Artificial Intelligence Assessment Document

Iris & digits recognition classification

Daniel Tang

2022

Contents

[Explain the problem(s), and an overview of the solution. 2](#_Toc103950316)

[Explain the initial data framework, mathematical algorithms, models, theories, tools and techniques that will contribute to the solution. Provide any sketches/diagrams/calculations that might have used to support your ideas. 2](#_Toc103950317)

[Explain how you retrieved/created/sourced the dataset and any loading/cleaning that you had to undertake to prepare it. 3](#_Toc103950318)

[Show what you did to explore the data using the appropriate libraries. Explain any created definitions and summaries, visualisations or transformations of the data. Outline any necessary tests on the variables; explain the results and how these were be used. 3](#_Toc103950319)

[Iris Dataset processing 3](#_Toc103950320)

[Digits Recognition dataset processing 9](#_Toc103950321)

[Explain and describe the details of the model(s) you built and implemented. 10](#_Toc103950322)

[Explain your training and/or testing strategy. How you deployed it, explain your results and any tuning of the model(s) that you did. 11](#_Toc103950323)

[Explain your evaluation methods/metrics and how you evaluated its accuracy. 12](#_Toc103950324)

[Explain any estimators/error function/loss functions/optimisation methods you used to help to select the best (final) model. 12](#_Toc103950325)

# Explain the problem(s), and an overview of the solution.

There are multiple problems that I would like to solve, one of them is a personal problem such as understanding how classification algorithms work, understanding which classification algorithm can be used and which is the best algorithm to solve different problems. Following this personal problem is how we could implement and use different classification algorithms and which algorithms are suitable or optimised for different projects. The end goal for this problem is to:

* Understand how classification algorithms works
* Which is the best algorithm to solve a similar problem but a different dataset.

Three problems I want to explore are understanding which algorithms are suitable for two classification problems one is understanding the iris flower to distinguish the species of iris and comparing the model to understand how classification requires more data to enable itself to produce more accurate corrections.

Another problem I want to explore is the digit recognition, using machine learning to aid and possibly use these algorithms to automate bank cheques processing for the bank, the problem of this would be low accuracy would render the automation process useless and unreliable which is not recommended to be used in the real-world environment. The final problem is to understand the simplistic neural network for digit recognition.

The plan for iris and digit recognition is to implement algorithms such as Support Vector Machines (SVM), Gaussian Naïve Bayes (GNB), Decision Tree (DT) and K-Nearest Neighbour (K-NN) which four algorithms will be used to analyse the accuracy and the time it takes to process iris and digit predictions.

# Explain the initial data framework, mathematical algorithms, models, theories, tools and techniques that will contribute to the solution. Provide any sketches/diagrams/calculations that might have used to support your ideas.

SVM or Support Vector Machine is a linear model for classification and regression problems, which can solve linear and non-linear problems and work with practical problems. SVM is simple to implement, and the algorithms create a line or a hyperplane which separates the data into classes.

Another algorithm that considers is the Gaussian Naïve Bayes or known as Naïve Bayes or NB for this project, Naive Bayes is a classification technique that is based on Bayes’ theorem, in which Naïve Bayes classifiers assume that the presence of a particular feature in a class is unrelated to the presence of any other features.

A decision tree is part of a supervised learning algorithm, however to other supervised learning algorithms the decision tree can be used for solving regression and classicality problems which is useful to calculate all possible outcome for a problem and require less data cleaning compared to another algorithm.

k-NN is used in a variety of machine learning tasks; for example, in [computer vision](https://brilliant.org/wiki/computer-vision/), k-NN can help identify handwritten letters and in [gene expression](https://brilliant.org/wiki/gene-expression/?wiki_title=gene%20expression) analysis, the algorithm is used to determine which genes contribute to a certain characteristic. Overall, k-nearest neighbours provide a combination of simplicity and effectiveness that makes it an attractive algorithm to use for many machines learning tasks.

# Explain how you retrieved/created/sourced the dataset and any loading/cleaning that you had to undertake to prepare it.

Dataset for Iris was received from UCI Machine Learning Repository which can be found here: <https://archive.ics.uci.edu/ml/datasets/iris> for the iris dataset.

The dataset will be imported from Sklearn’s dataset (<https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html>) which has 10 different classes (digits 0 through 9), with 64 features and 1797 sample data. The Sklearn library includes a lot of tools for machine learning and statistical modelling, which is useful for this problem to classify, not only to classify but also to conduct regression clustering and dimensionality reductions. For both datasets, loading and cleaning data will describe in the heading section below.

# Show what you did to explore the data using the appropriate libraries. Explain any created definitions and summaries, visualisations or transformations of the data. Outline any necessary tests on the variables; explain the results and how these were be used.

Libraries used for Iris and digitsdataset:

Numpy, pandas, seaborn, matplotlib, time and sklearn (metrics, accuracy\_score, train\_\_test\_split, and algorthms.

