0917451 AI and Machine Learning (Spring 2022)

Midterm Exam

رقم التسجيل: | | الاسم: | |

==

Instructions: Time **50** min. Open book and notes exam. No electronics except using the lab PC. Please answer all problems in the respective shaded rectangular spaces and limit your answer to the space provided. There are five problems. Notice that this exam has 4 CSV files that we you need to copy to the working directory of your Python project.

==

P1. (a) Give the main three reasons why AI is succeeding now.

[3 marks]

1.	**Availability of training data**
2.	**Availability of improved ML algorithms**
3.	**Availability of fast processors**

(b) Give two main branches of AI.

[2 marks]

1.	**Machine learning and pattern recognition**
2.	**Logic-based AI**

Other acceptable answers:
3. Search
4. Knowledge representation
5. Planning
6. Heuristics
7. Genetic programming

P2. (a) What are the types of the following two machine learning problems?

[2 marks]

1. Predicting the cost of a hotel room given its features.	**Supervised learning/Regression**
2. Sorting data instances into normal and abnormal (not normal) given their attributes.	**Unsupervised learning/Anomaly Detection**

(b) Given the following error rates on the train and test sets, classify these three cases into *just right*, *over-fitting*, and *under fitting*.

[3 marks]

	Train Error	Test Error	Model Condition
1.	**40%**	**42%**	**Under fitting**
2.	**10%**	**12%**	**Just right**
3.	**10%**	**42%**	**Over fitting**

P3. Complete the following Python code to split the DataFrame `data` into 70% train set and 30% test set and separate the two sets to disjoint DataFrames: the features `x1`, `x2`, and `x3` DataFrame and the response `y` DataFrame.

[7 marks]

```
import pandas as pd
from numpy.random import randn
from sklearn.model_selection import train_test_split

data = pd.DataFrame(randn(100, 4),
                    columns=['x1', 'x2', 'x3', 'y'])
```

```
train_set, test_set = train_test_split(data, test_size=0.3)

X_train = train_set.drop('y', axis=1)
y_train = train_set['y'].copy()

X_test = test_set.drop('y', axis=1)
y_test = test_set['y'].copy()
```

P4. The following Python code loads the features and labels of the Iris dataset. Complete this code to evaluate the accuracy of the *k* nearest neighbors classifier for *k*=1 and *k*=3. For your evaluation, use the 3-fold cross validation technique. Which number of neighbors is best?

[6 marks]

```
import pandas as pd
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier

X = pd.read_csv('iris-features.csv')
y = pd.read_csv('iris-classes.csv').to_numpy().flatten()
```

```
knn_clf = KNeighborsClassifier(n_neighbors=1)
scores = cross_val_score(knn_clf, X, y, scoring="accuracy", cv=3)
print('k=1', scores)

knn_clf = KNeighborsClassifier(n_neighbors=3)
scores = cross_val_score(knn_clf, X, y, scoring="accuracy", cv=3)
print('k=3', scores)

k=1 [0.98 0.94 0.96]
k=3 [0.98 0.96 0.98]
```

Best *k* is	3

P5. The following Python code loads the features and labels of the Diabetes dataset. Complete this code to evaluate the RMSE of the SVM regressor using polynomial kernel of degree 5 and `C` parameter = 100. Evaluate the trained model using the scaled features.

[7 marks]

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error

X = pd.read_csv('diabetes.features.csv')
y = pd.read_csv('diabetes.labels.csv').to_numpy().flatten()

scaler = StandardScaler()
```

```
scaler.fit(X)
X_scaled = scaler.transform(X)

svm_reg = SVR(kernel="poly", degree=5, C=100)
svm_reg.fit(X_scaled, y)

yh = svm_reg.predict(X_scaled)
mse = mean_squared_error(y, yh)
print('rmse =', np.sqrt(mse))
```

<Good Luck>