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)}')
```

```
r2:  0.9347
mae: 7514.2937
rmse: 9137.9902
```

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]}')
```

```
Uuden yrityksen voitto: [192919.5753746], todellinen:
192261.83
```

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)}')
```

```
r2:  0.9347
mae: 7514.2937
rmse: 9137.9902
```

```
# 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_
```

```
R&D Spend          165349.2
Administration     136897.8
Marketing Spend    471784.1
State              New York
Name: 0, dtype: object
Voitto: 2.062813154946949
```

Tehtävä 4

housing_train.py

```python
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)}')
```

```
r2:  0.6131
mae: 51619.5098
rmse: 71023.6036
```

```
# 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)
```

```python
# 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}')
```

```
longitude             -122.29
latitude                37.81
housing_median_age       41.0
total_rooms             880.0
total_bedrooms          129.0
median_income          8.3252
ocean_proximity       NEAR BAY
Name: 0, dtype: object
Predicted value: 420632.1874042561

longitude             -122.28
latitude                37.81
housing_median_age       21.0
total_rooms            7099.0
total_bedrooms         1106.0
median_income          8.3014
ocean_proximity       NEAR BAY
Name: 1, dtype: object
Predicted value: 403110.96309793345

longitude             -122.27
latitude                37.81
housing_median_age       52.0
total_rooms            1467.0
total_bedrooms          190.0
 ...
Coefficients: [-3.84014116e+04  1.74830165e+05  1.85326464e+03  1.15243163e+04
 -2.41752365e+04 -2.20548948e+04  1.09689857e+03 -1.24847235e+01
  8.52386080e+01  4.10556637e+04]
Intercept: -2090493.852951864
```

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