UNIVERSITY OF CANBERRA
AUSTRALIA'S CAPITAL UNIVERSITY

Assignment Cover Sheet

Pattern Recognition & Machine Learning UG+PG 11482, 11512

You must keep a photocopy or electronic copy of your assignment.

Name of Tutor:	Dr Xing Wang			
Student ID:	U3284513	Unit/Subject Code:	11482	
Assignment No.:	1B	Number of pages: (including this cover sheet)		7
Date and Time Due:		29th of August, 23:59pm		
Date and Time Submitted:		10:00pm 31/08/2025		
Mark:				

I declare that this assignment is solely my own work, except where due acknowledgements are made. I acknowledge that the assessor of this assignment may provide a copy of this assignment to another member of the University, and/or to a plagiarism checking service whilst assessing this assignment. I have read and understood the University Policies in respect of Student Academic Honesty.

Date: 23/08/2025

INTRODUCTION

This problem explores the Fashion MNIST dataset which contains 70,000 gray-scale images in a 28 by 28-pixel grid, divided into ten different categories. The main objective of the problem is to observer whether a logistic regression model can accurately predict different clothing categories. Some of the key questions include but not limited to:

- Can the model achieve a good accuracy, i.e. more than 70% on the classification problem?
- How does L2 generalization affect the performance of the model?

THE DATASET

The Fashion MNIST dataset consists of 70,000 images divided into a training set of 60,000 images and a testing set of 10,000 images, with each image formatted as a 28 by 28 black and white pixel grid. The integers from 0 to 9 corresponds to the following label: 0 - T-shirt/top, 1 - Trouser, 2 – Pullover, 3 – Dress, 4 – Coat, 5 – Sandal, 6 – Shirt, 7 – Sneaker, 8 – Bag, 9 - Ankle boot. Some important characteristics of this data set include:

- The pixel values range from 0 to 255.
- The dataset is balance, meaning each category have the same number of samples.
- Visual similarities between some categories such as T-shirt/top and Shirt might lower the accuracy of the model.

LOGISTIC REGRESSION

A logistic regression is a linear classification algorithm used to predict the probability of whether an input belongs to a specific class (Geeksforgeeks, 2017). There are certain advantages and disadvantages to this algorithm:

Advantages:

- Easy to implement, interpret and efficient.
- Can handle multiple class and make no assumption about classes (Geeksforgeeks, 2020).
- Can efficiently handle moderate dataset such as the Fashion MNIST.

Disadvantages:

- It has a linear boundary which may fail to recognize complex patterns.
- It assumes linearity between the dependent and independent variables.
- It cannot solve non-linear problems.

DATA EXPLORATION

First, we import the necessary libraries for the problem:

```
import numpy as np
import matplotlib.pyplot as plt
import joblib
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import KFold, cross_val_score
from keras.datasets import fashion_mnist
```

To load the dataset, we use the following snippet:

```
from keras.datasets import fashion_mnist
# Loading the data
(X_train, y_train), (X_test, y_test) = fashion_mnist.load_data()
```

This returns the training and testing images – label pair and allow us to begin working on the dataset. To visualize some sample images, we use the following snippet:

```
plt.figure(figsize=(10, 4))
for i in range(10):
    plt.subplot(2, 5, i + 1)
    plt.imshow(X_train[i], cmap='gray')
    plt.title(className[y_train[i]])
    plt.axis('off')
```

This returns the first 10 sample images from the dataset. We can observe that the pixels weight ranges from 0 to 255; Classes are visually distinct, but some classes are hard to separate as seen in Figure 1.

We begin by transforming the 0 – 255 value range of the pixel into a 0 – 1 value range, which the sigmoid function uses to make predictions. To do this, we take the pixel values and divide them by 255, which is the highest number:

```
X_train_flat =X_train.reshape(X_train.shape[0], -1).astype('float32') / 255.0
X_test_flat = X_test.reshape(X_test.shape[0], -1).astype('float32') / 255.0
```

We pass the following parameters into the regression model:

```
modelL2 = LogisticRegression (penalty='l2', max_iter=500, C=1.0, random_state=42)
```

L2 regularization is used because it pushes the feature weights towards 0 and penalize large values. The parameter “c” controls the strength of regularization A higher “c” score means weaker regularization, while the inverse is true. Adjusting the “c” score might change the output of the program, but it does not mean that the model is more accurate in predicting the values. Our target variables are y_test and y_train, with the featured variables being the flattened and normalized 28 x 28 images. We can begin training the model and making predictions:

```
# Fitting and predicitng
modelL2.fit(X_train_flat, y_train)
```

```
y_pred = modelL2.predict(X_test_flat)
```

To measure the accuracy of the model, we use the following metrics – accuracy score, confusion matrix, classification report and K-Fold cross validation with 5 folds:

```
accuracy = modelL2.score(X_test_flat, y_test)
print(f"Model accuracy: {accuracy:.4f}")
cm = confusion_matrix(y_test, y_pred)
print("\nConfusion matrix: \n", cm)
print("\nClassification report: \n", classification_report(y_test, y_pred))
cvScore = cross_val_score(modelL2, X_test_flat, y_test, scoring="accuracy",
n_jobs=-1)
print(f"\nCross validations score on 5 fold: {cvScore.mean():.4f} +/-
{cvScore.std():.4f}")
```

The accuracy of the model yields an 84.32% accurate prediction with a K-Fold score of 82.02. The confusion matrix in Figure 2 shows us the comparison between the model's prediction output against the true label. We can see the first 10 correct predictions and incorrect predictions in Figure 4 and Figure 5. A likely reason for the incorrect predictions observed seems to be the similarity between the different classes, such as ankle boots and sandals.

