IAML (LEVEL 10/11) – INFR10069/11152/11182: Assignment #1

Due on Monday, October 14, 2019 @ 16:00

NO LATE SUBMISSIONS

Student: **s1705544**

IMPORTANT INFORMATION

It is very important that you read and follow the instructions below to the letter - we will not be responsible for incorrect marking due to non-standard practices.

General Instructions

- This assignment is formative. Nonetheless, we will provide certain feedback on your answers. While you should use the providede notebooks, we only require you submit answers to Question 2.2, Question 2.6, Question 4.3 and Question 4.4.
- In order to simplify the submission and speed up the marking process, we have provided this template for filling in your answers to the questions. You will see that each question is individually annotated, and there is space for you to input your answer within the `\answer` environment. **DO NOT Modify** this template in any other way except where noted.
- Collaboration: You may discuss the assignment with your colleagues, provided that the writing that you submit is entirely your own. That is, you must **NOT** borrow actual text or code from others. We ask that you provide a list of the people who you've had discussions with (if any). Please refer to the Academic Misconduct page for what consistutes a breach of the above.
- You should have enough space in the answers boxes to answer all questions. Please do not change the heights of the answer boxes – this will greatly speed up grading.

In fact, most answer boxes are longer than needed so if you don't have enough space, you can definitely shorten your answer.

Assignment Structure

- This assignment is formative so that you can get familiar with the Gradescope system and so we can provide formative feedback on select questions. It still has a similar structure to the graded Assignment 2.

Submission Mechanics

Important: ***You must submit this assignment by Monday 14/10/2019 at 16:00. We do not accept Late Submissions for this coursework, except in the case of mitigating circumstances. Please refer to the ITO Website for further details.***

- We will use the Gradescope submission system for uploading assignments. Submission is in PDF Format (compilation of this latex document after filling in your answers).
- Make sure to input your Student ID in place of the above placeholder. You can do this by modifying line 33 in this tex-file.
- Make sure that you have filled in all the answers.
- Input your answers **ONLY** in the textboxes provided (this includes any images you may be asked for). **DO NOT Modify** this template in any other way.

Gradescope Instructions

You should receive an email titled "Welcome to Gradescope for IAML (Level 10/11)" in your student mailbox (e.g. s1234567@sms.ed.ac.uk). First you should set a password for your Gradescope account and remember it. **You will use the same account for Assignment 2** (and that one's graded), so make sure you have access to it! To submit:

- Run `pdflatex assignment.tex` 2 times to compile this into `assignment.pdf`
- Log on to Gradescope and select `Assignment 1` under Assignments.
- Submit the generated pdf.

The following are links to Gradescope's own instructions for using the system and submitting:

- An instructional video from Gradescope on their submission system
- A link to Gradescope's help page

PART A: 20-NEWSGROUPS

Naïve Bayes

Question 2.2 (4 points)

1. [Text] What is the assumption behing the Naive Bayes Model?
2. [Text] What would be the main issue we would have to face if we didn't make this assumption?

Answer Box (1.):
Naive Bayes assumes that each feature value in the same class are all independent, regardless of any correlation between the different features of that class. So the words are randomly distributed across the document.

Answer Box (2.):
Without the assumption, we need to calculate a 520-dimensional Gaussian for each class, but with the assumption, we just need to calculate 520 1-dimensional Gaussians for each class, which requires less calculations.

Question 2.6 (3 points)

[Text] Comment on the confusion matrix from the previous question. Does it look like what you would have expected? Explain.

Answer Box (1.):
The confusion matrix gives a good represenatation of the performance of the model and is better than the baseline. This model performs reasonable for the dataset and reaches around 87.8% accuracy and the main diagonal is close to 1 as expected, so accuracy is a reasonable metric to use as it is a multi-class classifier and the correct classification for each class matters.

PART B: AUTOMOTIVE DATASET

Multivariate Linear Regression

Question 4.3 (10 points)

In class we discussed ways of preprocessing features to improve performance in such cases.

1. [Code] Transform the 'engine-size' attribute using an appropriate technique from the lectures (document it in your code) and show the transformed data (scatter plot).

2. [Code] Then retrain a (Multi-variate) LinearRegression Model (on all the attributes including the transformed 'engine-size') and report R^2 and RMSE.

3. [Text] How has the performance of the model changed when compared to the previous result? and why so significantly?

Answer Box (1.)

```
Enter your code for part 1 here.

auto_numeric_copy = auto_numeric.copy()
plt.scatter(x=np.log(auto_numeric_copy['engine-size']), y=
    auto_numeric_copy['price'])
plt.title("price vs. engine-size")
plt.xlabel('engine-size')
plt.ylabel('price')
plt.show()
```

Answer Box (2.)

```
Enter your code for part 2 here.

auto_numeric_copy['engine-size'] = np.log2(auto_numeric_copy['engine-
    size'])
price = auto_numeric['price'].values
X = auto_numeric_copy.drop('price',axis = 1)
y = auto_numeric_copy['price']

lr = LinearRegression(fit_intercept=True, normalize=True, copy_X=True,
    n_jobs=1)
pred_cv = cross_val_predict(lr, X, y, cv=kf)

r2 = r2_score(price, pred_cv)
rmse = np.sqrt(mean_squared_error(price, pred_cv))
print('The Coefficient of Determination (R^2) on linear regression:
    {}'.format(r2))
print('The Root Mean Squared Error (RMSE) on linear regression: {}'.
    format(rmse))
```

Answer Box (3.):
The performance is better when compared to the previous results. This increase in performance is because linear regression is sensitive to outliers and outliers exist in engine-size. The outliers have a relatively large value, the logarithm minimises the value, and the influence of the outliers would minimise with it.

Question 4.4 (10 points)

The simplicity of Linear Regression allows us to interpret the importance of certain features in predicting target variables. However this is not as straightforward as just reading off the coefficients of each of the attributes and ranking them in order of magnitude.

1 [Text] Why is this? How can we linearly preprocess the attributes to allow for a comparison? Justify your answer.

2 [Code] Perform the preprocessing you just mentioned on the transformed data-set from Question 4.3, retrain the Linear-Regressor and report the coefficients in a readable manner. Tip: To simplify matters, you may abuse standard practice and train the model once on the entire data-set with no validation/test set.

3 [Text] Which are the three (3) most important features for predicting price under this model?

Answer Box (1.):
Linear regression is extremely sensitive to outliers. For a better performance, we can remove records with values more than threshold or we can transform the data, this will disregard useless features. The threshold can be adjusted for more accuracy. To preprocess the data wecan normalize the data. A preprocessing utility from the sklearn class that can be used is the StandardScaler which computes the mean and standard deviation on a training set.

Answer Box (2.)

```
Enter your code for part 2 here.
from sklearn.preprocessing import Normalizer, StandardScaler
#remove useless features
feature_names = auto_numeric.columns[1:16].values
X_tr= auto_numeric[feature_names]
# preprocessing using normalize data
normalizer = Normalizer()
X_tr = normalizer.fit_transform(X_tr)
# preprocessing using normalize data
standardiser = StandardScaler()
X_tr = standardiser.fit_transform(X_tr)
y_tr = auto_numeric1['price']
lr = LinearRegression(fit_intercept=True, normalize=True, copy_X=True,
    n_jobs=1)
pred_cv = cross_val_predict(lr, X_tr, y_tr, cv=kf)
r2 = r2_score(price, pred_cv)
rmse = np.sqrt(mean_squared_error(price, pred_cv))
print('R^2: {}'.format(r2))
print('RMSE: {}'.format(rmse))
```

Answer Box (3.):