# Заданне 1

Узяў прыведзены ў заданні датасэт, <https://archive.ics.uci.edu/ml/datasets/adult>​

На віндаус не працуе auto-sklearn, таму выкарыстоўваў h2o.ai і TPOT бібліятэкі.

У h2o перамог GBM\_1, (Gradient Boosting Machine) з рэзультатам accuracy = 0.926067

У ТРОТ перамог RandomForestClassifier, accuracy = 0.915770656359169

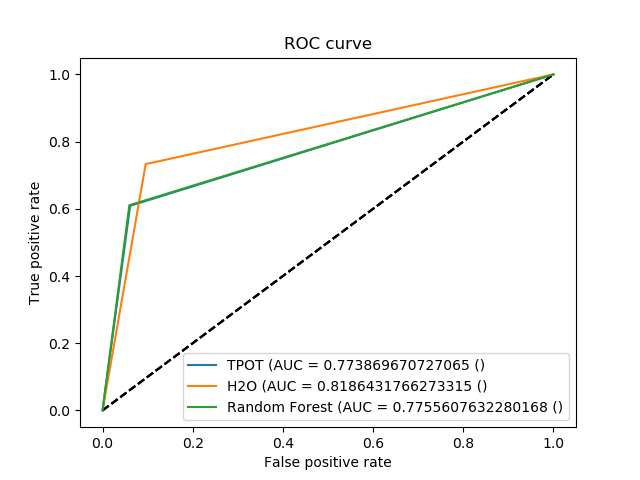
Дарэчы, у мяня ні ў адной з гэтых ліб не запрацаваў XGBoost. Але ён уплывае толькі на хуткасць, як я зразумеў, таму нічога асабліва страшнага.

Так атрымалася, што я не дачакаўся, пакуль дапрацуе ТРОТ і ўжо таксама скарыстаўся RandomForestClassifier з sklearn, бо гэта добрая мадэль для класіфікацыі і я ўжо яе выкарыстоўваў для лабы па спарку. Але ўсё ж я запусціў і на гэтым лесе невялічкі грыд па параметрах. Праўда, я забыў вывесці канчатковую accuracy, даруйце, калі ласка :(

Код можна паглядзець у архіве codebase.7z, тэчка task1(autoML). Каб запусціць усё адразу, быў прыдуманы скрыпт run\_whole\_stuff.py. Трэба, каб тэчка data з нашым наборам дадзеных знаходзілася ў той жа лакацыі, што і файлы пітона.

Мой камп’ютар не самы моцны, таму я перабіраў не самы шырокі спектр у аўта-мл бібліятэках. Калі хочацца прам пагрыдзіць, то трэба павялічыць параметры ў функцый run\_automl\_tpot і run\_automl\_h2o, тады грыд будзе больш дробнаячэістым.

Таксама ўвесь лог я захаваў у console\_output.txt, там можна глянуць прагрэс ліб і прыгожыя радкі, якімі я аддзяліў іх вывады адно ад аднаго.



Вось таксама крывая ROC curve, як я зразумеў, гэта і ё ROC-AUC. AUC указаны ў дужках.

Вынік такі, што н2о лепей за усіх, а мой класіфікатар паказаў той жа рэзультат, што і ТРОТ, нават не бачна сіні графік за зялёным. Пэўна, варта было ТРОТ даць больш пакаленняў, тады хутчэй за ўсё лес саступіў каму-небудзь яшчэ. Даруйце, так ужо атрымалася, мы ж тут не за скорамі ўсё ж сядзім, галоўнае, я зразумеў, як карыстацца лібай :)

# Заданне 2

Напісаў алгарытм Нэлдэра-Міда. Код у codebase.7z, Nelder-Mead\_algorithm.py.

Патэсціў на некалькіх простых функцыях накшталт x^2, x^4 + Const

Алгарытм збягаецца даволі хутка. Параметры канструктара класа:

dim=3, max\_iter=1000, distance\_f=0.000001, distance\_x=0.000001, alpha=1, beta=0.5, gamma=2

Крытэріі астанова:

1. max\_iter – максімальная колькасць ітэрацый
2. distance\_f – мінімальная дыстанцыя паміж max і min значэннямі функцыі
3. distance\_x – мінімальная дыстанцыя паміж вектарам Х, якому адпавядае min(f), і ўсімі астатнімі вектарамі