DATA201-Test-Solutions

June 4, 2020

1 DATA201 Mid-term Test

Please use this page http://apps.ecs.vuw.ac.nz/submit/DATA201 for submitssion.

The due date is **Friday 22/5/2020 at 4pm**.

(Total 50 Marks)

1. Go to the Moral Machine website (http://moralmachine.mit.edu/) and do the "Judge" exercise. Also watch this clip https://www.youtube.com/watch?v=nhCh1pBsS80 about the Moral Machine, driverless cars, and related issues. Then write 200-300 words on how you would approach working on a team of engineers and programmers designing a driverless car with the capability to make choices of the kind that are needed in the Judging exercise. In your answer consider the ethical choices that are implied - including a discussion of which ones the team **does** have responsibility for, and which ones they **don't**. **(10 Marks)**

Answer:

ANS: Answers should comment on:

- The difficulty of making choices which may be culturally relative, and differ strongly between individuals
- The difference between action and inaction, which may lead to a difference in moral responsibility
- The difficulty in working with a team which may have a variety of views on values
- The counterbalancing value of having driverless cars making better and faster decisions that humans in scenarios where there is **not** a choice between two bad outcomes, but instead where lives can be saved.
- How decisions will be made: will the team follow the decision of its manager, the consensus of its members, or will it seek to follow societal norms, even if the team members disagree?
- The role of regulators
- Could comment on the fact that even though the algorithm may make mistakes/have unfortunate outcomes, overall the introduction of driverless cars will lead to fewer deaths
- Manufacturers should be open and honest about their design decisions
- A phase of public education may be required when driverless cars are introduced, to explain these risks
- Is it feasible for the car to identify and classify its victims in the time available?
- Will the safety algorithm degrade the performance of the other algorithms the car is running?
- Should the car's owner be able to set the level of selfishness of the car's actions? what would society think of that?

- What will the impact on the car's market be if people think the car will decide to kill them in certain circumstances?

(Answers cannot deny the existence of the problem: and say that if the car obeys the road rules then these choices will never arise, or that these events are rare. Accidents happen all the time, and designers always need to consider low-probability-high-consequence events.)

2. The following table of relationship status by age group potentially exposes characteristics of the individuals it contains, with one individual at particular risk of disclosure. Confidentialise the table in two ways (cell suppression to conceal the riskiest cell, and random rounding to base 3) – clearly explain the steps you have taken in each case, and comment on the advantages/disadvantages of the two approaches. **(10 Marks)**

Relationship Status by age:

Age Group	Single	Opposite sex couple	Same sex couple	Other	Total
15-19	0	0	1	0	1
20-24	8	7	4	2	21
25-29	14	10	4	0	28
30-34	7	8	0	1	16
34-39	3	2	3	1	9
Total	32	27	12	4	75

Answer:

Method 1: Cell suppression - suppress the risky cell (15-19, status "Same sex couple"), and three others so that the risky cell cannot be deduced. Firstly we suppress the cell itself, then choose one column in the same row ("Opposite sex couple"), one row in the same column (30-34), and then consequentially suppress a fourth cell (30-34, status "Same sex couple").

Age Group	Single	Opposite sex couple	Same sex couple	Other	Total
15-19	0	-	-	0	1
20-24	8	7	4	2	21
25-29	14	10	4	0	28
30-34	7	-	-	1	16
34-39	3	2	3	1	9
Total	32	27	12	4	75

Method 2: Random rounding - random round each cell to base 3. If a cell is a multiple of 3 leave it unchanged, otherwise with probability 2/3 round to the nearest multiple of 3, and with probability 1/3 round to the multiple of 3 that is two units away.

Age Group	Single	Opposite sex couple	Same sex couple	Other	Total
15-19	0	0	3	0	3
20-24	6	6	3	3	21
25-29	15	12	3	0	27
30-34	9	9	0	0	15

Age Group	Single	Opposite sex couple	Same sex couple	Other	Total
34-39	3	3	3	0	9
Total	33	27	12	6	75

(Answers will differ for each student - the above is just one example)

Comparison
- Both methods remove information from the user.
- With cell suppression we retain exact marginal counts, but can't use the table for calculations without making a guess at the content of the suppressed cells.
- The consequential cell suppression of four cells in order to conceal the content of just one cell means a lot of safe data are having to be suppressed.
- The counts in the table are still low, and there is still a lot of potential risk in the table: for example we can still deduce that the single 15-19 year old is not single, but is in a couple of **some** sort - Random rounding gives us a value in every cell - However the cells do not add up to the marginal totals, which means the table will be awkward in computations.