To export the model and train it on unseen data, we use the following snippet:

```
# Exporting the model
joblib.dump(modelL2, "FashionMNIST.pkl")
exportedModel = joblib.load("FashionMNIST.pkl")
y_pred_exported = exportedModel.predict(X_test_flat)
print(y_pred_exported[:10])
```

This will export the model into a .pkl file, which we can then load and use it to make predictions on new values.

CONCLUSION

The model achieved an accuracy of 84.28%, which is good for a simple linear model. The use of regularization has improved the K-Fold validation score by a small margin, meaning the model is more accurate with regularization. The incorrect predictions is a direct result of the disadvantage of the model, which is its inability to recognize complex patterns within a complicated dataset.

REFERENCES:

- GeeksforGeeks. “Logistic Regression in Machine Learning.” *GeeksforGeeks*, 9 May 2017, www.geeksforgeeks.org/machine-learning/understanding-logistic-regression/
- GeeksforGeeks. “Advantages and Disadvantages of Logistic Regression.” GeeksforGeeks, 25 Aug. 2020, www.geeksforgeeks.org/data-science/advantages-and-disadvantages-of-logistic-regression/

APPENDICES:

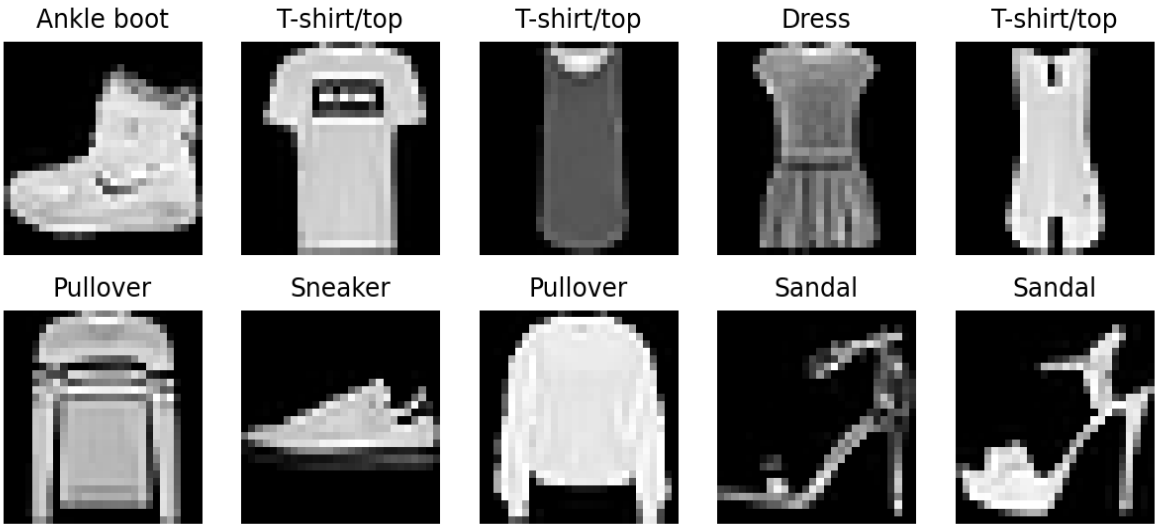

Figure 1: First 10 samples.

```
Confusion matrix:
 [[806   2  12  53   4   2 110   0  11   0]
 [  4 958   3  26   4   0   3   0   2   0]
 [ 25   3 737  10 124   1  86   1  13   0]
 [ 26  17  17 859  29   0  41   0  11   0]
 [  0   2 115  36 764   0  77   0   6   0]
 [  0   0   0   1   0 919   0  49   7  24]
 [145   1 121  38 101   0 571   0  23   0]
 [  0   0   0   0   0  34   0 937   0  29]
 [  8   1   6  13   5   6  20   5 935   1]
 [  0   1   0   0   0  13   1  39   0 946]]
```

Figure 2: Confusion matrix.

Classification report:

	precision	recall	f1-score	support
0	0.79	0.81	0.80	1000
1	0.97	0.96	0.97	1000
2	0.73	0.74	0.73	1000
3	0.83	0.86	0.84	1000
4	0.74	0.76	0.75	1000
5	0.94	0.92	0.93	1000
6	0.63	0.57	0.60	1000
7	0.91	0.94	0.92	1000
8	0.93	0.94	0.93	1000
9	0.95	0.95	0.95	1000
accuracy			0.84	10000
macro avg	0.84	0.84	0.84	10000
weighted avg	0.84	0.84	0.84	10000

Figure 3: Classification matrix.

