# Viikko 40 -tehtävät

## Tehtävä 1

import numpy as np

import pandas as pd

from sklearn.metrics import mean\_absolute\_error, r2\_score, mean\_squared\_error

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

*# lue data sekä jaa X ja y*

df = pd.read\_csv('./work/viikko6/datasets/startup.csv')

X = df.iloc[:, :-1]

y = df.iloc[:, [-1]]

*# dummies*

X\_org = X

*# parempi tapa dymmy-muuttujille*

ct = ColumnTransformer(transformers=[('encoder',

OneHotEncoder(drop='first'), ['State'])], remainder='passthrough')

X = ct.fit\_transform(X)

*# Splitting the dataset into the Training set and Test set*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2,

                                                    random\_state = 0)

*# Training the Multiple Linear Regression model on the Training set*

model = LinearRegression()

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

*# Predicting the Test set results*

y\_pred = model.predict(X\_test)

*# Regression metrics*

mae=mean\_absolute\_error(y\_test, y\_pred)

r2=r2\_score(y\_test, y\_pred)

mea = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mea)

print(f'r2:  {round(r2,4)}')

print(f'mae: {round(mae,4)}')

print(f'rmse: {round(rmse,4)}')

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus, numero

Kuvaus luotu automaattisesti

## Tehtävä 2

df\_new\_company = pd.read\_csv('./work/viikko6/datasets/new\_company\_ct.csv')

df\_new\_company = ct.transform(df\_new\_company)

y\_comp = model.predict(df\_new\_company)

print (f'Uuden yrityksen voitto: {y\_comp[0]}, todellinen: ')

print (f'{df.iloc[0:1,-1].values[0]}')

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus

Kuvaus luotu automaattisesti

## Tehtävä 3

startup\_train.py

import numpy as np

import pandas as pd

from sklearn.metrics import mean\_absolute\_error, r2\_score, mean\_squared\_error

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import StandardScaler

import pickle

*# lue data sekä jaa X ja y*

df = pd.read\_csv('./work/viikko6/datasets/startup.csv')

X = df.iloc[:, :-1]

y = df.iloc[:, [-1]]

*# dummies*

X\_org = X

*# parempi tapa dymmy-muuttujille*

ct = ColumnTransformer(transformers=[('encoder',

OneHotEncoder(drop='first'), ['State'])], remainder='passthrough')

X = ct.fit\_transform(X)

*# Splitting the dataset into the Training set and Test set*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2,

                                                    random\_state = 0)

*# skaalataan data*

scaler\_x = StandardScaler()

X\_train = scaler\_x.fit\_transform(X\_train)

X\_test = scaler\_x.transform(X\_test)

scaler\_y = StandardScaler()

y\_train  = scaler\_y.fit\_transform(y\_train)

*# Training the Multiple Linear Regression model on the Training set*

model = LinearRegression()

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

*# Predicting the Test set results*

y\_pred = scaler\_y.inverse\_transform(model.predict(X\_test))

*# Regression metrics*

mae=mean\_absolute\_error(y\_test, y\_pred)

r2=r2\_score(y\_test, y\_pred)

mea = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mea)

print(f'r2:  {round(r2,4)}')

print(f'mae: {round(mae,4)}')

print(f'rmse: {round(rmse,4)}')

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus, numero

Kuvaus luotu automaattisesti

*# tallennetaan malli levylle*

with open('startup-model.pickle', 'wb') as f:

    pickle.dump(model, f)

*# tallennetaan encoderi*

with open('startup-ct.pickle', 'wb') as f:

    pickle.dump(ct, f)

*# tallennetaan skaaleri x*

with open('startup-scaler-x.pickle', 'wb') as f:

        pickle.dump(scaler\_x, f)

*# tallennetaan skaaleri y*

with open('startup-scaler-y.pickle', 'wb') as f:

        pickle.dump(scaler\_y, f)

startup\_predict.py

import pandas as pd

import pickle

*# ladataan malli levyltä*

with open('startup-model.pickle', 'rb') as f:

    model = pickle.load(f)

