Part 2-1

We define a list of dictionaries, where each dictionary represents the hyper parameters for a different model. We then loop over each dictionary, build and compile a Keras model using the hyper parameters, train it on the training data, and evaluate its test accuracy using the model.evaluate method.

After each model is evaluated, we print the test accuracy for that model. This allows us to compare the performance of different models and select the best one.

For these models:

```
models = [
    {'reg_l1_param': 10e-6, 'hidden_size': 16},
    {'reg_l1_param': 10e-5, 'hidden_size': 32},
    {'reg_l1_param': 10e-4, 'hidden_size': 64}
]
```

Results are as follow:

Test accuracy for model 1 : 0.8587999939918518

Test accuracy for model 2 : 0.8629999756813049

Test accuracy for model 3 : 0.8784000277519226

Therefore, we will chose model 3.

Part 2-2

Feature importance is calculated as the sum of the absolute weights of each input feature in the first layer of the trained neural network.

With k=1000, we will get:
Score: accuracy(0.9014) - penalties(0.75=)
0.15139999999999998

So even that we get a high accuracy but the final score is not good.

If we use k=100,

Score: accuracy (0.8728) - penalties (0.075) =
0.7978000000000001

We will get a better score.

But with k=10

Score: accuracy(0.5318) - penalties(0.0075) =
0.5243000000000001

The accuracy will be too low.

K=200:

Score: accuracy(0.8964) - penalties(0.15) = 0.7464

K=110:

Score: accuracy(0.8788) - penalties(0.0825) = 0.7963

So the best number of feature will be k=100