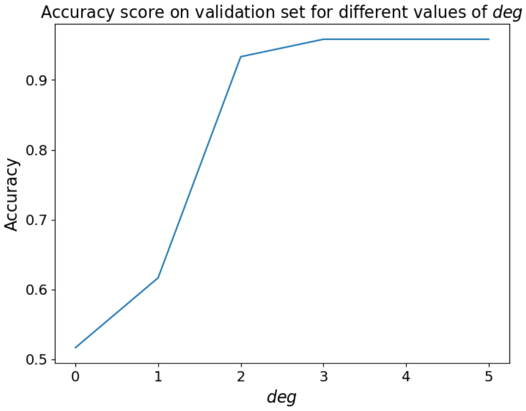
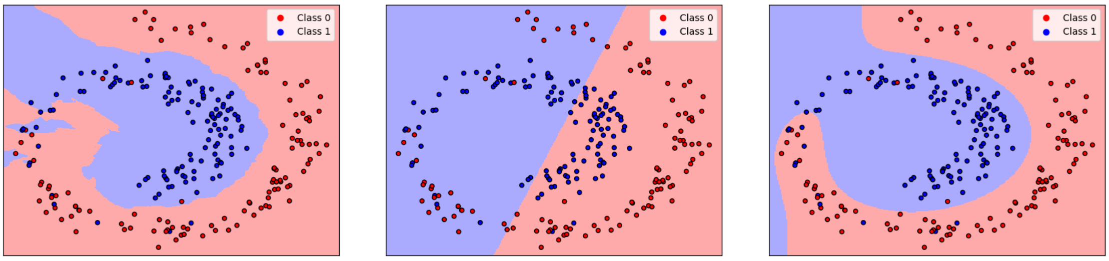


Figure 3: Accuracy on Dataset 1 for different polynomial degrees

5 Discussion

During this project, we applied standard methods of Classification to small, simple datasets with two classes and two features. The results we obtained were, as expected, reasonably good.
Apart from the results we already analyzed, we have also discovered that it is necessary to use the same test-train-split for all methods to obtain compareable results. To get particularly reliable results, it is ideal to perfrom corss-validation. This means that the dataset is initially partitioned into an arbitrary number n (e.g. 5) of splits of equal size. Each split gets chosen as the test set in turn, with all other making up the training set. The scores are then averaged over all n training rounds.
Furthermore, we reason that the accuracy is only a good score for balanced datasets. Unblanaced datasets require a more throrough analysis, using measures such as the F1-score, computed out of precision and recall. Plotting the confusion matrix is also a good idea for unbalanced datasets in particular.
We are exited to apply those findings in our future projects.

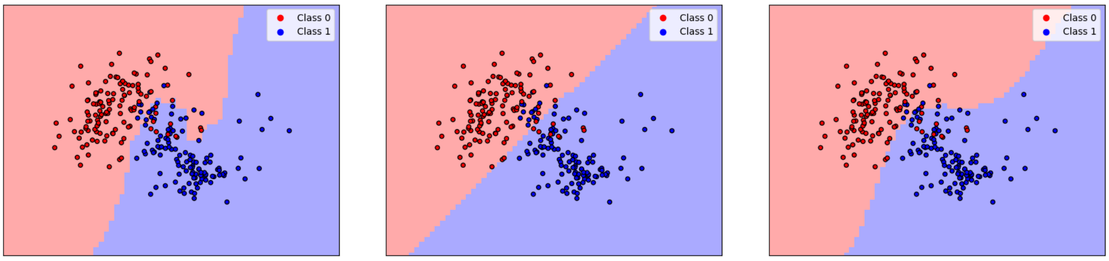


(a) kNN

(b) Linear Regression

(c) Logistic Regression with Polynomial Preprocessing

Figure 4: Results for Dataset 1 - Test Set



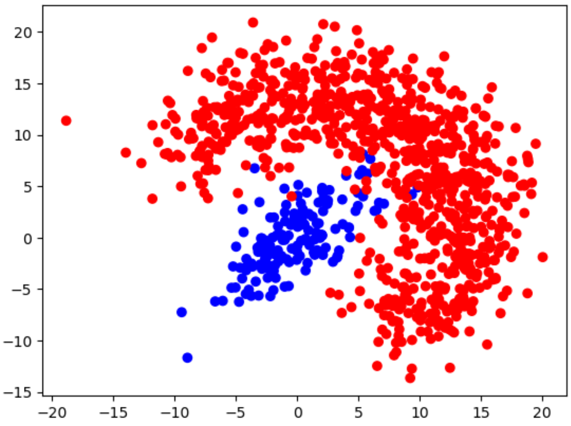
(a) kNN

(b) Linear Regression

(c) Logistic Regression with Polynomial Preprocessing

Figure 5: Results for Dataset 2 - Test Set

Figure 6: Dataset 3



Class	kNN	Logistic Regression
Class 0	0.93	0.54
Class 1	0.99	0.97

Table 3: Class-wise Accuracy for Dataset 3