

## ΑΝΑΦΟΡΑ ΓΙΑ ΤΗΝ 1Η ΕΡΓΑΣΙΑ ΣΤΗΝ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΚΑΘΗΓΗΤΗΣ	ΚΩΝΣΤΑΝΤΙΝΟΣ ΜΠΛΕΚΑΣ	
ΑΜ	4679	4065
ΜΕΛΗ ΟΜΑΔΑΣ	ΚΑΖΑΚΙΔΗΣ ΘΕΟΧΑΡΗΣ	ΚΑΖΑΚΙΔΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ
ΕΤΟΣ	3 <sup>ο</sup>	5 <sup>ο</sup>

Για την υλοποιηση της εργασιας χρησιμοποιησαμε την πλατφορμα jupyter. Διευκολυνε την υλοποιηση καθως σπαι τον κωδικα σε κελια και μπορουμε να τρεχουμε ξεχωριστα κομματια κωδικα αλλα σε ένα script

Από το datasheets ορισαμε τα δεδομενα (bone\_length,rotting\_flesh,hair\_length,has\_soul,color) ως x(αυτά τα χαρακτηριστικα που θα κανουν το train,το id δεν επηρεαζει το αποτελεσμα αρα δεν το περνουμε υποψιν) και το type ως y που είναι και το target μας

Εχουμε τα δεδομενα μας σε κανονικοποιημενη μορφη επομενως μπορουμε να τα χειριστουμε πιο ευκολα και να βγαλουμε καλυτερα αποτελεσματα και δεν χρειαζεται να ταμετατρεψουμε αφου είναι ηδη

Γραψαμε σε μορφη κωδικα τις συναρτησεις που μας δωσατε στο pdf καθως μας είναι πολύ χρησιμες στην συνεχεια και στην υλοποιηση του project

Επειδη το project βασιζεται σε μαθηματικα μοντελα πρεπει να μετατρεψουμε τα string σε αριθμους (όπως το color και το type) για να τα φορτωσουμε στο x,y μοντελο αρα τα αντιστοιχουμε με αριθμους

**\*\*Αμα δεν τρεχει στην εντολη train\_set=pd.read\_csv('..../input/ghouls-ghost-goblin/train.csv') παρακαλουμε βαλτε το δικος σας path για το train.csv. Επισης χρησιμοποιησα το train.csv και όχι το train.csv.zip ,μπορει να φαινονται αυτονοητα αλλα οφειλουμε να τα αναφερουμε**

Μεσω τις βιβλιοθηκης sklearn και συγκεκριμενα from sklearn.model\_selection import train\_test\_split εχουμε μια βοηθητικη συναρτηση που σπαι τα x και y δεδομενα σε train και test αντιστοιχα

αφου όταν κανουμε εκπαιδευση θελουμε να σπασουμε τα δεδομενα σε δυο ομαδες x\_train,y\_train και x\_test,y\_test.Επισης βαζουμε το stratify για εξασφαλισουμε μια ισορροπια στο μοντελο μας αρα θα εκπαιδευσουμε εξισου καλα ολα τα ειδη monster

## KNN Μεθοδος:

Πρόκειται για ένα σχετικά απλό αλγόριθμο ο οποίος ελέγχει όλες τις περιπτώσεις και ταξινομεί τις καινούριες σύμφωνα με τα γειτονικά στοιχεία. Για να γίνει η ταξινόμηση αυτή χρησιμοποιείται μία συνάρτηση απόστασης

Η 1η μεθοδο που χρησιμοποιουμε είναι η knn όπως γνωριζουμε το knn δεν εχει εκπαιδευση για αυτό χρησιμοποιει την ευκλειδια απόσταση γιατι ουσιαστικα μετραι την αποσταση μεταξυ των features των monsters και αμα η αποσταση είναι μεσα στην γειτονια που εχει σχηματισει τοτε βρισκουμε οντως το κάθε type.Για κάθε τυπου monster υπαρχει διαφορετικη γειτονια.

