# Sample Solution for Take-home Exam
# DAT340/DIT867: Applied Machine Learning
# 28-30 May 2022

## Part 1: Basic questions

### Question 1 of 11: Developing a machine learning model (8 points)

A company specialising in car/truck driver monitoring wants to build a machine learning system to recognise if the driver is holding the steering wheel or not, and whether the driver uses one or both hands when they hold the steering wheel. The company has collected lots of images from thousands of hours of videos of real driving data collected using a camera fitted in the cabin facing to the steering wheel. A few examples of the images can be seen in Figure 1.

Figure 1: Examples of the collected images related to Question 1.

Assume that you are working at the company. Your team gets the task to build a system for recognising hand(s) on the steering wheel as described earlier.

**(a, 6p)** Given the provided data, explain the steps you would suggest to implement the system.

**(b, 2p)** Discuss your opinion about any potential issues using images collected in this way for this purpose.

**Example solution**

**(a)** The following steps must be included.

- Learning task - There can be several ways of implementing this as a machine learning task. One possibility is to implement the system as a classification task with 0, 1, or 2 hand(s) on steering wheel. Doing it this way, we will be dealing with *multi-class classification problems*. For this, we will need labelled data.
- Dataset - Based on the information given, we have unlabelled dataset. Therefore, we need to do *annotation*. Even if we cannot annotate all images, we should annotate large enough numbers (perhaps several thousands). Since the data was collected from real driving data, it is highly likely that the data would be *imbalanced*, i.e., the number of images with no hands on steering wheel is likely to be much smaller than either one hand or two hands on steering wheel. If that is the case, we could apply oversampling or undersampling to get a more balanced dataset.
- Classifier - Which classifier to use needs to be determined. Since we are dealing with images, CNN could be something to try. We can use 2D convolutional and pooling layers, followed by dense layers to compute the output. The last layer is a softmax with 3 outputs.
- Data preprocessing - Appropriate data processing should be discussed, depending on the chosen classifier. Since we use CNN, we can normalise pixel values to a range [0,1] and standardize the images so that all have the same dimensions.

- Training/validation/test set - For the experiments, we set aside a small portion of the dataset as test set. Depending on the size of the dataset, we may want to either split the rest of the data into training set and validation set for development, or use all of the rest of the data for development using cross-validation.
- Evaluation metric - Target *evaluation metric* should be defined/chosen. Accuracy could be a choice to consider here (after the imbalanced data is addressed).
- Hyperparameter tuning - Appropriate hyperparameter tuning should be discussed, depending on the chosen classifier. Since we use CNN, we can experiment with size of convolutional filters and pooling regions, applying regularisation (such as dropout, early stopping), and also data augmentation (e.g., shifting the colours) to reduce overfitting. Training can be done using different optimisation methods, for example, Adam or basic SGD, with different learning rates. We can also experiment with using pre-trained model. In that case, the convolutional and pooling layers are from the pre-trained model. We then train our output layers on top of those layers from the pre-trained model.

**(b)** The issues you chose must be a result of using images collected in this way (i.e., from videos of real driving data) for the purpose stated in the question. One possible issue is the driver's face could be included in the images; in that case, such images should somehow be taken away to avoid any breached of privacy issue. There can be other issues as well such as a driver is caught by the camera when not holding on the steering wheel while driving in real traffic, especially if that leads to a crash.

## Question 2 of 11: Evaluating machine learning systems (3 points)

You are a representative of a private hospital. You are in charge of choosing a company that will be given a task to build a machine learning model that uses data from a simple and cheap screening to decide which patients should go for a second rather intrusive screening for a rare but dangerous cancer. Three companies that have joined the bidding to get the project were asked to build a small scale system. They were provided with a dataset to build the model, and a separate test set to measure the performance of their model. The provided data are in a form of numerical and categorical data. The provided data are from patients who have given consent. The test set consists of data from 920 healthy patients and 80 patients with cancer. The confusion matrices from the three companies are shown in Table 1, Table 2, and Table 3.

Table 1: Confusion matrix from the model from Company 1.

| | Predicted | |
|---|---|---|
| | Cancer | Non-cancer |
| Cancer | 68 | 12 |
| Non-cancer | 12 | 908 |

Table 2: Confusion matrix from the model from Company 2.

| | Predicted | |
|---|---|---|
| | Cancer | Non-cancer |
| Cancer | 76 | 4 |
| Non-cancer | 20 | 900 |

Table 3: Confusion matrix from the model from Company 3.

| | Predicted | |
|---|---|---|
| | Cancer | Non-cancer |
| Cancer | 77 | 3 |
| Non-cancer | 30 | 890 |

Compute the accuracy, precision, and recall of each of the models.

