AI UX & Data Visualisation Design Principles (CA6002)

Associate Professor Goh Wooi Boon

College of Computing and Data Science
Nanyang Technological University

email: *aswbgoh@ntu.edu.sg*

1

Chapter 2.2 – Visualisation for AI

Contents

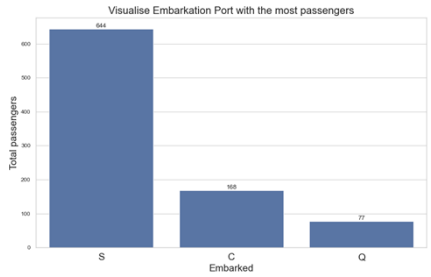
- Count plots - Counting and sorting
- Scatter plots - Visualising 2D relationships
- Parallel coordinates plots
  - Interactive PCP
  - Interpreting the PCP
- Visualising model performance
  - Confusion matrix
  - ROC curve
  - Precision-Recall curve
  - Learning curve

2

Count Plots

Counting and Sorting

- A common preliminary analysis of a dataset is to visualise the **quantity** of each category in a variable of interest.
- A useful plot for this analysis is a bar plot from Seaborn's `countplot`.
- For more effective visual analysis, the categories should be **sorted** from largest to the smallest and the actual **count values annotated** at the top of each bar.
- For example, in the "*Titanic.csv*" dataset, we may want to visualise the number of passengers from each port of embarkation.



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[1] Seaborn Countplot - https://seaborn.pydata.org/generated/seaborn.countplot.html

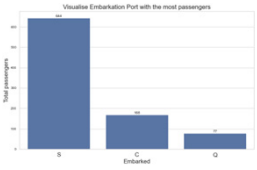
3

3

Count Plots

Counting and Sorting

- The segment of Python code to render this sorted and annotated count plot is given below.



```
: # read in Titanic dataset into dataframe df and do the necessary data wrangling
:
varX = 'Embarked'                                  # variable of interest
order = df[varX].value_counts().index              # order categories
ax = sns.countplot(x=varX, data=df, order=order)   # count plot

for container in ax.containers:
    ax.bar_label(container)                        # annotate each bar
:                                                  # title plot, label axes
```

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[1] Seaborn Countplot - https://seaborn.pydata.org/generated/seaborn.countplot.html

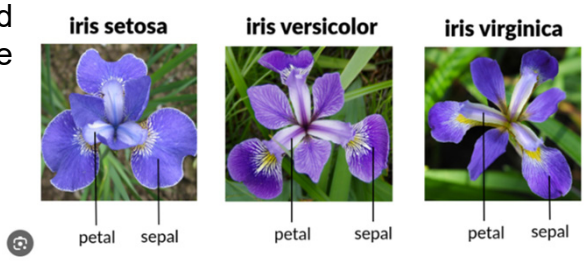
4

4

Scatter Plots

The Iris dataset

- A small and popular dataset that is commonly used in the field of machine learning and statistical analysis is the '*iris*' dataset.
- It was first introduced by Ronald Fisher in 1936 and is commonly used to distinguish between three different species of Iris flowers (Iris setosa, Iris virginica, and Iris versicolor).
- This dataset is widely used for training and testing classification algorithms in machine learning and statistical applications.
- Four **numeric** features are used in the classification of the Species, namely **Sepal Width**, **Sepal Length**, **Petal Width** and **Petal Length**.



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[2] Iris Species Dataset - https://www.kaggle.com/datasets/uciml/iris

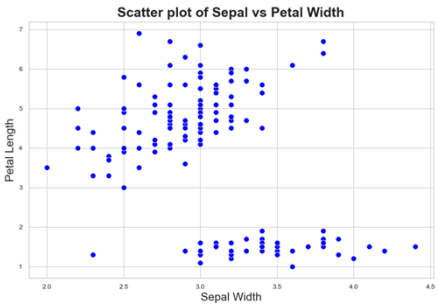
5

5

Scatter Plots

Visualising 2D Relationships

- A useful plot to visualise the relationship between two features for a given category of interest is the 2D scatter plot.
- The Seaborn's `scatterplot` is an easy to use function with some useful features
- The limitation of the 2D scatter plots is that they only allow the visualisation of relationship between **two variables** at a time.
- For example, in the "*iris_data.csv*" dataset, we may want to visualise the relationship between the **Sepal Width** and the **Petal Length**.



