1)

1. W uczeniu nadzorowanym dane są opisane etykietami (np. tom, jerry), w uczeniu nienadzorowanym dane nie są opisane, a w uczeniu ze wzmocnieniem początkowo nie mamy danych, tylko środowisko z którego model będzie sam pobierał dane i maksymalizował otrzymywaną nagrodę.
2. Regresja rozpoznaje wartości rzeczywiste, a klasyfikacja odróżnia jeden obiekt od drugiego.
3. Regresja liniowa to metody oparte o liniowe kombinacje zmiennych i parametrów dopasowujących model do danych. Drzewa decyzyjne służą do wyodrębniania wiedzy z zestawu przykładów. Zakładamy, że posiadamy zestaw przykładów: obiektów opisanych przy pomocy atrybutów, którym przyporządkowujemy jakąś decyzję. Szukamy parametrów rozdzielających dane. Sieć neuronowa to system przeznaczony do przetwarzania informacji, którego budowa i zasada działania są w pewnym stopniu wzorowane na funkcjonowaniu fragmentów rzeczywistego systemu nerwowego.
4. Zbiór uczący - zbiór na którym model dokonuje analiz. Zbiór testowy - zbiór służący do sprawdzania prawdziwości wyników ze zbioru uczącego.
5. Overfitting - przeuczenie. Model będzie dawał świetne wyniki na danych uczących, ale słabe na danych z którymi się nie zetknął. Zbyt dużo parametrów w stosunku do rozmiaru próby. Underfitting - zbyt mało parametrów opisujących dane, model będzie dawał niedokładne wyniki.
6. regresja - regularyzacja, Cross-Validation
   drzewa decycyjne - używanie większej ilości danych do trenowania
   sieci neuronowe - dropout
7. EDA - proces mający na celu zrozumienie charakterystyki danych przy użyciu technik wizualizacji oraz metod statystyki opisowej. W wyniku tego procesu możemy dostrzec i pozbyć się niepasujących danych.
8. outlier - niepasujące, odstające dane.
9. Metryki:
   Accuracy - procent dokładnych przewidywań dla modelu.
   Precision - procent odpowiednich przykładów (prawdziwych pozytywów) wśród wszystkich przewidywanych przykładów.
   Recall - proporcja przykładów, które model przewidział że należą do klasy, do przykładów które rzeczywiście do niej należą.
   krzywa ROC - ROC to skrót od Receiver Operating Characteristics. Im bardziej jest uniesiona i znajduje się dalej od przekątnej, tym lepszy jest model.
10. walidacja krzyżowa - metoda statystyczna polegająca na podziale próby statystycznej na podzbiory, a następnie przeprowadzeniu wszelkich analiz na niektórych z nich, tzw. zbiór uczący, podczas gdy pozostałe służą do potwierdzenia wiarygodności jej wyników, tzw. zbiór testowy.

4)

1. Zbiór danych ma 6 rekordów, z których każdy ma 7 atrybutów(sky, airTemp, humidity, wind, water, forecast, enjoy).
2. mamy 7 atrybutów - 6 pierwszych to cechy, ostatnia to etykieta. Zmienna niezależna jest tą zmienną, którą w badaniu manipulujemy. Zmienna zależna jest tą zmienną którą mierzymy.
3. jest 6 instancji, 4 pozytywne i 2 negatywne.
4. najlepiej rozdziela airTemp

5. są 4, mają numery 1,2,3 i 5

5)

1. cpu-vendor ma dodatkowo kolumnę “vendor”
2. vendor to object a class to int64
3. typ danych ventor z object został zmieniony na uint8
4.

```
[37] # sprawdź dane i wykonaj regresję
     import pandas as pd
     import numpy as np
     df_v = pd.read_csv('cpu-vendor.csv')
     df = pd.read_csv('cpu.csv')

     df_v['vendor'] = pd.get_dummies(df_v['vendor'])
```

```
     # przewidywanie
     from sklearn.linear_model import LinearRegression
     from sklearn.model_selection import train_test_split

     x = df_v.drop(columns=['class'])
     y = df_v['class']

     X_train,X_test, y_train, y_test = train_test_split(x,y,test_size=0.2)
     model = LinearRegression()
     model.fit(X_train, y_train)
     r_sq = model.score(X_test,y_test)
     y_pred = model.predict(X_test)



     print("coefisient: ", model.coef_)
     print("intersept: ", model.intercept_)
     print('predicted response:', y_pred, sep='\n')
     print('coefficient of determination:', r_sq)
```

```
[45] X = df.drop(columns=['class'])
     y = df['class']
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
     model = LinearRegression()
     model.fit(X_train,y_train)
     r_sq = model.score(X_test,y_test)
     y_pred = model.predict(X_test)
     print("coefisient: ", model.coef_)
     print("intersept: ", model.intercept_)
     print('predicted response:', y_pred, sep='\n')
     print('coefficient of determination:', r_sq)
```

5. współczynniki dla df_v: `[-9.20113356e+01  7.32361017e-02 1.47750342e-02 6.50288550e-03  5.25025634e-01 -2.49148430e-01  1.40993327e+00]`
   model.intercept_ dla df_v: `-71.83454501631314`
   współczynnik determinacji dla df_v: `0.840816910975272`

współczynniki dla df: `[ 0.05254026  0.01538387  0.00518948  0.79114886 -0.49430198  1.68703889]`
model.intercept_ dla df: `-57.35182839450542`
współczynnik determinacji dla df: `0.7048771529364473`

