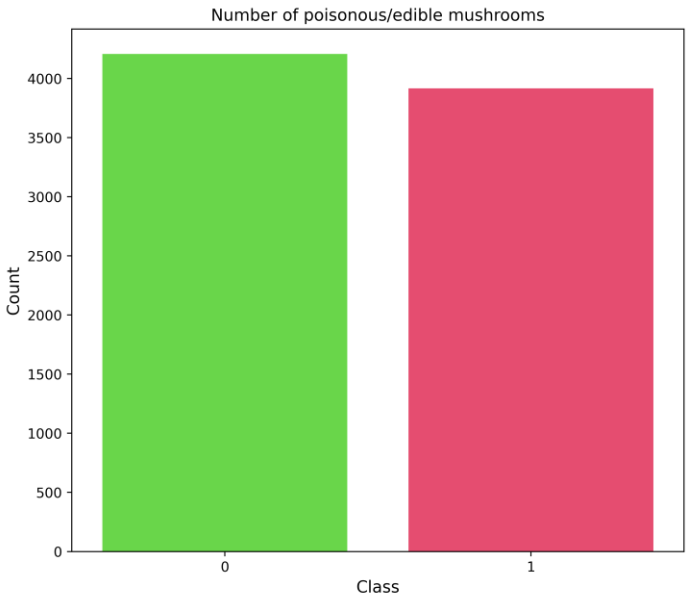
Mushroom classification project

A dolgozatom témája a különböző fajtájú gombák klasszifikációja (osztályozása) aszerint, hogy az egyes gombafajok ehetők vagy mérgezők. Az osztályozás során az ehető gombafajok 1-es értéket kapnak, míg a mérgező gombafajokat 0-os értékkel látjuk el. Különböző osztályozási algoritmusok segítségével betaníthatjuk a modelljeinket teszt adatokon, így a program képes lesz az újonnan kapott adatokat osztályokba sorolni, jelen esetben 1-es vagy 0-os osztályba.

A témám választásakor több dolgot vettem figyelembe: először is egy olyan területet akartam, ami napjainkban releváns, foglalkoztatja a tudományt. A gépi tanulás remek lehetőség az előbb említettekre, mivel a mesterséges intelligencia egy részhalmaza és köztudott, hogy a mesterséges intelligencia napjaink egyik leggyorsabban fejlődő, a jövő meghatározó technológiája. Másrészt személyes érdeklődés is köt a választott projekt felé, mert mindig is érdekelt gépi tanulás. Célom elsajátítani a gépi tanulás alapjait, olyan gépeket tanítani be, amelyek az adatok emberi szintű elemzésére és értékelésére lesznek alkalmasak.

Ebben a projektben tehát adatokat gyűjtünk különböző gombákról, majd több gépi tanulásos modellt hozzunk létre, amelyek segítségével megvizsgáljuk a gombák mivoltját a különböző gomba tulajdonságok alapján, mint például a gomba kalapjának formája, színe vagy a lemezének színe stb.…

Az adatok a Kaggle weboldalról származnak. Kaggle, a Google LLC leányvállalata, az adattudósok és gépi tanulással foglalkozó online közösség. Az adathalmaz különböző gomba tulajdonságokat tartalmaz, mint például: a kalap formája, színe, felülete, illata stb.… Ugyanakkor tartalmazza, hogy milyen osztályba csoportosítható: e – mint edible (ehető), p – mint poisonous (mérgező). Az alábbi ábrán láthatjuk, hogy az adathalmaz kiegyensúlyozott.



1. Ábra. Mérgező/ehető gombák száma

Az adatunk kategorikus, azaz minden tulajdonság egy-egy betűvel van jelölve (például: a kalap barna színe „n” betűvel van jelölve, a szürke szín „g” és így tovább), emiatt át kell alakítanunk sorrendi adattá, amit megértenek a különböző osztályozási algoritmusok.

Ha megnézzük a korrelációt a különböző gomba tulajdonságok változói és a csoporteloszlás között, akkor azt véljük felfedezni, hogy a „gill-color” rendelkezik a legnagyobb abszolút értékkel (-0.53), általában ezt tekintjük a legfontosabb változónak a klasszifikáláskor.

```
gill-color     class
0              1.000000
8              1.000000
3              0.721311
2              0.670213
7              0.428954
11             0.255814        - láthatjuk, hogy a 0, 8, 1 és 6-os értékek
10             0.204659        teljes mértékben meghatározzák a gombának
4              0.156863        ehető vagy mérgező mivoltját
5              0.106870
9              0.097561
1              0.000000
6              0.000000
```

Ezek után felosztjuk az adathalmazunkat teszt és train (betanítási) adatokra. Az arány a felosztás során 10% tesztadat és 90% az adatoknak betanításra használódik fel.