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

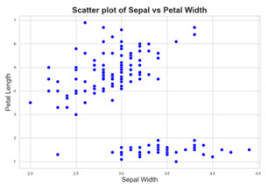
[3] Seaborn Scattertplot - https://seaborn.pydata.org/generated/seaborn.scatterplot.html

6

6

Scatter Plots

Visualising 2D Relationships



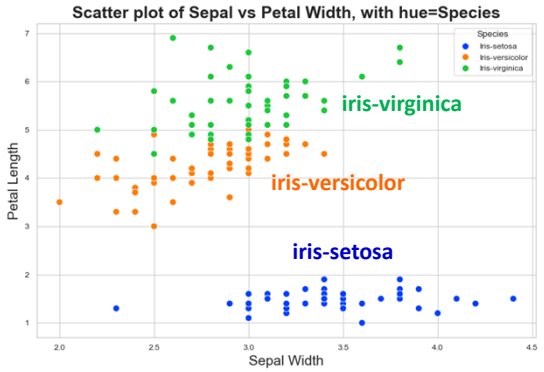
- The segment of Python code to render the basic scatter plot is given below.

```
:                                    # read in iris dataset
varX = 'Sepal Width'                 # dataframe variable on the x-axis
varY = 'Petal Length'                # dataframe variable on the y-axis
ax = sns.scatterplot(
          data=df,                   # dataframe
          x=varX, y=varY,            # assign dataframe columns to x and y
          s=50,                      # set size of dots
          color='blue'               # set colour of dots
          )
:                                    # title plot, label axes
```


Scatter Plots

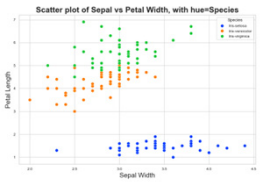
Visualising 2D Relationships

- The `hue` parameter in `scatterplot` can be used to see how these 2D relationships differ for each of the categories of the variable of interest.
- The `hue` parameter must be assigned to a categorical data type.
- For the iris example, we could use a different colour for each of the **Species**.
- The different **Species** colours shows that the **Sepal Length** and **Petal Width** relationship for **iris-setosa** is quite distinct from the other two and these features can be very useful in predicting the **setosa** class of iris.



Scatter Plots

Visualising 2D Relationships



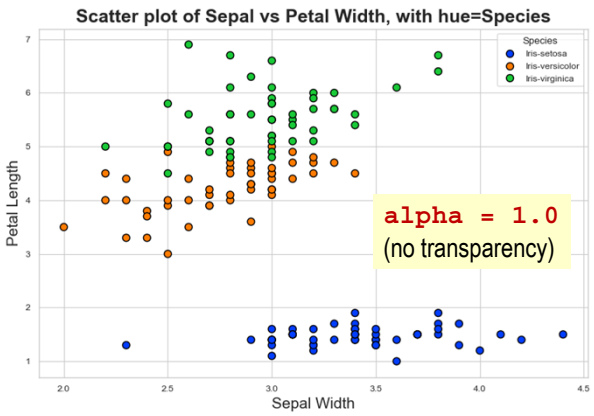
- The segment of Python code to render the scatter plot with hue is given below.

```
:
varX = 'Sepal Width'                # read in iris dataset
varY = 'Petal Length'               # dataframe variable on the x-axis
ax = sns.scatterplot(               # dataframe variable on the y-axis
          data=df,                  # dataframe
          x=varX, y=varY,           # assign dataframe columns to x and y
          hue='Species',            # assign hue to 'Species'
          palette='bright',         # choose a bright colour palette
          s=50,                     # set size of dots
          )
```