Δημιουργησαμε έναν πινακα με τις τιμες του k και μια loor που φορτωνει τα δεδομενα στο knn κανει τις πραξεις βρισκει τις συσχετισεις με διαφορετικο k κάθε φορα και στην συνεχεια κανει τις προβλεψεις με τα x\_test.Τα x\_test δεδομενα βγαζουνε y\_test αποτελεσματα αλλα αυτά τα y είναι τα predictions.Για να δουμε αμα οντως πετυχε το αποτελεσμα γ κανουμε το πινακα confusion matrix δηλαδη οσα είναι TP(πανω στην διαγωνιο).Ετσι βρισκουμε τα κελια που μας δινουν το TP,FN,TN,FP και το μεταφραζουμε σε κωδικα (ξεχωριστα για ghost,goblins,ghoul) .Επειτα καλουμε τις συναρτησεις που φτιαξαμε παραπαν για το (accuracy,precision,recall,f1) και τελος αφου κανουμε για τοκαθενα ξεχωριστα κανουμε την διαδιακασια για το συνολικο και βρισκουμε τα συνολικα αποτελεσματα(ουσιαστικα κανουμε ένα μεσο ορο το αποτελεσματων)

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <b>Type II Error</b>	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <b>Type I Error</b>	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

### Τα αποτελεσματα για KNN:

Παρατηρούμε ότι το καλυτερό accuracy το πετυχαίνουμε με k=5 και συγκεκριμένα είναι Accuracy = 0.8103448275862069

Ghost k: 1 Accuracy: 0.8717948717948718 Precision: 0.8888888888888888 Recall: 0.6666666666666666 F1 score: 0.761904761904762	Ghost k: 3 Accuracy: 0.9 Precision: 0.7857142857142857 Recall: 0.9166666666666666 F1 score: 0.8461538461538461
Goblin k: 1 Accuracy: 0.6578947368421053 Precision: 0.5 Recall: 0.7692307692307693 F1 score: 0.6060606060606061	Goblin k: 3 Accuracy: 0.6578947368421053 Precision: 0.5 Recall: 0.38461538461538464 F1 score: 0.4347826086956522
Ghoul k: 1 Accuracy: 0.7894736842105263 Precision: 0.7777777777777778 Recall: 0.5384615384615384 F1 score: 0.6363636363636364	Ghoul k: 3 Accuracy: 0.7105263157894737 Precision: 0.5714285714285714 Recall: 0.6153846153846154 F1 score: 0.5925925925925927
Total k: 1 Accuracy: 0.7739130434782608 Precision: 0.6578947368421053 Recall: 0.6578947368421053 F1 score: 0.6578947368421053	Total k: 3 Accuracy: 0.7586206896551724 Precision: 0.631578947368421 Recall: 0.631578947368421 F1 score: 0.631578947368421

<p>Ghost k: 5 Accuracy: 0.95 Precision: 0.8571428571428571 Recall: 1.0 F1 score: 0.923076923076923</p> <p>Goblin k: 5 Accuracy: 0.7105263157894737 Precision: 0.5833333333333334 Recall: 0.5384615384615384 F1 score: 0.5599999999999999</p> <p>Ghoul k: 5 Accuracy: 0.7631578947368421 Precision: 0.6666666666666666 Recall: 0.6153846153846154 F1 score: 0.64</p> <p>Total k: 5 Accuracy: 0.8103448275862069 Precision: 0.7105263157894737 Recall: 0.7105263157894737 F1 score: 0.7105263157894737</p>	<p>Ghost k: 10 Accuracy: 0.9047619047619048 Precision: 0.75 Recall: 1.0 F1 score: 0.8571428571428571</p> <p>Goblin k: 10 Accuracy: 0.6842105263157895 Precision: 0.5384615384615384 Recall: 0.5384615384615384 F1 score: 0.5384615384615384</p> <p>Ghoul k: 10 Accuracy: 0.7894736842105263 Precision: 0.7777777777777778 Recall: 0.5384615384615384 F1 score: 0.6363636363636364</p> <p>Total k: 10 Accuracy: 0.7966101694915254 Precision: 0.6842105263157895 Recall: 0.6842105263157895 F1 score: 0.6842105263157895</p>
--	---