A felosztott adathalmazokat alávetjük különböző osztályozási algoritmusoknak, a projektben a különböző algoritmusokkal dolgozunk: Decision Tree Classification, Logistic Regression Classification, KNN Classification, SVM Classification, Naive Bayes Classification és Random Forest Classification.

Általában a fent említett algoritmusok két paramétert várnak, az egyik az X, ami egy tömböt jelent és a betanítási mintákat tartalmazza és egy Y egész számokat tartalmazó tömböt, ami a betanítási minták osztályainak címkéit tartalmazza, példa:

```
>>> from sklearn import tree
>>> X = [[0, 0], [1, 1]]
>>> Y = [0, 1]
>>> clf = tree.DecisionTreeClassifier()
>>> clf = clf.fit(X, Y)
```

Betanítás után, a model használható a minták osztályának megjósolására:

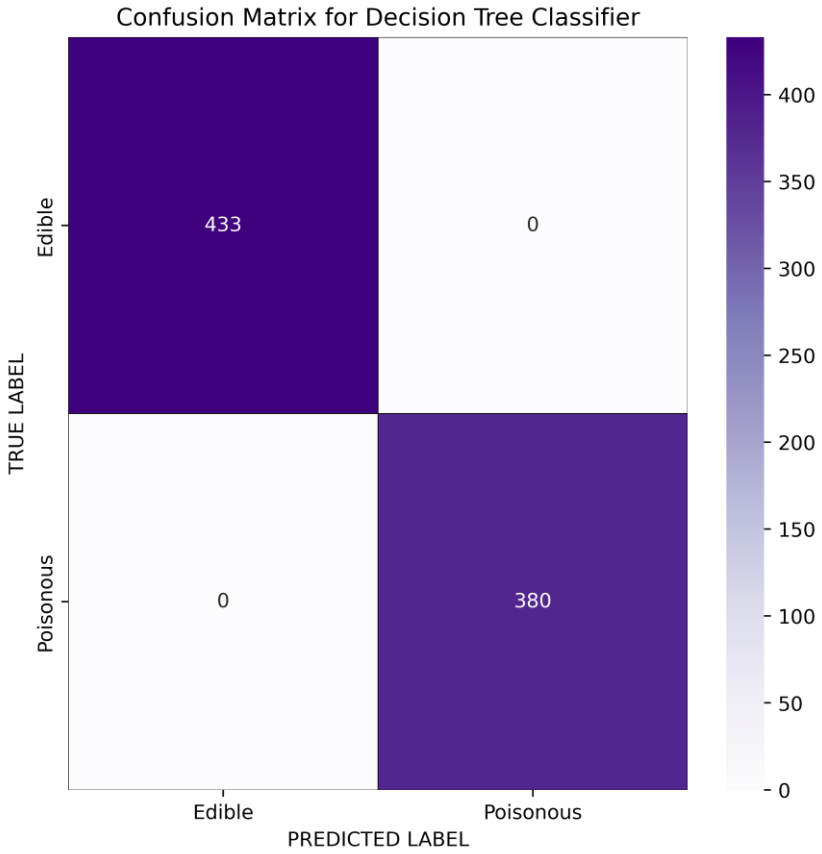
```
>>> clf.predict([[2., 2.]])
array([1])
```

Minden algoritmus után készítünk egy reportot (jelentést), ami a következőket tartalmazza:

- Precision: hány százaléka a jóslásoknak helyes?
- Recall: a pozitív esetek hány százalékát találta el az algoritmus?
- F1 score: hány százaléka a pozitív eseteknek helyes?
- Support: az osztály eseteinek száma az adathalmazban

	precision	recall	f1-score	support
0	1.00	1.00	1.00	433
1	1.00	1.00	1.00	380

Ugyanakkor egy Confusion Matrixot is készítünk, ami egy N x N mátrix, amely kiértékeli a teljesítményét az osztályozási modellnek, ahol N a célzott számú osztályokat jelöli, példa:



2. Ábra. Decision Tree Classifier Confusion Matrix

A különböző modellek segítségével különböző jóslatokat tudunk futtatni a teszt adatokra.

A projektben nem beszélhetünk nagyon integrált informatikai rendszerekről, mivel csupán egy jupyter notebookot tartalmaz a docker-compose.yml fájlunk, viszont ezt ha egy dockeren belül composoljuk akkor kész is van a dolgozó munkafüzet.

Összefoglalóként elmondható, hogy egy számomra nagyon érdekes munkafolyamatként éltem meg a választott projektet, tanultam a különböző osztályozási algoritmusokról és nem utolsó sorban a jósoló rendszer segítségével ezentúl ehetünk egészséges gombákat.