**Example solution**

Here, we calculate recall, precision, and accuracy, with respect to cancer.

For model from Company 1: Recall/sensitivity/true positive rate = 68/80 = 0.85; precision = 68/80=0.85; accuracy = (68+908)/1000 = 0.976.

For model from Company 2: Recall/sensitivity/true positive rate = 76/80 = 0.95; precision = 76/96 = 0.79; accuracy = (76+900)/1000 = 0.976.

For model from Company 3: Recall/sensitivity/true positive rate = 77/80 = 0.9625; precision = 77/107 = 0.72; accuracy = (77+890)/1000 = 0.967.

## Question 3 of 11: Dealing with features (3 points)

You are using machine learning method that takes time to train. To simplify and reduce the training time, you investigated various feature selection and dimensionality reduction techniques. Explain the difference between feature selection and dimensionality reduction techniques, using a particular feature selection method and a particular dimensionality reduction technique.

**Example solution**

Let's pick feature importance in random forest as a feature selection method and PCA as the chosen dimensionality reduction technique. The main difference is that: feature importance of random forest chooses a subset of the original features set (e.g., based on Gini importance), but when using PCA, we get a number of principal components (each of them is a linear combination of the features) not a subset of the original features. Therefore, it would be more difficult to explain the important features from PCA, compared to important features from random forest.

## Question 4 of 11: Annotating data (4 points)

A startup company working in the field of autonomous driving would like to build a machine learning system that is able to recognise new mobility tools (like e-bike, e-scooter, etc.) in traffic. For this purpose, the company has collected several thousands of traffic images. Before building the classifier, the company needs to label the images. For this purpose, they crowdsource the annotation task to anyone on the Internet. Within a short time, all of the several thousands of images are annotated; each of the images is annotated by two annotators.

To check on the quality of the annotation, the company checks the agreement between the annotators. It turns out that the agreement between the annotators is very high. Further, the Cohen's Kappa was calculated; it turns out to be 0.95.

**(a, 2p)** Does the company need to check the quality of the annotation further? Justify your answer.

**(b, 2p)** Are there potential ethical related issues of using the model trained using this annotated data?

**Example solution**

**(a)** Yes, if the company wants to use the data as ground truth, then the company needs to check the quality of the annotated data further. Agreement between the annotators does not mean that the annotation was correct.

**(b)** Yes. Since the annotation was done by crowdsourcing, there can be questions of whether the annotators were paid when conducting the work or whether they were voluntarily or being forced to participate.

## Question 5 of 11: Convolutional Neural Network (7p)

Consider the following convolutional neural network for image classification (see Figure 2):

**(a, 2p)** Do we need to do any data processing before using this convolutional neural network? Explain your answer.

**(b, 1p)** One of the first used kernel in this network is as follow:

$$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

What is this kernel detecting?

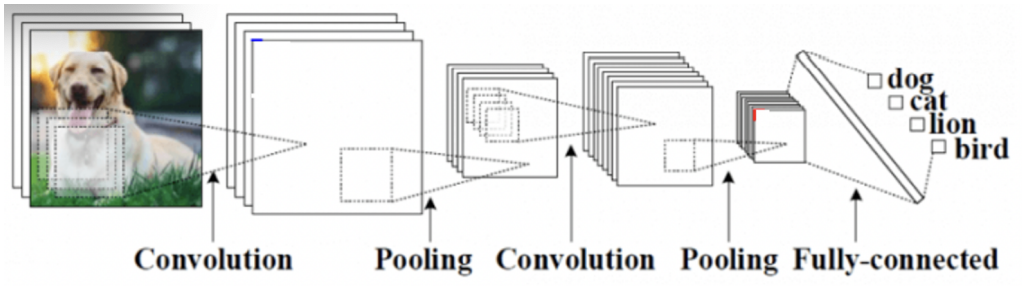


Figure 2: Examples of CNN for classification for Question 5.

**(c, 2p)** Which activation functions should be used in this convolutional neural network? Explain your answer.

**(d, 2p)** What does a pooling layer do in this architecture? What is the output of applying max pooling with pool size 3x3, with strides 1 to the feature map shown in Figure 3.

| | | | | |
|---|---|---|---|---|
| 11 | 14 | 4 | 5 | 8 |
| 34 | 5 | 9 | 15 | 27 |
| 32 | 6 | 18 | 3 | 13 |
| 4 | 29 | 14 | 18 | 20 |
| 1 | 5 | 9 | 13 | 15 |

Figure 3: Feature map for Question 5.

**Example solution**

**(a)** Yes. For example, neural networks models usually require scaled data and since we are dealing with images, we should normalise the data from [0-255] to [0-1]. We may also want to set so that the size of all of the images to be the same.