3. Study the Python code below and its output the answer the questions that follow. **(10 marks)**

Code (the dataset is not provided so you should not try running the code):

[ ]:
```
import pandas as pd
import numpy as np
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split, GridSearchCV

np.random.seed(0)

data = pd.read_csv("dataset.csv")

print(data.info())
print(data.survived.unique())

numeric_features = ['age', 'fare']
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())])

categorical_features = ['embarked', 'sex', 'pclass']
categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='constant', fill_value='missing')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))])

preprocessor = ColumnTransformer(
```

```
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)])

clf = Pipeline(steps=[('preprocessor', preprocessor),
                      ('classifier', LogisticRegression(solver='lbfgs'))])

X = data.drop('survived', axis=1)
y = data['survived']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
print(X_train.shape[0])

clf.fit(X_train, y_train)
print("%.3f" % clf.score(X_test, y_test))

param_grid = {
    'preprocessor__num__imputer__strategy': ['mean', 'median'],
    'classifier__C': [0.1, 1.0, 10, 100],
}

grid_search = GridSearchCV(clf, param_grid, cv=10, iid=False)
grid_search.fit(X_train, y_train)
print(grid_search.best_params_)
print("%.3f" % grid_search.score(X_test, y_test))
```

```
    Output:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1309 entries, 0 to 1308
Data columns (total 14 columns):
pclass       1309 non-null int64
survived     1309 non-null int64
name         1309 non-null object
sex          1309 non-null object
age          1046 non-null float64
sibsp        1309 non-null int64
parch        1309 non-null int64
ticket       1309 non-null object
fare         1308 non-null float64
cabin        295 non-null object
embarked     1307 non-null object
boat         486 non-null object
body         121 non-null float64
home.dest    745 non-null object
dtypes: float64(3), int64(4), object(7)
memory usage: 143.2+ KB
None
```

```
[1 0]
1047
0.790
{'classifier__C': 0.1, 'preprocessor__num__imputer__strategy': 'mean'}
0.798
```

Questions:

a. How many examples and features are there in the given dataset? (1 mark)

b. Which feature has the largest number of missing values? How many? (1 mark)

c. Which feature is used as the label for the classification problem in the code? How many classes are there? (1 mark)

d. How many samples are there in the test set? (1 mark)

e. What are the features used for model training? (1 mark)

f. What is the name of the classification algorithm used in the code? (1 mark)

g. What was done to the numerical features before model training? (2 marks)

h. From the outcome of using `GridSearchSV`, what should be done to improve the model `clf`? (1 mark)

i. If the hyperparameters found from using `GridSearchSV` are used, will the accuracy of the model on the test set improve? (1 mark)

Answer:

a. 1309 examples and 14 features

b. Feature `body` having 1309-121=1188 missing values

c. Feature `survived`. Two classes: 1 and 0

d. The test set has 1309-1047=262 samples.

e. 5 features: `age`, `fare`, `embarked`, `sex`, `pclass`.

f. Logistic Regression

g. (2 marks) Replace missing values using the median along each of the two columns `age` and `fare`, and then standardize those two features by removing the mean and scaling to unit variance.

h. Replace missing values using the *mean* instead of the *median* along each of the two columns `age` and `fare`, and increase regularization with `C = 0.1`.

i. Yes, it will increase from 79.0% to 79.8%.

4. The purpose of setting the random_state parameter in `train_test_split` is: (select all that apply) **(1 mark)**

    a. To avoid predictable splitting of the data

    b. To make experiments easily reproducible by always using the same partitioning of the data

c. To avoid bias in data splitting

d. To split the data into similar subsets so that bias is not introduced into the final results

Answer:

b

5. Given a dataset with 10,000 observations and 50 features plus one label, what would be the dimensions of X_train, y_train, X_test, and y_test? Assume a train/test split of 75%/25%. **(1 mark)**

   a. X_train: (2500,) ; y_train: (2500, 50) ; X_test: (7500,) ; y_test: (7500, 50)

   b. X_train: (10000, 28) ; y_train: (10000,) ; X_test: (10000, 12) ; y_test: (10000,)

   c. X_train: (2500, 50) ; y_train: (2500,) ; X_test: (7500, 50) ; y_test: (7500,)

   d. X_train: (7500, 50) ; y_train: (7500,) ; X_test: (2500, 50) ; y_test: (2500,)

   e. None of the above

Answer:

d

6. Which of the following is an example of multiclass classification (select all that apply)? **(1 mark)**

   a. Classify a set of fruits as apples, oranges, bananas, or lemons

   b. Predict whether an article is relevant to one or more topics (e.g. sports, politics, finance, science)