6)

1.

```
[46] # wczytaj zbiór danych
     swimming = files.upload()
```

```
[72] from sklearn import tree
     from sklearn.tree import DecisionTreeClassifier

     clf_entropy = DecisionTreeClassifier(criterion='entropy')
     clf_gini = DecisionTreeClassifier
```

```
     # pokaż zbiór danych
     df = pd.read_csv('swimming.csv')
     df.head(6)
```

```
[73] x = pd.get_dummies(df.drop('enjoy', axis=1))
     y = df['enjoy']

     clf = tree.DecisionTreeClassifier()

     clf = clf.fit(x, y)
```

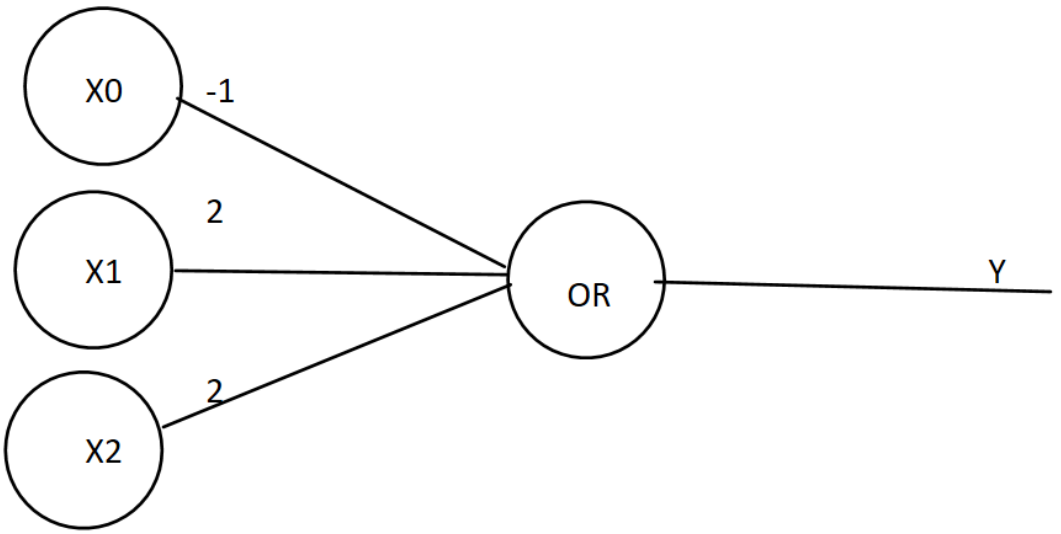
```
[66] pd.get_dummies(df.drop('enjoy', axis=1)) # co tu się dzieje?
```

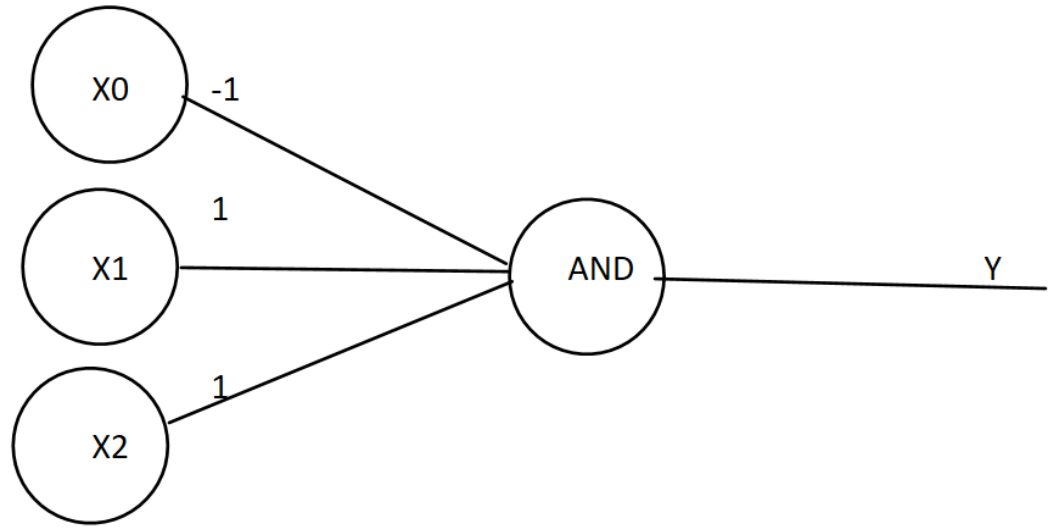
```
     # wykorzystując poniższy kod zwizualizuj oba uzyskane drzewa

     from IPython.display import Image
     import pydotplus

     columns = pd.get_dummies(df.drop('enjoy', axis=1)).columns

     dot_data = tree.export_graphviz(clf, out_file=None, rounded=True, filled=True,
                                     feature_names=columns)
     graph = pydotplus.graph_from_dot_data(dot_data)
     Image(graph.create_png())
```

2. Trzeba przekodować, ponieważ wartości class nie są numeryczne.
3. max_depth - odpowiada za maksymalną głębokość drzewa
   min_samples_split - minimalna wartość próbek do podziału
   min_samples_leaf - gwarantuje minimalną ilość próbek w każdym liściu

7)

1.





Funkcja OR: 2x1+2x2-1
Funkcja AND: x1+x2-1

3. Dzieli tablicę danych na losowe zbiory testowe i treningowe.