Prac W2 - Working with Data

COMP4702/COMP7703 - Machine Learning

Aims:

- To gain familiarity with loading datasets and suitable pre-processing procedures in either Python or Matlab.
- To develop practical exploratory data analysis and data visualisation skills, and to understand why they are important.
- To produce some assessable work for this subject.

Part I: Data Loading and Pre-processing:

For this first section, you will load and pre-process a dataset to transform it into a suitable format for running machine learning algorithms. Choose one or more of the datasets available on the course Blackboard (Learning Resources/Week 2/):

(**Q1**) Download one of the datasets listed above from the course blackboard page, and import it into either your Python environment or MATLAB. You can also just open the data in a spreadsheet if you want to look at it (MATLAB's Import Data button gives you a view like this).

Code 1: Python import data and print stats

```python
import pandas as pd
df = pd.read_csv("YOUR PATH HERE.csv")
print(f"{df.head()}\n")
print(f"{df.info()}\n")
print(f"{df.isnull().sum()}\n")
for column in df:
  print(f"{df[column].describe()}\n")
```

Code 2: MATLAB import data and print stats

```matlab
t = readtable('YOUR PATH HERE.csv');
summary(t);
sum(ismissing(t));
```

(a) Take a brief look at the data and note the data types of each feature (Continuous? Integer? Categorical? Non-numerical?). This should give you a general understanding of the data you are working with and how it may need to be transformed or pre-processed.

(b) Analyse some numerical statistics of each feature in the dataset, e.g. the number of null values, minimum/maximum values, frequency of classes or values, standard deviation. These will inform your pre-processing decisions such as filling in (imputing) missing values, removing outliers, or normalising features.

(**Q2**) Now we will pre-process the data! This will eliminate the missing values which are unsuitable for our machine learning algorithms. The following commands you may find useful for this task:

- **Python**: pandas.DataFrame.drop(), sklearn.impute.SimpleImputer(), sklearn.preprocessing.LabelEncoder()
- **MATLAB**: removevars(), deleterows(), fillmissing(), grp2idx()

(a) If there are missing values in your data, the simplest approach is to remove the rows or columns that contain missing values. Try removing all rows or all columns that contain missing data. Or, a combination of these, attempting to minimise the overall amount of data lost.

(b) Often imputing the data (replacing missing values) is a better approach. Replace your missing values with the mean of the feature if the feature is numerical, and the most frequent value if it is a categorical feature. Make a copy of the data before you do this for a later question.

(c) If your data has non-numerical features, try encoding a feature as an integer value.

Part II: Data Visualisation and Exploratory Data Analysis (EDA):

(**Q3**) Now it is time to visualise the data. Effective visualisation of the data helps when choosing a suitable machine learning algorithm, and informs decisions regarding more sophisticated pre-processing techniques such as feature selection and feature engineering. The following commands you may find useful for this task:

- **Python**: seaborn.pairplot(), seaborn.histplot()
- **MATLAB**: pairplot(), histogram()

There are plenty of nice libraries for doing data plots. Useful links:

- Plotly - a graphing library. Python: `https://plotly.com/python/`, MATLAB: `https://plotly.com/matlab/`
- MATLAB has a rich library of built in plotting methods: `https://www.mathworks.com/help/matlab/creating_plots/types-of-matlab-plots.html`
- Matplotlib: The most widely-used plotting library for Python. `https://matplotlib.org/stable/plot_types/index.html`

(a) Create a draftsman display (also called a pair plot) of your dataset and examine the correlation between features. Comment on the trends or notable dataset features you find.

(b) Plot the data (from Q2(b)) before and after doing imputation. Describe any visual differences that you can see.

(**Q4**) There are some fantastic data visualisations on the web. Find one that you think is particularly good and put a post on Ed about it (in a thread if other people have already posted some). In your post, comment on why you think this visualisation is good. What does it show?