Scatter Plots

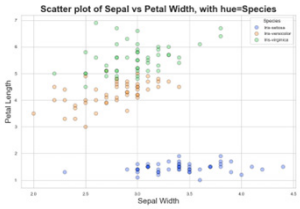
Visualising 2D Relationships

- Some additional enhancements for scatterplot includes **adding edges** to the dots to improve visual discrimination between dots that are adjacent and touching.
- The `alpha` transparency parameter can be used in situations where there are many **overlapping dots** that occlude each other.
- Transparency allows visualisation of areas in the 2D scatter plot with high **density of dots**, as occluding dots hide other dots beneath them, giving a false impression of dot count.



Scatter Plots

Visualising 2D Relationships



- Python code to render the scatter plot with dot edges and transparency:

```
:                                       # read in iris dataset
varX = 'Sepal Width'                    # dataframe variable on the x-axis
varY = 'Petal Length'                   # dataframe variable on the y-axis
ax = sns.scatterplot(
          data=df, x=varX, y=varY       # dataframe, assign x & y columns
          hue='Species',                # assign hue to 'Species'
          edgecolor='black',            # set edge colour to dots
          linewidth=1,                  # set thickness of edge
          alpha=0.3,                    # set alpha value of 0.3
          s=50 )                        # set size of dots
```

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[3] Seaborn Scattertplot - https://seaborn.pydata.org/generated/seaborn.scatterplot.html

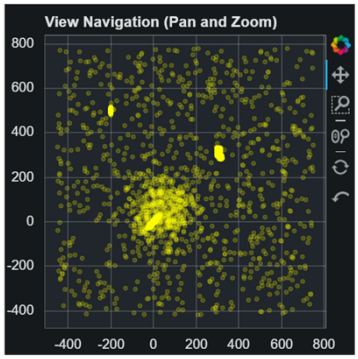
11

11

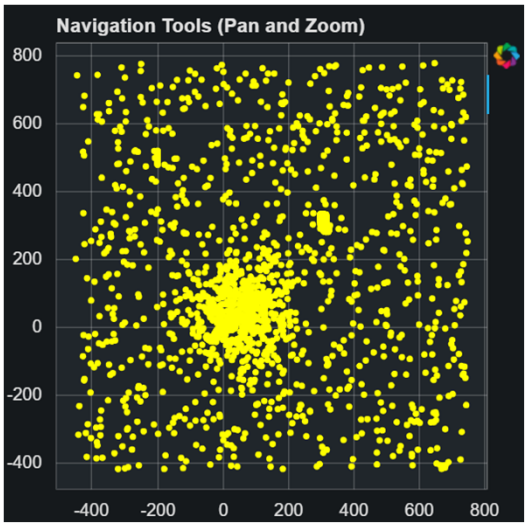
Scatter Plots

Transparency Control

- Controlling the transparency of the dots in a scatter plot (through the use of appropriate `alpha` values) can reveal the presence of hidden clusters.



alpha = 0.2

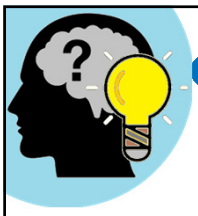


alpha = 1.0

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

12

12

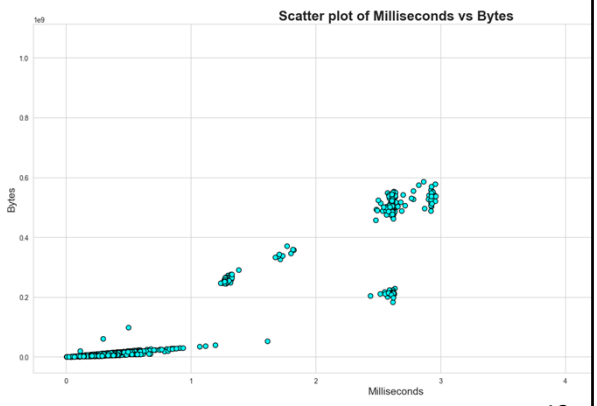


Think and Apply

iTunes

How to predict iTune track file sizes based on their play times?