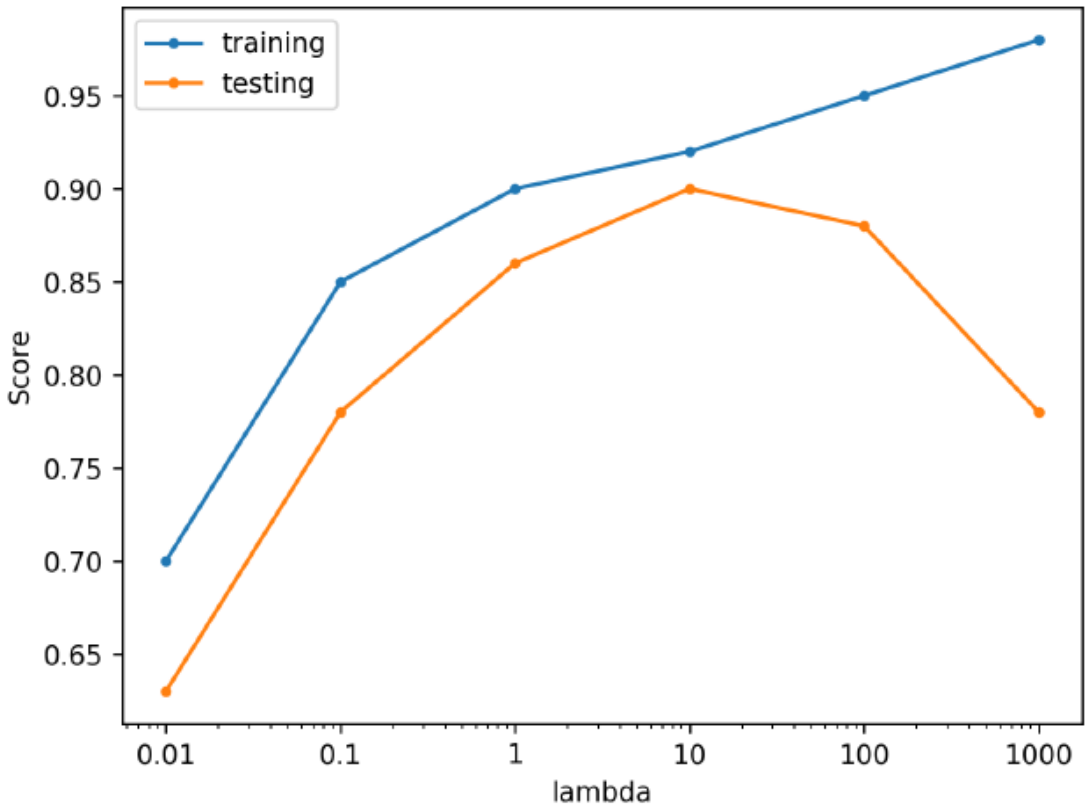
   c. Predicting both the rating and profit of soon to be released movie

   d. Classify a voice recording as an authorized user or not an authorized user.

Answer:

a

7. Looking at the plot below which shows accuracy scores for different values of a regularization parameter *lambda*, what value of *lambda* is the best choice for generalization? **(2 marks)**



Answer:

10

8. Which of the following is true of cross-validation (select all that apply)? **(1 mark)**
    a. Helps prevent knowledge about the test set from leaking into the model
    b. Fits multiple models on different splits of the data
    c. Increases generalization ability and computational complexity
    d. Increases generalization ability and reduces computational complexity
    e. Removes need for training and test sets

Answer:

a, b, c

9. A supervised learning model has been built to predict whether someone is infected with a new strain of a virus. The probability of any one person having the virus is 1%. Using accuracy as a metric, what would be a good choice for a baseline accuracy score that the new model would want to outperform? **(1 mark)**

Answer:

99%

10. Given the following confusion matrix:

	Predicted Positive	Predicted Negative
Condition Positive	96	4
Condition Negative	8	19

Compute the *accuracy*, *precision*, *recall*, and *specificity* (each to three decimal places) **(4 marks)**