*# ladataan enkooderi*

with open('startup-ct.pickle', 'rb') as f:

    ct = pickle.load(f)

*# lue skaalerit*

with open('startup-scaler-x.pickle', 'rb') as f:

    scaler\_x = pickle.load(f)

with open('startup-scaler-y.pickle', 'rb') as f:

    scaler\_y = pickle.load(f)

*# ennusta uudella datalla*

Xnew = pd.read\_csv('./work/viikko6/datasets/new\_company\_ct.csv')

Xnew\_org = Xnew

Xnew=ct.transform(Xnew)

*# skaalaa Xnew*

Xnew = scaler\_x.transform(Xnew)

*# ennusta ja aja inverse scaler*

ynew = scaler\_y.inverse\_transform(model.predict(Xnew))

*#ynew = model.predict(Xnew)*

for i in range (len(ynew)):

    print (f'{Xnew\_org.iloc[i]}\nVoitto: {ynew[i][0]}\n')

coef = model.coef\_

inter = model.intercept\_

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus

Kuvaus luotu automaattisesti

## Tehtävä 4

housing\_train.py

import numpy as np

import pandas as pd

from sklearn.metrics import mean\_absolute\_error, r2\_score, mean\_squared\_error

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import StandardScaler

import pickle

*# lue data sekä jaa X ja y*

df = pd.read\_csv('./work/viikko6/datasets/housing.csv')

y = df['median\_house\_value']

X = df[['longitude', 'latitude', 'housing\_median\_age', 'total\_rooms', 'total\_bedrooms', 'median\_income', 'ocean\_proximity']]

*# count of Nan values*

print(X.isnull().sum())

*# unique ocean\_proximity values*

print(X['ocean\_proximity'].unique())

*# replace NaN values with mean*

X['total\_bedrooms'].fillna(X['total\_bedrooms'].mean(), inplace=True)

*# dummies*

X\_org = X

ct = ColumnTransformer(transformers=[('encoder',

OneHotEncoder(drop='first'), ['ocean\_proximity'])], remainder='passthrough')

X = ct.fit\_transform(X)

*# Splitting the dataset into the Training set and Test set*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2,

                                                    random\_state = 0)

*# Training the Multiple Linear Regression model on the Training set*

model = LinearRegression()

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

*# Predicting the Test set results*

y\_pred = model.predict(X\_test)

*# Regression metrics*

mae=mean\_absolute\_error(y\_test, y\_pred)

r2=r2\_score(y\_test, y\_pred)

mea = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mea)

print(f'r2:  {round(r2,4)}')

print(f'mae: {round(mae,4)}')

print(f'rmse: {round(rmse,4)}')

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus, numero

Kuvaus luotu automaattisesti

*# tallennetaan malli levylle*

with open('housing-model.pickle', 'wb') as f:

    pickle.dump(model, f)

*# tallennetaan encoderi*

with open('housing-ct.pickle', 'wb') as f:

    pickle.dump(ct, f)

housing\_predict.py

import pandas as pd

import pickle

*# ladataan malli levyltä*

with open('housing-model.pickle', 'rb') as f:

    model = pickle.load(f)

*# ladataan enkooderi*

with open('housing-ct.pickle', 'rb') as f:

    ct = pickle.load(f)

*# ennusta uudella datalla*

Xnew = pd.read\_csv('./work/viikko6/datasets/new\_house\_ct.csv')

Xnew\_org = Xnew

Xnew=ct.transform(Xnew)

*# ennustetaan*

ynew = model.predict(Xnew)

for i in range (len(ynew)):

    print (f'{Xnew\_org.iloc[i]}\nPredicted value: {ynew[i]}\n')

coef = model.coef\_

inter = model.intercept\_

print (f'Coefficients: {coef}\nIntercept: {inter}')

Kuva, joka sisältää kohteen teksti, kuvakaappaus, ruokalista

Kuvaus luotu automaattisesti

Tässä mallissa r2 ja mae ovat pienempiä verrattuna viime viikon malliin. Voidaan siis sanoa usean muuttujan regressiomalli on tarkempi.