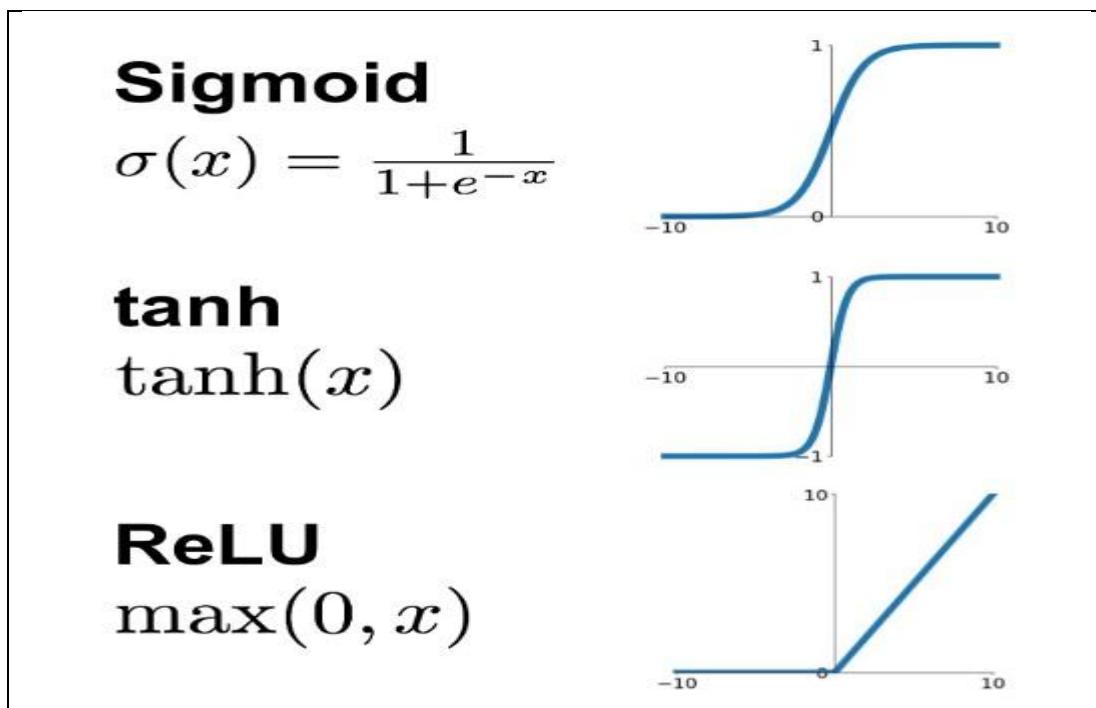
<p>Total k: 1 Accuracy: 0.7739130434782608 Precision: 0.6578947368421053 Recall: 0.6578947368421053 F1 score: 0.6578947368421053</p> <p>Total k: 5 Accuracy: 0.8103448275862069 Precision: 0.7105263157894737 Recall: 0.7105263157894737 F1 score: 0.7105263157894737</p>	<p>Total k: 3 Accuracy: 0.7586206896551724 Precision: 0.631578947368421 Recall: 0.631578947368421 F1 score: 0.631578947368421</p> <p>Total k: 10 Accuracy: 0.7966101694915254 Precision: 0.6842105263157895 Recall: 0.6842105263157895 F1 score: 0.6842105263157895</p>
---	---

### Νευρωνικα Δικτυα Μεθοδος:

Κανουμε τρεις εκπαιδευσεις ένα κρυφο επιπεδο με 50,100 και 200 νευρωνες και τρεις εκπαιδευσεις με δυο κρυφα επιπεδα με (50 νευρωνες το πρωτο και 25 το δευτερο ),(100 νευρωνες το πρωτο και 50 το δευτερο),(200 νευρωνες το πρωτο και 100 το δευτερο). Κανουμε μια λουπα που θα κανει δοκιμες αντιστοιχα για κάθε περιπτωση νευρωνων.

Χρησιμοποιούμε για συναρτηση ενεργοποίησης αναμεσα στους νευρωνες, την tanh\*\*. Χρησιμοποιούμε ως optimazer το Stochastic Gradient Descent (sgd) ώστε να γινονται καλυτερες συσχετισεις μεταξυ των δεδομενων. Επειτα φορτωνουμε τα δεδομενα στο νευρωνικο δίκτυο, κανουμε τις προβλεψεις και μετα κανουμε confusion matrix όπως και στην παραπάνω μεθοδο

\*\* παρατηρησαμε ότι με την logistic μερικα αποτελεσματα εβγαζαν σφαλαματα όπως παν και ετσι πειραματιστηκαμε και βαλαμε σαν συναρτηση ενεργοποιησης την tanh και ειδαμε καλυτερα αποτελεσματα, επισης χρησιμοποιησαμε την relu καπεριμεναμε ότι θα βγαλει ακομα καλυτερα αποτελσματα καθως ειδαμε ότι θεωρητικα είναι η πιο αποδοτικη αλλα στην περιπτωση αυτή η tanh ηταν καλυτερη