Answer:

TP=96, TN=19, FP=8, FN=4

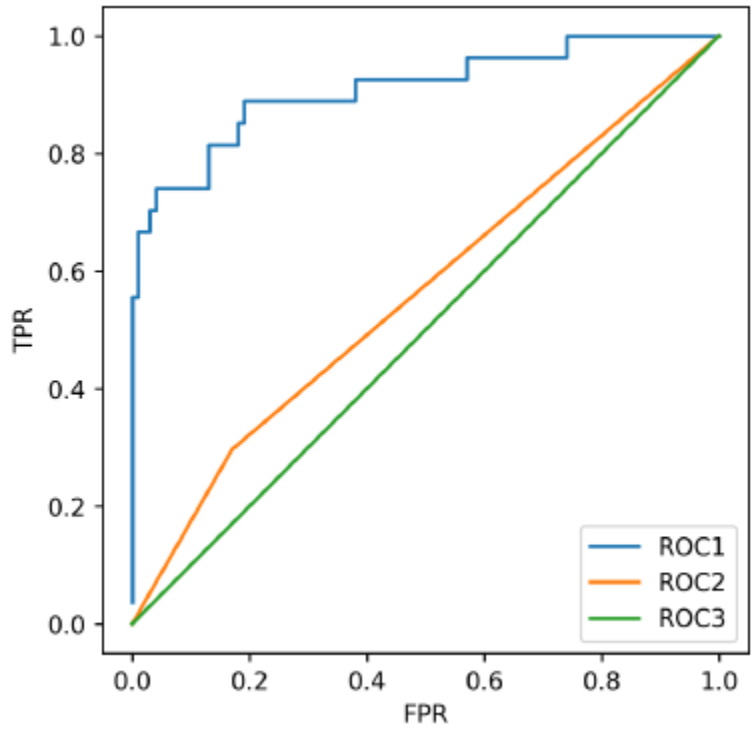
accuracy = 0.906

precision = 0.923

recall = 0.960

specificity = 0.704

11. Given the following models and AUC scores, find the corresponding ROC curve to each model. **(1 mark)**
    - Model 1 test set AUC score: 0.91
    - Model 2 test set AUC score: 0.50
    - Model 3 test set AUC score: 0.56



Answer:

- Model 1: ROC 1
- Model 2: ROC 3

- Model 3: ROC 2

12. A feature F1 can take certain value: A, B, C, D, E, & F and represents grade of students from a college. Which of the following statement is true in following case? **(1 mark)**
    a. Feature F1 is an example of nominal variable.
    b. Feature F1 is an example of ordinal variable.
    c. It doesn't belong to any of the above categories.
    d. Both of a and b

Answer:

b

13. [True or False] It is possible for a Pearson correlation between two variables to be zero but their values are still related to each other. **(1 mark)**

Answer:

TRUE

14. Suppose you are given 7 scatter plots from 1 to 7 as below (from left to right) and you want to compare Pearson correlation coefficients between variables of each scatterplot.

Consider the following statements about the relative values of the coefficients.

A. 1<2<3<4
B. 1>2>3>4
C. 7<6<5<4
D. 7>6>5>4

Which pair of statements is correct? **(1 mark)**

a. A and C
b. B and C
c. A and D
d. B and D

Answer:

b

15. Run the code in the Notebook cell below and write **one line** of code for each of the following questions **(5 marks)**

[1]:
```
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score, recall_score
from sklearn.pipeline import Pipeline
from sklearn.linear_model import SGDClassifier

X, y = make_classification(random_state=0)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
pipe = Pipeline([('scaler', StandardScaler()), ('classifier',␣
 ↪SGDClassifier(loss='log', random_state=42))])
pipe.fit(X_train, y_train);
```

a. Compute the precision score of the model on the test set (1 mark):

[2]:
```
precision_score(y_test, pipe.predict(X_test))
```

[2]:
```
0.7857142857142857
```

b. Compute the recall score of the model on the test set (1 mark):

[3]:
```
recall_score(y_test, pipe.predict(X_test))
```

[3]:
```
0.9166666666666666
```

c. Predict the class of the last instance in the test set (1 mark):

[4]:
```
pipe.predict([X_test[-1]])
```

[4]:
```
array([1])
```

d. Print the total number of instances in the test set which are correctly classified (2 marks):

[5]:
```
sum(y_test == pipe.predict(X_test))
```

[5]:
```
21
```