- A dataset of 3503 iTune tracks with information on their file size, play duration, genre, album title, artist, etc.
- What is the relationship between play time (***milliseconds***) and file size (***bytes***)?
- Is this relationship consistent or is it influenced by other factor(s)?
- Design an accurate prediction algorithm to estimate the file size of the iTune track given its duration and other information.



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

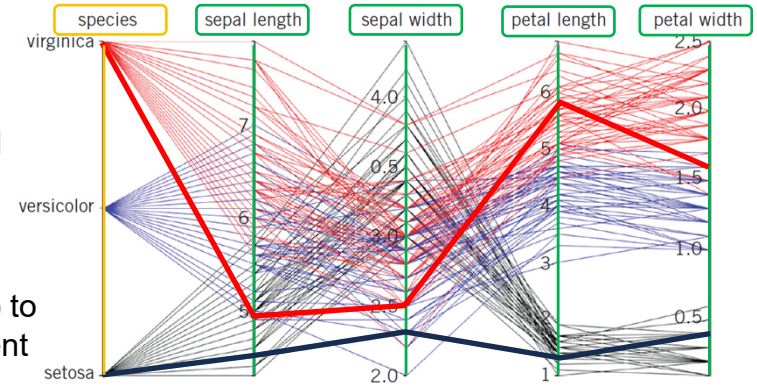
13

13

Parallel Coordinates Plot

What is a Parallel Coordinates Plot?

- The parallel coordinates plot (PCP) visualises relationships between **multiple variables** (**multivariate data**) in a dataset.
- Each **vertical axis** represents a **numeric** variable (**feature**) in the dataset.
- Each **data point** is represented by a **line** connecting values across all axes.
- A colourmap is assigned to the target feature (**colour variable**) to help distinguish between different target categories.



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[4] J. Heinrich & D. Weiskopf, Parallel Coordinates for Multidimensional Data Visualization: Basic Concepts, Computing in Sciences & Engineering, May/June 2015 - https://joules.de/files/heinrich_parallel_2015.pdf

14

14

Parallel Coordinates Plot

Python code segment to render a simple PCP using Plotly Express

```
import plotly.express as px
import plotly.io as pio
pio.renderers.default = 'browser'     # necessary for Spyder to open PCP in browser
:
mapping_dict = {                      # Convert categorical Species to numeric values
   'Iris-setosa': 1, 'Iris-versicolor': 2,'Iris-virginica': 3, }
df['Species'] = df['Species'].map(mapping_dict)

ax = px.parallel_coordinates(dfNorm,          # PCP using plotly express
       color='Species',                       # colour variable is Species
       color_continuous_scale=px.colors.diverging.Tealrose)  # colormap
ax.update_traces(                             # light colour for unselected lines
       unselected={'line':{'color': 'lightgray', 'opacity': 0.1}})
ax.show()
```

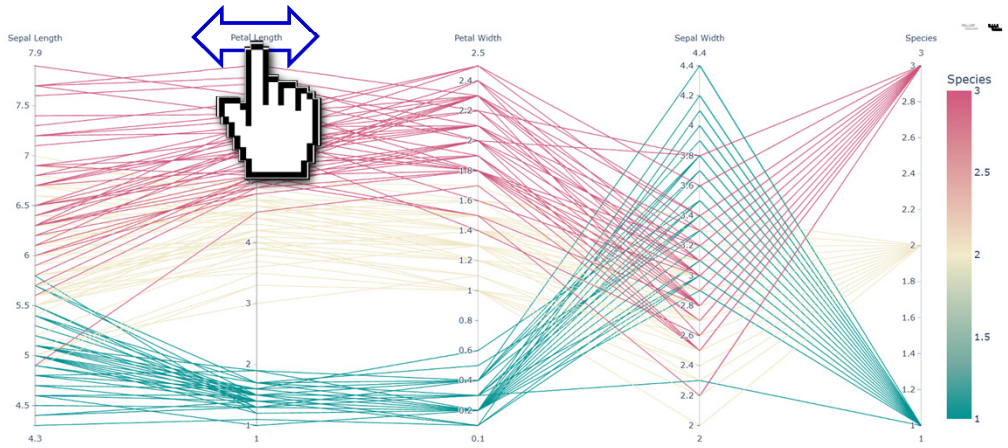
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE [5] Parallel Coordinates plot with Plotly Express - https://plotly.com/python/parallel-coordinates-plot/ 15

