1. Como você definiria Machine Learning (ML)?
Machine Learning é a ciência da computação que busca criar programas que aprendem com dados sem a necessidade de serem reprogramados.

2. Você pode citar três tipos de problemas onde ML se destaca?
ML pode se destacar no desenvolvimento de BOT's, na classificação de Spam ou no reconhecimento de pessoas e objetos em fotos e vídeos

3. O que é um conjunto de treinamento rotulado?
É um treinamento que o usuário determina rótulos que o programa usa como base para reconhecer padrões e realizar as tarefas

4. Que tipo de algoritmo de aprendizado de máquina você usaria para permitir que um robô ande por vários terrenos desconhecidos?
Reinforcement Learning, visto que, o agente visualiza o ambiente e executa ações buscando recompensas positivas ou negativas, seguindo um regulamento estabelecido, aprendendo as melhores decisões e escolhendo o melhor caminho

5. Que tipo de algoritmo você usaria para segmentar os clientes de um e-commerce em vários grupos?
Instance-Based, pois ele reconhece e aprende padrões que usa para agrupar novos casos semelhantes

6. Você enquadraria o problema da detecção de spam como um problema de aprendizado supervisionado ou problema de aprendizado não supervisionado?
Supervisionado, visto que necessita da ajuda do usuário com o reconhecimento de novos padrões suspeitos que devem ser analizados

7. Utilizando as bases de dados:
- http://archive.ics.uci.edu/ml/index.php
- https://www.kaggle.com/datasets
- https://registry.opendata.aws/

Encontre um dataset (conjunto de dados) que ache interessante para preencher a seguinte definição:
A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.
Você deve apontar o dataset, e apontar pelo menos um artigo científico onde aquele dataset foi usado. Descreva pelo menos uma das tarefas (task T) em que o dataset tem sido utilizado. Descreva pelo menos uma métrica (performance measure P) utilizada com aquele dataset, e como os dados estão organizados (experience E).
Veja o último slide disponível no moodle para um exemplo.
Data set: Wine Quality
P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties.
In Decision Support Systems, Elsevier, 47(4):547-553, 2009.
Task: Classificar a qualidade dos vinhos, visto que, existem muito mais vinhos vinhos

Performance Measure: Foram alcançados resultados encorajadores, com o modelo SVM a apresentar os melhores desempenhos, superando as técnicas NN e MR, sobretudo para o vinho verde branco, que é o tipo mais comum. Ao admitir apenas as classes classificadas corretas (T = 0,5), as acurácias globais são de 62,4% (vermelho) e 64,6% (branco). Deve-se notar que os conjuntos de dados contêm seis/sete classes (de 3 a 8/9). Essas precisões são muito melhores do que as esperadas por um classificador aleatório. O desempenho melhora substancialmente quando a tolerância é definida para aceitar respostas corretas dentro de uma das duas classes mais próximas (T = 1,0), obtendo uma precisão global de 89,0% (vermelho) e 86,8% (branco). Em particular, para ambas as tarefas a maioria das classes apresenta uma acurácia individual (precisão) superior a 90%.
Experience: 4898

8. Considere o notebook (código Python) que acompanha a atividade. Ele inclui apenas um classificador, baseado em decision trees (https://en.wikipedia.org/wiki/ Decision_tree). Pesquise no site do scikit-learn (https://scikit-learn.org/stable/ supervised_learning.html#supervised-learning) um algoritmo adicional e inclua-o no código. Execute o novo algoritmo e calcule a acurácia de suas classificações (siga o modelo passado). A acurácia do novo algoritmo é melhor do que aquela da Decision Tree? Utilize o esquema de visualização disponível no arquivo para visualizar graficamente quais dígitos seu algoritmo tem mais dificuldade em classificar corretamente.
Algoritmo de Classificação: Recognizing hand-written digits
O algoritmo baseado em Decision Tree possui uma accuracy de 81,25%, já o baseado em Clusterização tem uma accuracy de aproximadamente 97%, e os dígitos que ele tem mais dificuldade de classificar são 9(93%), 8(94%) e 5(95%)

```
###Coloque seu código Aqui
# Author: Gael Varoquaux <gael dot varoquaux at normalesup dot org>
# License: BSD 3 clause

# Standard scientific Python imports
import matplotlib.pyplot as plt

# Import datasets, classifiers and performance metrics
from sklearn import datasets, svm, metrics
from sklearn.model_selection import train_test_split


digits = datasets.load_digits()

_, axes = plt.subplots(nrows=1, ncols=4, figsize=(10, 3))
for ax, image, label in zip(axes, digits.images, digits.target):
    ax.set_axis_off()
    ax.imshow(image, cmap=plt.cm.gray_r, interpolation="nearest")
    ax.set_title("Training: %i" % label)
```

```
# flatten the images
n_samples = len(digits.images)
data = digits.images.reshape((n_samples, -1))

# Create a classifier: a support vector classifier
clf = svm.SVC(gamma=0.001)

# Split data into 50% train and 50% test subsets
X_train, X_test, y_train, y_test = train_test_split(
    data, digits.target, test_size=0.5, shuffle=False
)

# Learn the digits on the train subset
clf.fit(X_train, y_train)

# Predict the value of the digit on the test subset
predicted = clf.predict(X_test)

_, axes = plt.subplots(nrows=1, ncols=4, figsize=(10, 3))
for ax, image, prediction in zip(axes, X_test, predicted):
    ax.set_axis_off()
    image = image.reshape(8, 8)
    ax.imshow(image, cmap=plt.cm.gray_r, interpolation="nearest")
    ax.set_title(f"Prediction: {prediction}")

print(
    f"Classification report for classifier {clf}:\n"
    f"{metrics.classification_report(y_test, predicted)}\n"
)
```