### Τα αποτελεσματα των νευρωνικων δικτυων:

- 1) Με ένα κρυφο επιπεδο

Ghost Hidden layers: (50,) Accuracy: 0.7 Precision:0.5 Recall: 0.3333333333333333 F1 score: 0.4	Ghost Hidden layers: (100,) Accuracy: 0.825 Precision:0.7777777777777777 Recall: 0.5833333333333333 F1 score: 0.6666666666666666	Ghost Hidden layers: (200,) Accuracy: 0.8292682926829 Precision:0.7272727272727272 Recall: 0.6666666666666666 F1 score: 0.6956521739130
Goblin Hidden layers: (50,) Accuracy: 0.7105263157894736 Precision:0.625 Recall: 0.3846153846153846 F1 score: 0.47619047619047	Goblin Hidden layers: (100,) Accuracy: 0.631578947368421 Precision:0.4285714285714285 Recall: 0.2307692307692307 F1 score: 0.3	Goblin Hidden layers: (200,) Accuracy: 0.631578947368421 Precision:0.4 Recall: 0.153846153846153846 F1 score: 0.2222222222222222
Ghoul Hidden layers: (50,) Accuracy: 0.5526315789473684 Precision:0.4090909090909090 Recall: 0.6923076923076923 F1 score: 0.5142857142857142857	Ghoul Hidden layers: (100,) Accuracy: 0.6052631578947368421 Precision:0.4545454545454545 Recall: 0.76923076923076923 F1 score: 0.57142857142857142857	Ghoul Hidden layers: (200,) Accuracy: 0.6578947368421 Precision:0.5 Recall: 0.846153846153846153846 F1 score: 0.62857142857142857142857
Total Hidden layers: (50,) Accuracy: 0.6551724137931052631 Precision:0.47368421052631 Recall: 0.473684210526315 F1 score: 0.473684210526315	Total Hidden layers: (100,) Accuracy: 0.6896551724137931052631 Precision:0.52631578947368421 Recall: 0.52631578947368421 F1 score: 0.52631578947368421	Total Hidden layers: (200,) Accuracy: 0.7094017094017094017094017 Precision:0.552631578947368421 Recall: 0.552631578947368421 F1 score: 0.552631578947368421

## 2)Με δυο κρυφά επιπέδα

Ghost Hidden layers: (50, 25) Accuracy: 0.8 Precision:0.75 Recall: 0.5 F1 score: 0.6	Ghost Hidden layers: (100, 50) Accuracy: 0.9 Precision:0.8333333333333333 Recall: 0.8333333333333333 F1 score: 0.8333333333333333	Ghost Hidden layers: (200, 100) Accuracy: 0.875 Precision:0.8181818181818181 Recall: 0.75 F1 score: 0.782608695652
Goblin Hidden layers: (50, 25) Accuracy: 0.578947368421052631 Precision:0.3333333333333333 Recall: 0.2307692307692307 F1 score: 0.2727272727272727	Goblin Hidden layers: (100, 50) Accuracy: 0.7368421052631 Precision:0.8 Recall: 0.3076923076923076 F1 score: 0.4444444444444444	Goblin Hidden layers: (200, 100) Accuracy: 0.6578947368421 Precision:0.5 Recall: 0.3076923076923076 F1 score: 0.380952380952
Ghoul Hidden layers: (50, 25) Accuracy: 0.52631578947368421 Precision:0.3809523809523 Recall: 0.6153846153846153 F1 score: 0.4705882352941	Ghoul Hidden layers: (100, 50) Accuracy: 0.78947368421052631 Precision:0.6190476190476190476 Recall: 1.0 F1 score: 0.764705882352941	Ghoul Hidden layers: (200, 100) Accuracy: 0.684210526315 Precision:0.52631578947368421 Recall: 0.76923076923076923076 F1 score: 0.625
Total Hidden layers: (50, 25) Accuracy: 0.6379310344827 Precision:0.447368421052631 Recall: 0.447368421052631 F1 score: 0.447368421052631	Total Hidden layers: (100, 50) Accuracy: 0.8103448275862 Precision:0.7105263157894736 Recall: 0.7105263157894736 F1 score: 0.7105263157894736	Total Hidden layers: (200, 100) Accuracy: 0.741379310344 Precision:0.605263157894 Recall: 0.60526315789473 F1 score: 0.605263157894

Παρατηρούμε ότι τα καλυτερά αποτελεσματα τα εχει το νευρωνικό επιπεδο με πρωτο κρυφο επιπεδο 100 και δευτερο 50 με Accuracy = 0.8103448275862069

