Michał Konieczny

Part 1:

1. Supervised learning – models train from labeled data
   Unsupervised learning – models identify patterns or structures in unlabeled data
   Reinforcement learning – models learn to make decisions by receiving reward or penalty for an action

2. Regression predicts numerical values, classification predicts discrete categories

3. Linear Regression - predicts outcomes by fitting a linear relationship between input features and the target
   Decision Tree - a flowchart-like structure for making decisions based on feature splits
   Neural Network - a computational model inspired by biological neural networks, consisting of layers of nodes (neurons) for complex pattern recognition

4. Training set is the data used to train the model and test set is data used to evaluate the model's performance on unseen data

5. Overfitting - the model captures noise and fits training data too well, performing poorly on new data
   Underfitting - the model is too simplistic and fails to capture patterns in the training data

6. Linear Regression - regularization techniques
   Decision Trees - pruning, limiting tree depth, or minimum samples per leaf
   Neural Networks - Dropout, regularization, or early stopping

7. Definition - EDA involves analyzing and summarizing datasets to understand their structure, detect anomalies, and derive insights
   Purpose - identify data patterns, assess data quality, and guide modeling decisions

8. Outlier - a data point that significantly deviates from the rest of the dataset, possibly due to variability or error

9. Accuracy - proportion of correct predictions (useful for balanced classification)
   Precision - true positives among predicted positives (important for low false positives)
   Recall - true positives among actual positives (important for low false negatives)
   F1-Score - balance between precision and recall.
   MSE - penalizes large prediction errors (regression)
   R-squared- explains variance captured by the model (regression)

10. Definition - technique to evaluate model performance by splitting the dataset into subsets (folds), training on some folds, and validating on others
    Purpose - mitigates overfitting and provides a more robust estimate of model performance\

Part 4:

1. The dataset has 150 rows and 5 columns.

```python
import numpy as np
import pandas as pd

iris = pd.read_csv('iris.csv')

rows, cols = iris.shape
print(f"The dataset has {rows} rows and {cols} columns.")

```

```
The dataset has 150 rows and 5 columns.
[Finished in 531ms]
```

2. Feature Variables: sepallength, sepalwidth, petallength, petalwidth
   Target Variable (Other): class (categorical variable with values Iris-setosa, Iris-versicolor, Iris-virginica).

```python
import numpy as np
import pandas as pd

iris = pd.read_csv('iris.csv')

print(iris.columns)
```

```
Index(['sepal.length', 'sepal.width', 'petal.length', 'petal.width',
       'variety'],
      dtype='object')
[Finished in 530ms]
```

3. Positive instances (Iris-setosa): 50.
   Negative instances (Iris-versicolor and Iris-virginica): 100

```python
import numpy as np
import pandas as pd

iris = pd.read_csv('iris.csv')

print(iris.columns)
print(iris['variety'].unique())

iris['variety'] = iris['variety'].str.strip()

iris['enjoy'] = (iris['variety'] == 'Setosa').astype(int)

positive_instances = iris['enjoy'].sum()
negative_instances = len(iris) - positive_instances

print(f"Positive instances: {positive_instances}")
print(f"Negative instances: {negative_instances}")
```

```
Index(['sepal.length', 'sepal.width', 'petal.length', 'petal.width',
       'variety'],
      dtype='object')
['Setosa' 'Versicolor' 'Virginica']
Positive instances: 50
Negative instances: 100
[Finished in 529ms]
```

4. petallength and petalwidth are likely the best features to separate the data correlation will confirm this

```python
import pandas as pd

iris = pd.read_csv('iris.csv')

iris['enjoy'] = (iris['variety'] == 'Setosa').astype(int)

correlations = iris[['sepal.length', 'sepal.width', 'petal.length',
 'petal.width']].corrwith(iris['enjoy']).abs()

print("Correlation between features and 'enjoy' (Setosa vs others):")
print(correlations.sort_values(ascending=False))

```

```
Correlation between features and 'enjoy' (Setosa vs others):
petal.length    0.922765
petal.width     0.887344
sepal.length    0.717416
sepal.width     0.603348
dtype: float64
[Finished in 517ms]
```

5. No such column as humidity.

Part 5:

1. MYCT: Processor time in microseconds or cycles. It represents the time taken by the CPU for certain operations.
   MMIN: Minimum memory size in kilobytes.
   MMAX: Maximum memory size in kilobytes.
   CACH: Cache size in kilobytes.
   CHMIN: Minimum cache size in kilobytes.
   CHMAX: Maximum cache size in kilobytes.
   Class: This represents the CPU class or performance category, and class ≡ PRP (which is another name for this column).

2. The main difference would be in the features and the target variable (class), and possibly the addition of categorical features like vendor in one dataset and not the other.

3. The data type of 'vendor' column is: object
   The data type of 'class' column is: int64

```python
import pandas as pd

df = pd.read_csv('cpu_vendor.csv')

vendor_dtype = df['vendor'].dtype
class_dtype = df['class'].dtype

print(f"The data type of 'vendor' column is: {vendor_dtype}")
print(f"The data type of 'class' column is: {class_dtype}")

```

```
The data type of 'vendor' column is: object
The data type of 'class' column is: int64
[Finished in 590ms]
```