1.

X\_text = car\_data.loc[:, features]

X = pandas.get\_dummies(X\_text)

print(X.columns)

y = car\_data["class"]

**lisab kõik featurid ja seejärel viimane column on class ehk mudeldatav Y**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.3, shuffle=False)

Jagab andmed esimesed 70% õppimiseks ja viimased 30% tulemuse testimiseks. Tulemus pole

dtc=tree.DecisionTreeClassifier()

dtc.fit(X\_train, y\_train)

# puu on treenitud, vaatame kiirelt, kui hästi ta testandmeid klassifitseerib:

dtc.score(X\_test, y\_test)

Dtc.score näitab 68.5%, sõltuvalt kasutustegurist kuid üldiselt 68,5% mudel ei sobi ärifunktsioonitäitmiseks

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.3)

Valin nüüd segamini andmed, sain tulemuseks 95,1%. Tulemus on kõvasti kasvanud ja võib sobida kliendile. Üldislet 95% peetakse matemaatilises maailma sobivaks statistiliselt

y\_predicted = dtc.predict(X\_test)

precision, recall, \_, \_ = precision\_recall\_fscore\_support(

y\_predicted, y\_test, average=None, labels=["vgood"])

Precisions näitab ennustuste täpsust test andmetel, milleks on tihti kas 0,96 või 1. 1 näitab 100%

nnc = neural\_network.MLPClassifier(hidden\_layer\_sizes=(10,))

nnc.fit(X\_train, y\_train)

nnc.score(X\_test, y\_test)

Maximum iterations (200) reached and the optimization hasn't converged yet. Ei leidnud tulemust

nnc = neural\_network.MLPClassifier(hidden\_layer\_sizes=(10,), max\_iter=1000)

Valides maksimum arv iteratsioone leidis tulemuseks 97,4%

n\_features = X\_train.shape[1]

nnc\_shape = (n\_features, n\_features, 10)

nnc = neural\_network.MLPClassifier(hidden\_layer\_sizes=nnc\_shape, max\_iter=1000)

Saime täpsemaks mudelit, valin erinevate layerite arvuks võrde arvu mudeldataiv muutujaid. Ehk 21 muutujat. Mudeli täpsus tõusis 97,8% peale

y\_nnc = nnc.predict(X\_test)

precision, recall, \_, \_ = precision\_recall\_fscore\_support(y\_nnc, y\_test, average=None, labels=["vgood"])

Precision test andmete puhul tõusis 1’le, mis näitab ideaalset 100% katvust.

X\_balanced = pandas.concat([X1, X2r, X3r, X4r])

y\_balanced = pandas.concat([y1, y2r, y3r, y4r])

provides ära andmete duubeldamine mudeli täpsus kasvab. Score näitab 1 ja precision on ka 1 ehk 100%. Mudeldamine oli edukas ja vastavalt testandmetele sai eduka mudeli. Närvivõrguga näitas edukamat mudelit