**(b)** The kernel detects vertical difference in pixel values.
Let's say we have an input image as shown in the left part of Figure 4. When we apply the kernel (the middle part of Figure 4) to the image on the left, this means that for each 2x2 block of pixels in the image, we multiply each pixel value in the block with the value in the corresponding position in the kernel and sum those up to give us a new pixel value. In the case of the block in grey, applying the kernel would give us: (56-17)+(16-32) = 23

| | | | |
|---|---|---|---|
| 56 | 17 | 31 | 38 |
| 16 | 32 | 79 | 10 |
| 46 | 36 | 94 | 75 |
| 70 | 15 | 85 | 55 |

| | |
|---|---|
| 1 | -1 |
| 1 | -1 |

| | | |
|---|---|---|
| 23 | | |
| | | |
| | | |

Figure 4: Applying kernel in CNN

**(c)** ReLU for the hidden layer because it helps speeding up training. Softmax should be used for the output since the task here is a multi-class classification problem. So the output will be expressed as the probability of an image belonging to a particular class.

**(d)** Pooling takes a feature map as input and produces another feature map. It summarises various regions on the feature map and by doing so it reduces the size/resolution of the feature map. The output of applying max pooling to Figure 3 is shown in Figure 5.

| | | |
|---|---|---|
| 34 | 18 | 27 |
| 34 | 29 | 27 |
| 32 | 29 | 20 |

Figure 5: Output of max pooling of feature map in Figure 3.

### Question 6 of 11: Decision tree and random forest (5 points)

**(a, 2p)** A decision tree is simple to use, can be visualised, and is easy to interpret. However, it can easily get overfitted. Mention and explain two ways to reduce overfitting when using a decision tree.

**(b, 3p)** Beside using a decision tree alone, we can reduce overfitting by making a forest of decision trees. Explain what makes an ensemble of trees (i.e., random forest) richer than a single decision tree.

**Example solution**

**(a)** There are a number of ways to reduce overfitting when using decision tree. Among them are limiting the tree depth (the deeper the tree the more likely to get overfitting) and the minimum number of samples required at a leaf node.

**(b)** The trees in a random forest will use different part of the data to train (as a result of using random samples and random subsets of the features). Therefore, a random forest reduces the overall overfitting and often has a better performance compared to a single decision tree.

## Part 2: Questions for the high grades

### Question 7 of 11: Continuation of Question 2 (4 points)

**(a, 2p)** To which company would you award the project? Justify your choice.

**(b, 2p)** What other factor(s) you might want to consider to help you make suggestion on which company to award the project to?

**Example solution**

**(a)** Since we are dealing with dangerous cancer, we want recall/sensitivity/true positive rate as high as possible (even though this could be at the cost of lower precision and accuracy). From this point-of-view, Company 3 seems to have the best potential. The reasoning: sending healthy patients for a second rather intrusive screening could cause unnecessary worry and inconvenient to patients but it is less of an issue compared to missing sending patients who have cancer for the second screening and therefore missing treatment. Others may have a different opinion on this issue.

**(b)** One important factor is interpretability/explainability of the model. Therefore, we might want to ask each company to mention at high level which method that they used to build their model. Note that it can be expected that the companies would not want to disclose too much information about their method at the bidding stage. Interpretability/explainability is important here because the patients may ask about the reasons for the decision.

### Question 8 of 11: Using small labelled data to label the unlabelled data (4 points)

To build a good classifier, we often need a large labelled dataset. However, labelling data is expensive. In many cases, we can only afford to annotate a very small fraction of the data. Discuss one common approach that would allow us to only annotate a very small fraction of the data, and yet still use the unlabelled data as much as possible.

**Example solution**

One common approach is to use the so called active learning. We trained a model using the small labelled data. The resulting model is then used to infer the label of unlabelled data. Depending on the confidence of the model when labelling the unlabelled data, we can then ask for annotator to annotate the data that has been labelled with low confidence by the model. The new labelled data will then be added to the set of labelled data for the next round of training. This way, we can reduce the need for annotator to label all of the data. Instead we rely on the model we built to help label most of the unlabelled data. We can also discuss the approach in the context of semi-supervised learning.

## Question 9 of 11: Mutual information (4 points)

Errata: the content of the second row, last column of Table 4 should be 1 (not 'J').

We work with a tabular dataset (in a file "train.csv") that has 26 columns, named with the letters 'A' to 'Z'. The last column, column 'Z', is the target variable. Most of the columns have string values. A few of the columns have numeric values. An example of the first 10 features of the first sample is shown in Table 4.

