# Viikko 41 -tehtävät

## Tehtävä 1, 2 ja 3

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, accuracy\_score

from sklearn.metrics import precision\_score, recall\_score

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import StandardScaler

df = pd.read\_csv('./work/viikko7/datasets/titanic-class-age-gender-survived.csv')

*#X = df[['Age', 'Gender']] # tehtävä 2*

X = df[['PClass', 'Age', 'Gender']] *# tehtävä 3*

y = df['Survived'] *# selvitettävä muuttuja*

*# dummies*

X\_org = X

ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(drop='first'), ['PClass', 'Gender'])], 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)

*# Scaling*

scaler\_x = StandardScaler()

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

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

*# Fitting Logistic Regression to the Training set*

model = LogisticRegression()

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

*# Predicting the Test set results*

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

y\_pred\_pros = model.predict\_proba(X\_test) *# probability estimates*

*# Making the Confusion Matrix*

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

*# Accuracy*

acc = accuracy\_score(y\_test, y\_pred)

*# Precision*

pc = precision\_score(y\_test, y\_pred)

*# Recall*

rc = recall\_score(y\_test, y\_pred)

*# Visualizing the Confusion Matrix*

tn, fp, fn, tp = cm.ravel()

ax = plt.axes()

sns.heatmap(cm, annot=True, fmt="g", ax=ax)

ax.set\_title(f'Confusion Matrix\nAccuracy: {acc:.5f}\nPrecision: {pc:.5f}\nRecall: {rc:.5f}')

plt.show()

*# Predicting new data*

Xnew = pd.read\_csv('./work/viikko7/datasets/titanic-new.csv')

Xnew\_org = Xnew

Xnew = ct.transform(Xnew)

Xnew = scaler\_x.transform(Xnew)

ynew = model.predict(Xnew)

ynew\_pros = model.predict\_proba(Xnew)

for i in range(len(Xnew\_org)):

    print(f'[{Xnew\_org.iloc[i, 0]}, {Xnew\_org.iloc[i, 1]}, {Xnew\_org.iloc[i, 2]}] Survived prediction: {ynew[i]}')

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus

Kuvaus luotu automaattisesti

Selittävä muuttuja Age:

Kuva, joka sisältää kohteen teksti, kuvakaappaus, diagrammi, muotoilu

Kuvaus luotu automaattisesti

Selittävät muuttujat Age, Gender:

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Suorakaide, muotoilu

Kuvaus luotu automaattisesti

Selittävät muuttujat Age, Gender, PClass:

Kuva, joka sisältää kohteen teksti, kuvakaappaus, diagrammi, muotoilu

Kuvaus luotu automaattisesti

Selkeästi tarkin malli saadaan kun käytetään kolmea selittävää muuttujaa.

## Tehtävä 4

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, accuracy\_score

from sklearn.metrics import precision\_score, recall\_score

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import StandardScaler

df = pd.read\_csv('./work/viikko7/datasets/diabetes.csv')

X = df.drop('Outcome', axis=1) *# selittävät muuttujat*

y = df['Outcome'] *# selitettävä muuttuja*

*# 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)

*# Scaling*

scaler\_x = StandardScaler()

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

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

*# Fitting Logistic Regression to the Training set*

model = LogisticRegression()

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

*# logistic regression coefficients*

print('Coefficients: \n', model.coef\_)

*# logistic regression intercept*

print('Intercept: \n', model.intercept\_)

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti

Kuvaus luotu automaattisesti

*# Predicting the Test set results*

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

y\_pred\_pros = model.predict\_proba(X\_test) *# probability estimates*

*# Making the Confusion Matrix*

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

*# Accuracy*

acc = accuracy\_score(y\_test, y\_pred)

*# Precision*

pc = precision\_score(y\_test, y\_pred)

*# Recall*

rc = recall\_score(y\_test, y\_pred)

*# Visualizing the Confusion Matrix*

tn, fp, fn, tp = cm.ravel()

ax = plt.axes()

sns.heatmap(cm, annot=True, fmt="g", ax=ax)

ax.set\_title(f'Confusion Matrix\nAccuracy: {acc:.5f}\nPrecision: {pc:.5f}\nRecall: {rc:.5f}')

plt.show()

Kuva, joka sisältää kohteen teksti, kuvakaappaus, diagrammi, muotoilu

Kuvaus luotu automaattisesti

*# Predicting new data*

Xnew = pd.read\_csv('./work/viikko7/datasets/diabetes-new.csv')

Xnew\_org = Xnew

Xnew = scaler\_x.transform(Xnew)

ynew = model.predict(Xnew)

ynew\_pros = model.predict\_proba(Xnew)

for i in range(len(Xnew)):

    print(f'{ynew[i]} {ynew\_pros[i]}')

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

Kuvaus luotu automaattisesti