15

Parallel Coordinates Plot

Interactive PCP using Plotly Express

- The PCP has several interactive features. **Feature axes** can be **swapped** by dragging.



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE [5] Parallel Coordinates plot with Plotly Express - https://plotly.com/python/parallel-coordinates-plot/ 16

16

Parallel Coordinates Plot

Interactive PCP using Plotly Express

- **Brushing** – holding cursor down & sliding on an axis, selects subset of data points to highlight.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[5] Parallel Coordinates plot with Plotly Express - https://plotly.com/python/parallel-coordinates-plot/

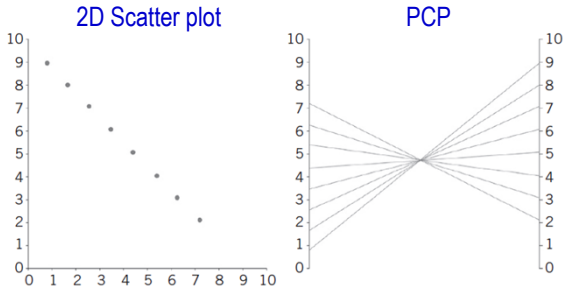
17

17

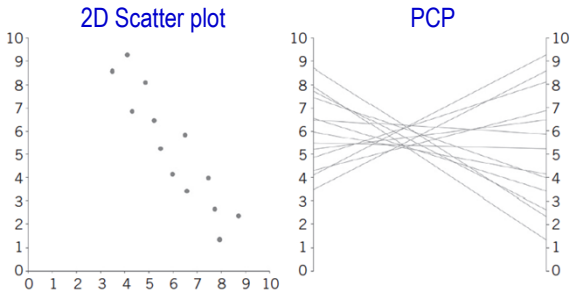
Parallel Coordinates Plot

Interpretting the Parallel Coordinates Plot

- Most real-world datasets results in fairly complex line patterns in the PCP. However, observing the PCP line patterns resulting from simple relationships between two features can help us interpreted the more complex and less structured ones.



Negative correlation (strong)



Negative correlation (moderate)

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[4] J. Heinrich & D. Weiskopf, Parallel Coordinates for Multidimensional Data Visualization: Basic Concepts, Computing in Sciences & Engineering, May/June 2015 - https://joules.de/files/heinrich_parallel_2015.pdf

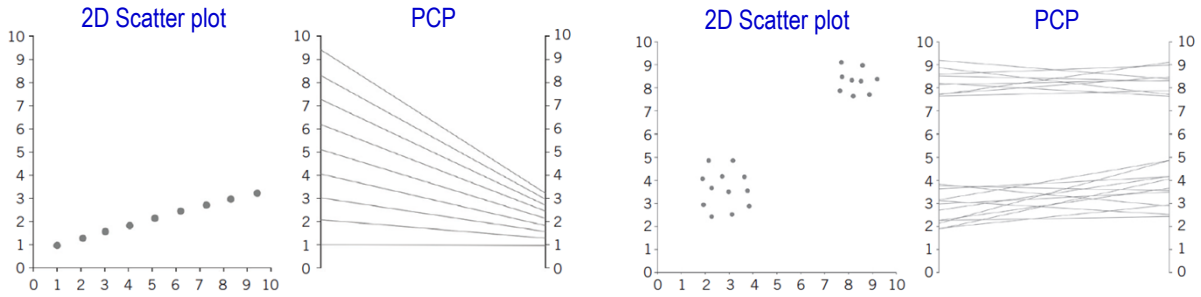
18

18

Parallel Coordinates Plot

Interpretting the Parallel Coordinates Plot

- Most real-world datasets results in fairly complex line patterns in the PCP. However, observing the PCP line patterns resulting from simple relationships between two features can help us interpreted the more complex and less structured ones.