## Iris Dataset processing

The iris dataset for will both have a dataset consisting of physical features between all three species of flower, versicolor, setosa and virginica. In which the dataset contains numerical values such as the Sepal width, sepal length, petal width and petal length will use to predict which species of plants.

First, we need to understand the data, we need to load the iris.csv to make it readable and understand the dataset.

Graphical user interface, application

Description automatically generated

Figure 1:Attribute of the DataFrame

The dataset is from iris.csv which contains, 150 samples, 3 labels: specifies of iris example Setosa, Virginica and Versicolor. It also contains four unique features, they are Sepal length, sepal width, sepal width, petal length, and petal width in centimetres, however, the fifth feature is the species name. We can see this in figure 2 where we load the CSV.

Text

Description automatically generated

Figure 2 Iris head data

The data need to be to understand which is currently in a mess and need to be defined, to do it need to create column names and define the column names.

Graphical user interface, text

Description automatically generated

Figure 3: Iris data defined

Figure 3 shows the column names of each feature to organize the data. Next step is to check the statistics of the dataset such as the mode, median, mean and standard deviation, to do this we need to use the iris.describe(). However, we need to verify the features which datatypes which will use iris.info() function to see any data type from the dataset.

Graphical user interface, table

Description automatically generated

Figure 4: .describe and .info function

After describing and looking at the boring numbers of stats, we can visualize the data to understand it more better using matplotlib to visualize the data. Visualizing species with features using sns.pairplot to use to analyse relationship between different species and 4 different features, this pairplot is useful to allow us to visualize all of the features from the dataset into one image.

A picture containing diagram

Description automatically generated

Figure 5: sns.pairplot

Next visual will use heatmap to see which features are correlated. Figure 6 shows the heatmap that you can see the petal length, and petal width there is a correlation with these two features.

A picture containing Teams

Description automatically generated

Figure 6: Heatmap

Final visualization for the dataset is the violin plot, which shows the density of the length width of each species.

Chart

Description automatically generated

Figure 7: Violin plot

This concludes with the visualization and data preparation for both Iris dataset A and B, include check if there were any null datatypes.

## Digits Recognition dataset processing

The digits data contains 64 features with 1797 sample data. However, the digits has been already processed into 8\*8 format/

Graphical user interface, text, application

Description automatically generated

Figure : Data sample

An 8\*8 image which the array corresponds to the pixel, the 0 represents a black pixel and the higher the number the less black on the pixel. You can see the text compared to the image.

A picture containing graphical user interface

Description automatically generated

Figure : 8\*8 image array

To visualize the digits from the dataset, the dataset will be flattered into 2D dimensions and turn it into greyscale which is important due to simplifying the algorithm and reducing computational resources.

# Explain and describe the details of the model(s) you built and implemented.

For iris and digits classification model will use multi-class classification, we use this classification unlike binary because it does not have abnormal outcomes. The class labels may be large, example model may predict photos as it belongs to thousands of faces in a face recognition system.

Models that will be implemented to check the accuracy of every algorithm, algorithms will use for the iris dataset will be:

* Support Vector Machine (SVM)
* Decision Tree (DT)
* K-Nearest Neighbour (KNN)
* Naïve Bayes (NB)

Overall, each algorithm will build using sklearn. Imports help us build the models for Iris and Digits.

Text

Description automatically generated

Figure 10: SVM Model for Iris dataset

Graphical user interface, text

Description automatically generated

Figure 11: Support Vector Classifier for Digits

# Explain your training and/or testing strategy. How you deployed it, explain your results and any tuning of the model(s) that you did.

For the training and testing strategy, the two-classification problem will be separated on how we train and test, and how the dataset will be split to train the model. For Iris and digits classification, to be on the safe side for training and testing data we will split the data into 75/25 because it is a good starting point.

# Explain your evaluation methods/metrics and how you evaluated its accuracy.

To get the accuracy of each algorithm, we will use the metrics.accuracy\_score function to get the accuracy from the prediction from X\_test and calculate it onto y\_test to calculate the accuracy, and calculate all the accuracy separately.

Table

Description automatically generated with medium confidence

Figure 12: Classification report of SVM for IRIS

Table

Description automatically generated

Figure : Classification report of SVM for Digits

# Explain any estimators/error function/loss functions/optimisation methods you used to help to select the best (final) model.

To optimize the iris dataset, I believe that we need to find the best performing algorithm and create it into a standalone project which includes defining the algorithm instead of using imports such as sklearn, and in the stand-alone project could use estimators or other functions to optimise the methods to create a optimise classification model for both Iris and Digits dataset.

For example, if KNN was the best performing algorithm I would just implement the KNN algorithm onto the iris data set and create the KNN algorithm from scratch such as defining and creating Euclidean Distance, finding the nearest neighbour, and counting the number of data points in each category and assign new data point to the neighbour to the maximum. For the digits I believe creating a neural network to predict the digits (which is currently under development).

# NOTES

Youtube Demo: <https://www.youtube.com/watch?v=3xM_MI5VcHs>

I’m believe that I may not be available until the end of June.