

UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

PREGLED SORODNIH DEL IN NAČRT

POVEZLJIVI SISTEMI IN INTELIGENTNE STORITVE

Gašper Gračner

Martin Oprešnik

Luka Koštomaj

Maribor, avgust 2016

1 Pregled sorodnih del

2 Načrt dela

Projektno delo smo si razdelili na tri dele in sicer tako, da bomo za priporočanje športne aktivnosti uporabili različne pristope strojnega učenja. Uporabili bomo Deep Neural networks, SVM in . Vsi se bomo prvič srečali z metodami stojnega učenja zato je načrtovanje dela kar zahtevno.

Pri delu bomo uporabili Python in njegove knjižnice, za sledenje delu pa GitHub.com.

2.1 Sprint1

V prvem sprintu smo si zadali analizo podatkov, v kateri bomo sodlovali vsi. Podatke bomo brali s pomočjo Python knjižnice *gpxpy* (<https://github.com/tkrajina/gpxpy>). Potrebne funkcionalnosti knjižnice bomo enkapsulirali, da jih bomo lahko enostavneje uporabili pri nadaljnjem delu.

Nad podatki bomo izvedli gručenje in jih prikazali v grafični obliki, da si bomo lažje predstavljali njihov pomen. Poskušali pa bomo ustvariti še učno množico podatkov.

2.2 Sprint2

2.2.1 Gašper - Analiza parametrov in delovanja metode

Med raziskovanjem o SVM sem naletel na opis te metode v programskem jeziku Python. Uporabil je knjižnico *sklearn* in modul *svm*. Zato bom v tem sprintu poskušal analizirati parametre *cache_size*, *class_weight*, *coef0*, *decision_function_shape*, *degree*, *gamma*, *kernel*, *max_iter*, *probability*, *random_state*, *shrinking*, *tol*, in *verbose*.

```
1 from sklearn import svm
2 X = [[0, 0], [1, 1]]
3 y = [0, 1]
4 clf = svm.SVC()
5 clf.fit(X, y)
6 SVC(C=1.0, cache_size=200, class_weight=None,
7      coef0=0.0, decision_function_shape=None,
8      degree=3,
9      gamma='auto', kernel='rbf',
10     max_iter=-1, probability=False,
11     random_state=None, shrinking=True,
12     tol=0.001, verbose=False)
```

Primer 1: Uporaba SVM v Pythonu

2.3 Sprint3

2.3.1 Gašper

Po analizi parametrov se bom lotil same implementacije rešitve in prilaganja parametrov, za naš problem, saj predvidevam, da bom s prilagajanjem parametrov lahko rahlo izboljšal začetno rešitev. Zasnoval pa si bom še tesno množico, v primeru, da tega ne bomo uspeli v prvem sprintu.

2.4 Sprint4

2.4.1 Gašper

Na koncu, bom poskusal uporabiti in primerjati še različne klasifikatorje, ki podpirajo SVM in so implementirani v uporabljeni knjižnici (*SVC*, *NuSVC* in *LinearSVC*). Moral pa bom določiti tudi primerno metodo primerjanja klasifikatorjev.

2.5 Viri

Exploring GPX files (<https://ocefpaf.github.io/python4oceanographers/blog/2014/08/18/gpx/>)

Support vector machine (https://en.wikipedia.org/wiki/Support_vector_machine)

SVM Tutorial (<http://www.svm-tutorial.com/2014/10/support-vector-regression-r/>)

Support Vector Machines (<http://scikit-learn.org/stable/modules/svm.html#complexity>)