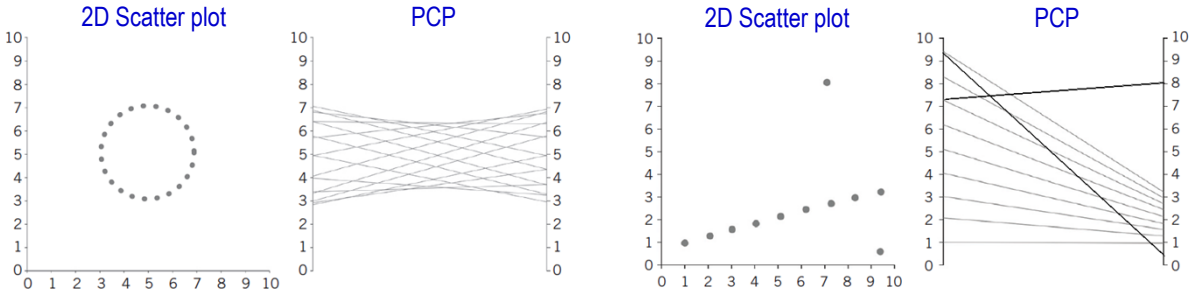
Positive correlation (strong) **Clearly segregated clusters**

[4] J. Heinrich & D. Weiskopf, Parallel Coordinates for Multidimensional Data Visualization: Basic Concepts, Computing in Sciences & Engineering, May/June 2015 - https://joules.de/files/heinrich_parallel_2015.pdf

Parallel Coordinates Plot

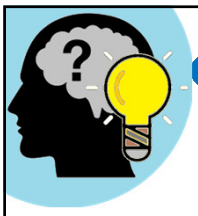
Interpretting the Parallel Coordinates Plot

- Most real-world datasets results in fairly complex line patterns in the PCP. However, observing the PCP line patterns resulting from simple relationships between two features can help us interpreted the more complex and less structured ones.



Circular 2D scatter pattern **Presence of outliers**

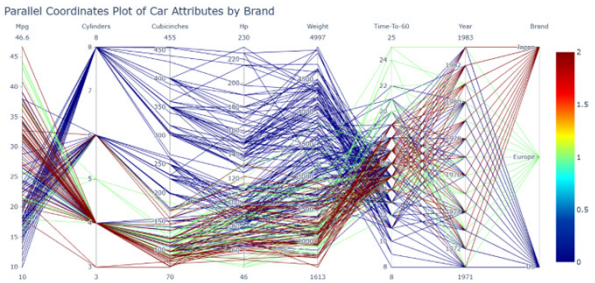
[4] J. Heinrich & D. Weiskopf, Parallel Coordinates for Multidimensional Data Visualization: Basic Concepts, Computing in Sciences & Engineering, May/June 2015 - https://joules.de/files/heinrich_parallel_2015.pdf



Think and Apply

cars.csv - Exploration with Parallel Coordinates Plot

- A dataset of 261 cars from US, Europe and Japan, with information on their horsepower, weight, engine capacity, fuel efficiency (mpg), etc.
- What are the relationships between the various car features? Correlated?
- What are the specification differences in car brand from different regions?
- Design an accurate classification algorithm to estimate which region the car is from given its feature values.



