# 5G00ET68-3001 Data-analyysi ja tekoälyn perusteet

T1-T2

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

df = pd.read_csv('startup.csv')

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

#dummies_state = pd.get_dummies(X['State'], drop_first=True)
#X = X.join(dummies_state)
#X.drop('State', inplace=True, axis=1)


X_org = X
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(drop='first'), ['State'])], remainder='passthrough')
X = ct.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)


model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print (f'r2: {r2}')
print (f'mae: {mae}')
print (f'rmse: {rmse}')

df_new_company = pd.read_csv('new_company_ct.csv')
df_new_company = ct.transform(df_new_company)

y_new_company = model.predict(df_new_company)
print (f'Uuden yrityksen voitto: {y_new_company}')
```

| | | | |
|---|---|---|---|
| ct | compose._column_transformer.ColumnTransformer | 1 | ColumnTransformer object of sklearn.compose._column_tr |
| df | DataFrame | (50, 5) | Column names: R&D Spend, Administration, Marketing Spe |
| df_new_company | Array of float64 | (1, 5) | [[0.000000e+00 1.000000e+00 1.653492e+05 1.368978e+05 |
| mae | float64 | 1 | 7514.293659640891 |
| model | linear_model._base.LinearRegression | 1 | LinearRegression object of sklearn.linear_model._base |
| mse | float64 | 1 | 83502864.03257468 |
| r2 | float64 | 1 | 0.9347068473282446 |
| rmse | float64 | 1 | 9137.990152794797 |
| X | Array of float64 | (50, 5) | [[0.0000000e+00 1.0000000e+00 1.6534920e+05 1.3689780e |
| X_org | DataFrame | (50, 4) | Column names: R&D Spend, Administration, Marketing Spe |
| X_test | Array of float64 | (10, 5) | [[1.0000000e+00 0.0000000e+00 6.6051520e+04 1.8264556e |
| X_train | Array of float64 | (40, 5) | [[1.0000000e+00 0.0000000e+00 5.5493950e+04 1.0305749e |
| y | DataFrame | (50, 1) | Column names: Profit |
| y_new_company | Array of float64 | (1, 1) | [[192919.57537463]] |
| y_pred | Array of float64 | (10, 1) | [[103015.20159796]<br>[132582.27760815]] |
| y_test | DataFrame | (10, 1) | Column names: Profit |
| y_train | DataFrame | (40, 1) | Column names: Profit |

```
r2: 0.9347068473282446
mae: 7514.293659640891
rmse: 9137.990152794797
Uuden yrityksen voitto: [[192919.57537463]]
```

# 5G00ET68-3001 Data-analyysi ja tekoälyn perusteet

Startup_predict

```
import pandas as pd
import pickle

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

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

Xnew = pd.read_csv('new_company_ct.csv')
Xnew_org = Xnew
Xnew = ct.transform(Xnew)
Ynew = model.predict(Xnew)

for i in range (len(Ynew)):
    print(f'{Xnew_org.iloc[i]}\nVoitto: {Ynew[i][0]}\n')
```

```
R&D Spend          165349.2
Administration     136897.8
Marketing Spend    471784.1
State              New York
Name: 0, dtype: object
Voitto: 192919.5753746262
```

# 5G00ET68-3001 Data-analyysi ja tekoälyn perusteet

Startup_train

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
import pickle


df = pd.read_csv('startup.csv')

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


X_org = X
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(drop='first'), ['State'])], remainder='passthrough')
X = ct.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)


model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print (f'r2: {r2}')
print (f'mae: {mae}')
print (f'rmse: {rmse}')

# tallentaan 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)
```