

# Project NN

## Classification

**BHMA 1:** Διάσπαση του dataset σε training and testing set (sklearn :: train\_test\_split)

**BHMA 2:** Χρήση Νευρωνικών Δικτύων με στόχο την δημιουργία και σύγκριση μοντέλων ταξινόμησης:

1. MLP Neural Networks με dense layers μονο: [https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/Dense](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense).
2. Χρήση activation functions ενδιάμεσα των layers όπως tanh, relu & leaky\_relu, ... etc.
3. Επειδή το πρόβλημα είναι binary classification τελευταίο activation function είναι αυτό του sigmoid, μετά το τελευταίο layer.

**BHMA 3:** Υπολογίστε τις μετρικές απόδοσης ταξινόμησης:

1. Precision: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html)
2. Recall: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html)
3. F1-score: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html)
4. Accuracy: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html)
5. ROC-AUC: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc\\_auc\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html)

**BHMA 4:** Συγκρίνετε αυτό μαζί με τα άλλα που έχετε ήδη σε bar plot, για να δούμε την αποτελεσματικότητα του νευρωνικού δικτύου σε αυτό το dataset.

Tutorials:

1. <https://www.geeksforgeeks.org/deep-learning/multi-layer-perceptron-learning-in-tensorflow/>
2. <https://javaeeeeee.medium.com/binary-classification-with-tensorflow-2-bc70f0cc7035>
3. <https://www.youtube.com/watch?v=BqjOc7iCut0>