Table 4: The first 10 features of the first sample.

| 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' |
|---|---|---|---|---|---|---|---|---|---|
| 30 | 'B1' | 'C1' | 10 | 'E1' | 'F1' | 'G1' | 'H1' | 'I1' | 'J' |

We run the following code:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction import DictVectorizer
from sklearn.feature_selection import mutual_info_classif

train_data = pd.read_csv('train.csv')
n_cols = len(train_data.columns)
Xtrain_dicts = train_data.iloc[:, :n_cols-1].to_dict('records')
Ytrain = train_data.iloc[:, n_cols-1]

dv = DictVectorizer()
dv.fit(Xtrain_dicts)

X_vec = dv.transform(Xtrain_dicts)

feature_scores = mutual_info_classif(X_vec, Ytrain)

for score, fname in sorted(zip(feature_scores, dv.get_feature_names()), reverse=True)[:10]:
    print(fname, score)
```

The output is:

```
E=E2 0.10543223425355985
J 0.08338237212343601
G=G1 0.08087684110742101
A 0.0687725396789363
T 0.064872227626807
P=P1 0.06195072410418583
Y 0.0422833222022355
G=G2 0.03821610420273137
K 0.03698048451035268
I=I2 0.025765242400373284
```

**(a, 2p)** What does this code try to achieve?

**(b, 2p)** Why did we see `E=E2`, `G=G1`, `P=P1`, `G=G2`, and `I=I2` appear in the output list? Why does the output not look like the following?

```
E 0.10543223425355985
J 0.08338237212343601
G 0.08087684110742101
A 0.0687725396789363
T 0.064872227626807
P 0.06195072410418583
Y 0.0422833222022355
G 0.03821610420273137
K 0.03698048451035268
I 0.025765242400373284
```

**Example solution**

**(a)** The first part of the code is to import necessary libraries, read input data, split the input data to `Xtrain_dicts` (all except the last column in the input data, saved in a dictionary structure) and `Ytrain` (the last column of the input data), and then apply transformer DictVectorizer to turn `Xtrain_dicts` to a sparse matrix. The next part of the code is to compute the mutual information score for each of the features, to rank the features using the scores, and finally print the top 10 features based on the scores.

**(b)** The original features 'E', 'G', 'P', and 'I' must be variables with string values. Each of these variables become more than one feature depending on how many different values the variable has, as a result of transforming the lists of feature-value mappings to vectors using DictVectorizer.

## Question 10 of 11: Logistic regression for multi-class problem (4 points)

Discuss two ways how you might want to build a logistic regression model that can handle three-class output (A, B, and C).

**Example solution**

- A common approach can be to use one-vs-rest approach. This means that we build one binary classifier dedicated to class A (vs non-A), another binary classifier dedicated to B (vs non-B), and one more dedicated for class C (vs non-C). When classifying a new sample, the system outputs the class that is associated with the binary classifier that gives the highest output score.
- Another approach can be to use a softmax function instead of a sigmoid logistic function; this will create a so-called softmax regression. The softmax will produce probabilities associated with the three classes. The system outputs the class with the highest probability.

## Question 11 of 11: Thinking critical (4 points)

A team of data scientists and machine learning engineers in an insurance company proposed to analyse Facebook accounts (posts, likes, but not photos)) of first-time car drivers/owners to look for clues to derive personality traits that can be linked to safe driving behaviour. People that are considered as well organised, using the clues like "writing in short concrete sentences, using lists, and arranging to meet friends at a set time and place, rather than just tonight" would score well according to this scheme. On the other side, clues that suggest "overconfident such as the use of exclamation marks and the frequent use of "always" or "never" rather than "maybe" will count against the drivers. The idea is so that young car drivers/owners could show that they are safe drivers earlier, not to wait until they have several years of driving experience. Participation will be voluntary; those who are considered safe drivers can get a discount. The intention is not to use the scheme to increase price.

**(a, 2p)** If you are the manager who decides whether this project should go ahead or not, what questions would you be asking the team to help you in making decision?

**(b, 2p)** What are the potential concerns that you see from such a scheme?

**Example solution**

**(a)** There can be many questions the manager would want to know before making a decision. A few of the questions that are related to the potential of building a reliable system are listed here:

- Has there been previous work/research that have confirm the link between behaviour in social media to safe driving behaviour? If yes, could you describe what they found? If not,
- What is the plan in terms of establishing the link between how people write in Facebook posts and what people 'likes' in Facebook to the personality trait?
- Is it planned to transform this as a machine learning task? If so, how would the system be implemented? how do we get enough data to train the system? how can we verify the system?

**(b)** There can be many concerns, two are listed here.

- One major concern is if there is actually a link between how people write in Facebook posts and what people 'likes' in Facebook to safe driving behaviour and how this can be established/verified.
- Another concern is how the company should explain how the system works and how decision process is made to the customers (i.e., the explainability aspect).