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

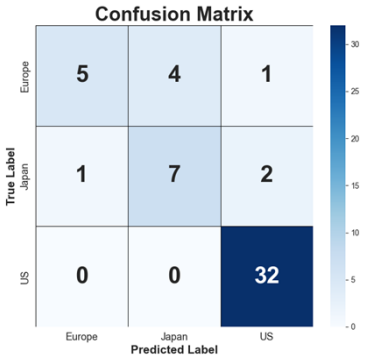
21

21

Visualising Model Performance

Confusion Matrix

- A confusion matrix (CM) is a table that visualizes a classification model's performance, comparing its **predicted** labels against the **actual** ground truth to show where it misclassify (i.e. got confused).
- Numeric values down the diagonal (i.e. top-left to bottom-right) show the **True Positive** (TP) and the off diagonal values show the different misclassification numbers.
- CM provides a tabulated means to quickly evaluate the model's accuracy, identify its strengths/weaknesses in classifying the different categories.
- CM can be visually enhanced with an appropriate **colourmap** as it is essentially a **heatmap** plot.



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[6] scikit-learn confusion_matrix - https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

22

22

Visualising Model Performance

ROC Curve

- A ROC (Receiver Operating Characteristic) curve is a graph showing a **binary** classifier's performance across all decision thresholds by plotting the True Positive rate (**TPR**) against the False Positive rate (**FPR**)
- The area under the ROC curve (**AUC score**) sums up how well a model can produce relative scores to discriminate between positive or negative instances across all classification thresholds.
- The ROC AUC score ranges from 0 to 1, where 0.5 indicates random guessing (i.e. the **red line** of chance), and 1 indicates perfect performance.

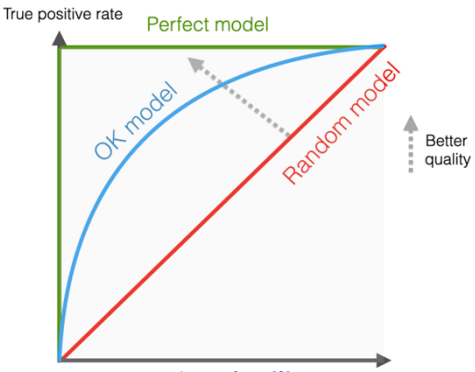


image from [8]

[7] scikit-learn, ROC curve - https://scikit-learn.org/1.1/auto_examples/model_selection/plot_roc.html
[8] How to explain the ROC curve and ROC AUC score? - https://www.evidentlyai.com/classification-metrics/explain-roc-curve

Visualising Model Performance

Precision-Recall Curve

- Precision-Recall (P-R) curve is a graph plotting a model's **Precision** (y-axis) against its **Recall** (x-axis) at various prediction threshold.
- The curve reveals the trade-off between correctly identifying positive cases (**Recall**) and ensuring those identified positives are truly positive (**Precision**).
- P-R curves can show the point when the **first misclassification** (i.e. false positive) starts to occur (i.e. the point the precision drops from 1.0).
- P-R curves are useful for evaluating binary classifiers on **imbalanced datasets** (e.g. fraud detection). In such cases, ROC curves might show overly optimistic results.

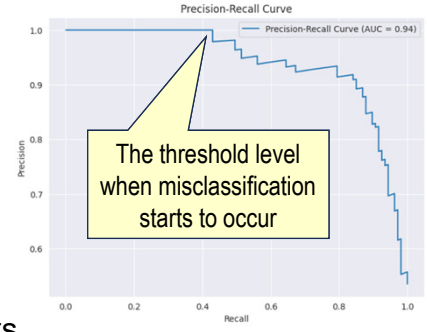


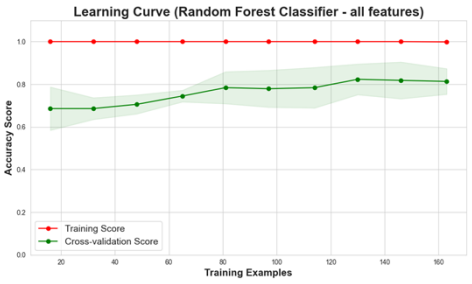
image from [8]