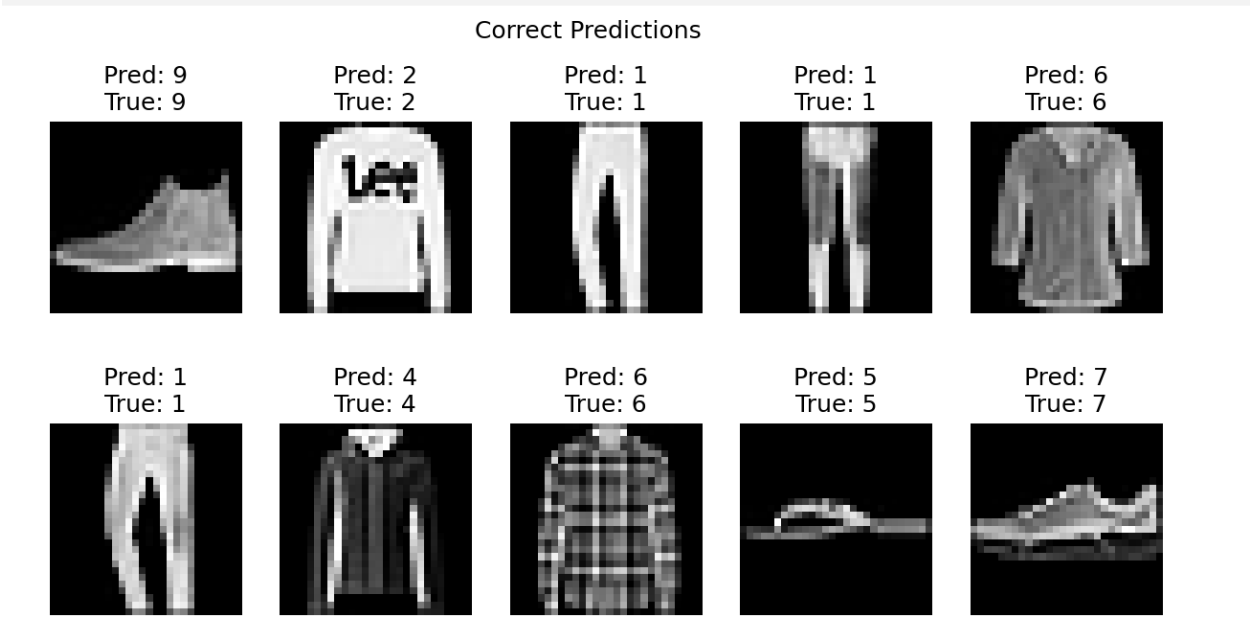

Figure 4: Correct prediction.

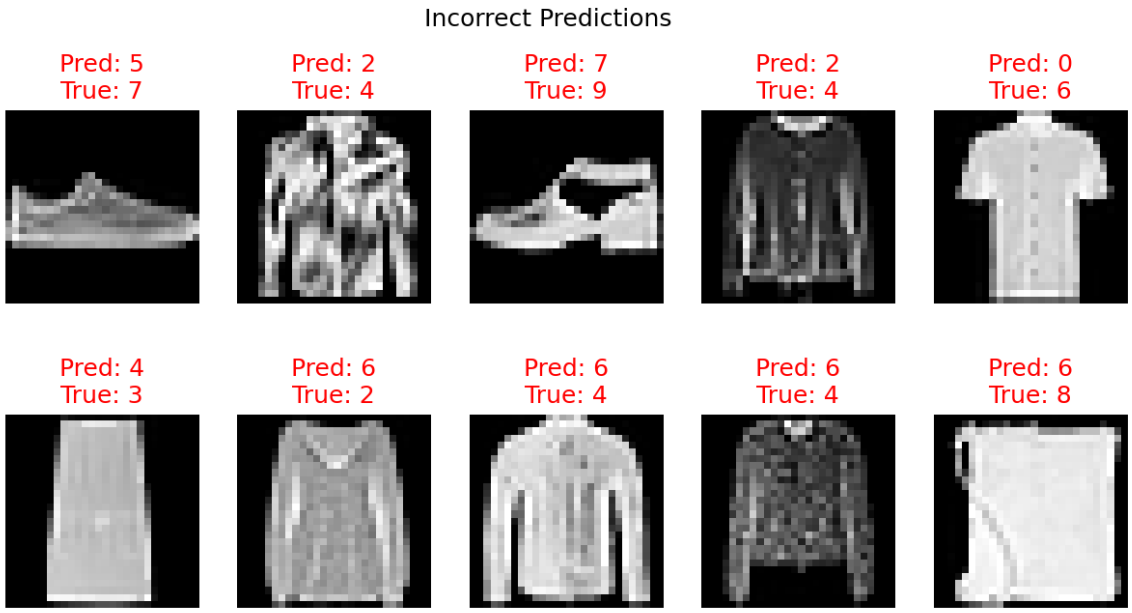

Figure 5: Incorrect prediction.