## SVM Μέθοδος:

Τα δεδομένα χωρίζονται σε όμοιους πληθυσμούς για να μπορέσουν αργότερα να δημιουργηθούν οι κλάσεις και ορίζονται συντεταγμένες, τοποθετώντας γραμμές σαν σύνορα για να χωρίσουν τα δεδομένα με την μικρότερη απόσταση από το πιο κοντινό αντικείμενο του κάθε group. Δημιουργούμε ένα πινακα με δυο ειδων kernel (linear και rbf). Φορτωνουμε τα δεδομενα στο svm μοντελο, το εκπαιδευουμαι και ακολουθουμε παλι την ιδια διαδικασια με confusion matrix όπως στις παραπανω μεθοδους

### Τα αποτελεσματα της SVM μεθοδου:

Linear kernel	RBF
Ghost Kernel: linear Accuracy: 0.9473684210526315 Precision:1.0 Recall: 0.8333333333333334 F1 score: 0.9090909090909091	Ghost Kernel: rbf Accuracy: 0.9 Precision:0.7857142857142857 Recall: 0.9166666666666666 F1 score: 0.8461538461538461
Goblin Kernel: linear Accuracy: 0.8157894736842105 Precision:0.75 Recall: 0.6923076923076923 F1 score: 0.7199999999999999	Goblin Kernel: rbf Accuracy: 0.8157894736842105 Precision:0.75 Recall: 0.6923076923076923 F1 score: 0.7199999999999999
Ghoul Kernel: linear Accuracy: 0.868421052631579 Precision:0.75 Recall: 0.9230769230769231 F1 score: 0.8275862068965517	Ghoul Kernel: rbf Accuracy: 0.868421052631579 Precision:0.8333333333333334 Recall: 0.7692307692307693 F1 score: 0.8
Total Kernel: linear Accuracy: 0.8771929824561403 Precision:0.8157894736842105 Recall: 0.8157894736842105 F1 score: 0.8157894736842104	Total Kernel: rbf Accuracy: 0.8620689655172413 Precision:0.7894736842105263 Recall: 0.7894736842105263 F1 score: 0.7894736842105263

Παρατηρούμε ότι το καλυτερο accuracy το πετυχαίνουμε με linear kernel και συγκεκριμενα είναι Accuracy = 0.8771929824561403

## Naïve Bayes Μεθοδος:

Είναι ένας αλγόριθμος για classification ο οποίος βασίζεται σύμφωνα με το θεώρημα του Bayes στη μοναδικότητα των δεδομένων. Δηλαδή στο ότι κάποιες ιδιότητες και χαρακτηριστικά είναι μοναδικά σε συγκεκριμένα αντικείμενα και μέσα από αυτά μπορούμε να καθορίσουμε το είδος και να τα διαχωρίσουμε

Περνούμε τα πρώτα τεσσερά features και τα ταξινομούμε με την κατανομή normal distribution(gaussian) και το color το κατανομούμε με multinomial distribution. Επειτα βρισκουμε το γινομενο των δυο πινακων (βρηκαμε στο github μια βιβλιοθηκη για mixed naive bayes κωδικα που ταιριαζει στην ασκηση μας δηλαδη με πολλαπλασιασμο multi με gaus και το χρησιμοποιησαμε για να βρουμε τα αποτελεσματα) επειτα τα κανονικοποιησαμε κανουμε τις προβλεψεις και confusion matrix όπως στις παραπνω μεθοδους και βρισκουμε τα τελικα αποτελεσματα

### Τα αποτελεσματα για Naïve Bayes μεθοδο:

```
Ghost
Accuracy: 0.8717948717948718
Precision:0.7692307692307693
Recall: 0.8333333333333334
F1 score: 0.8

Goblin
Accuracy: 0.8157894736842105
Precision:0.875
Recall: 0.5384615384615384
F1 score: 0.6666666666666667

Ghoul
MultinomialNB
Accuracy: 0.7894736842105263
Precision:0.6470588235294118
Recall: 0.8461538461538461
F1 score: 0.7333333333333334

Total
Accuracy: 0.8260869565217391
Precision:0.7368421052631579
Recall: 0.7368421052631579
F1 score: 0.7368421052631579
```

Παρατηρουμε ότι το Accuracy της Naive Bayes μεθοδου είναι Accuracy= 0,8260869565217391

Εν κατακλειδι το καλυτερο Accuracy το πηραμε από την SVM μεθοδο και συγκεκριμενα με linear kernel (Accuracy = 0.8771929824561403).Η εξηγηση της αποδοσης αυτης που μπορει να δωθει είναι η γραμμικοτητα του μοντελου (linear) και στην λειτουργια της SVM μεθοδου που με την χρηση επιτυγχάνονται αποτελεσματα υψηλης αποδοσης