[9] scikit-learn, precision_recall_curve - https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html
[10] Precision-Recall Curve - ML - https://www.geeksforgeeks.org/machine-learning/precision-recall-curve-ml/

Visualising Model Performance

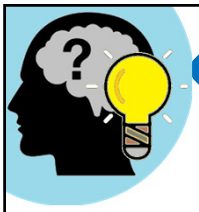
Learning Curve

- A learning curve (LC) is a plot showing a model's **performance** (e.g. accuracy) **as it trains**, typically against the amount of training data or iterations (epochs).
- LC is a useful diagnostic tool to understand if a model is underfitting (poor generalization) or overfitting (memorizing training data).
- Underfitting is seen when training and validation plateau out to low scores with more training data. A flattening curve indicates that adding more data will not improve model's performance.



- If there is a significant and **increasing gap** between the training and the validation performances, the model is **overfitting**.

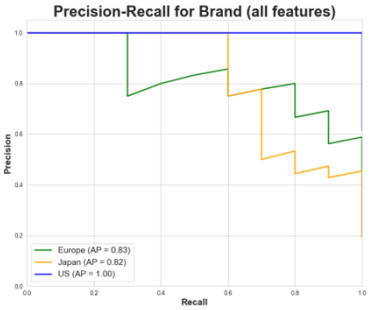
[11] scikit-learn, validation and learning curve - https://scikit-learn.org/stable/modules/learning_curve.html



Think and Apply

cars.csv – Visualising Model Performance

- There are numerous ways to visualise and evaluate the model's classification accuracy of cars into in '**brand**' categories from US, Europe and Japan.
- Does the confusion matrix tells us which category is easiest to classify? What colourmap is suitable?
- Precision-Recall versus ROC curves, when should we use which? What do they say about the impact of different features on the accuracy score?
- What can the Learning Curve tell us about the learning capacity of the model?



Summary

Visualisation for AI

- Seaborn's `countplot` can help us quickly check if there is an imbalance of data points for the different categories.
- Seaborn's `scatterplot` is useful for looking at the relationship between two features in a dataset. Its `alpha` transparency parameter helps overcome the problem of occlusion to find hidden clusters.
- Plotly Express's `parallel_coordinates` plot provides an interactive way to visualise relationships between multiple features in a dataset.
- Visualisation and evaluation of a model's accuracy and behaviour can be done through various visualisation tools like **confusion matrix** and plots like **ROC**, **Precision-Recall** and **Learning curves**.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

27

27

References for Visualisation for AI

[1] Seaborn Countplot - https://seaborn.pydata.org/generated/seaborn.countplot.html

[2] Iris Species Dataset - https://www.kaggle.com/datasets/uciml/iris

[3] Seaborn Scattertplot - https://seaborn.pydata.org/generated/seaborn.scatterplot.html

[4] J. Heinrich & D. Weiskopf, Parallel Coordinates for Multidimensional Data Visualization: Basic Concepts, Computing in Sciences & Engineering, May/June 2015 - https://joules.de/files/heinrich_parallel_2015.pdf

[5] Parallel Coordinates plot with Plotly Express - https://plotly.com/python/parallel-coordinates-plot/

[6] scikit-learn, confusion_matrix - https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

[7] scikit-learn, ROC curve - https://scikit-learn.org/1.1/auto_examples/model_selection/plot_roc.html

[8] How to explain the ROC curve and ROC AUC score? - https://www.evidentlyai.com/classification-metrics/explain-roc-curve

[9] scikit-learn, precision_recall_curve - https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html

[10] Precision-Recall Curve - ML - https://www.geeksforgeeks.org/machine-learning/precision-recall-curve-ml/

[11] scikit-learn, validation and learning curve - https://scikit-learn.org/stable/modules/learning_curve.html

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Note: All online articles were accessed and available on 17 Dec 2025

28

28