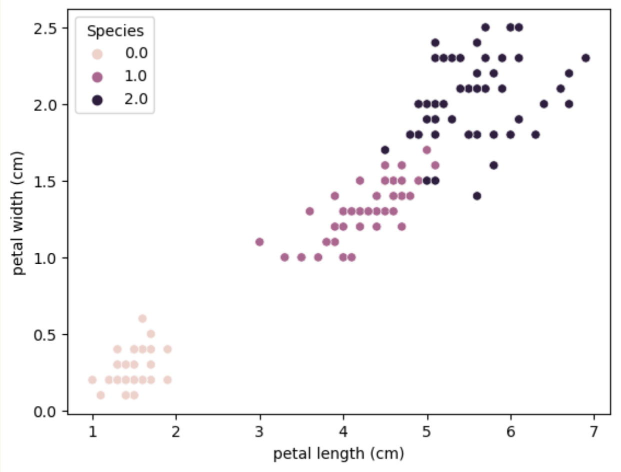
Viikko 43 -tehtävät

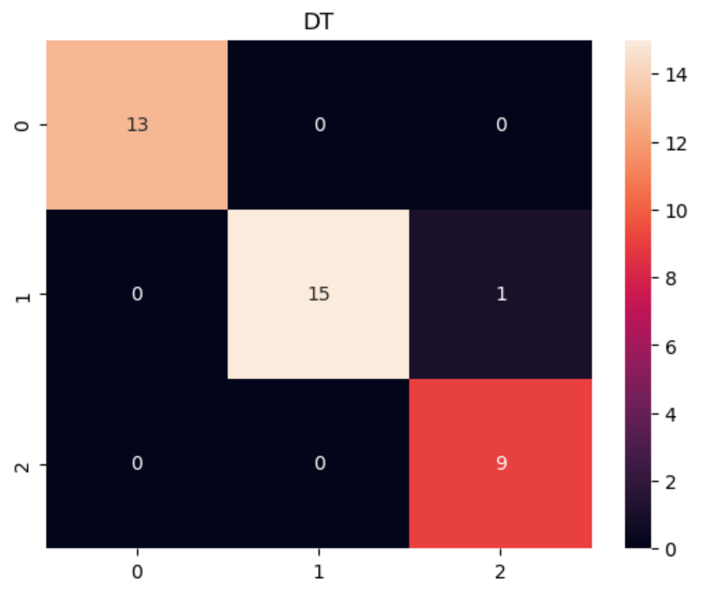
Tehtävä 1

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
recall_score
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import export_graphviz
import graphviz
from sklearn import tree

df = pd.read_csv('./work/viikko8/datasets/iris.csv')

sns.scatterplot(x='petal length (cm)', y='petal width (cm)', hue='Species', data=df)
plt.show()
```



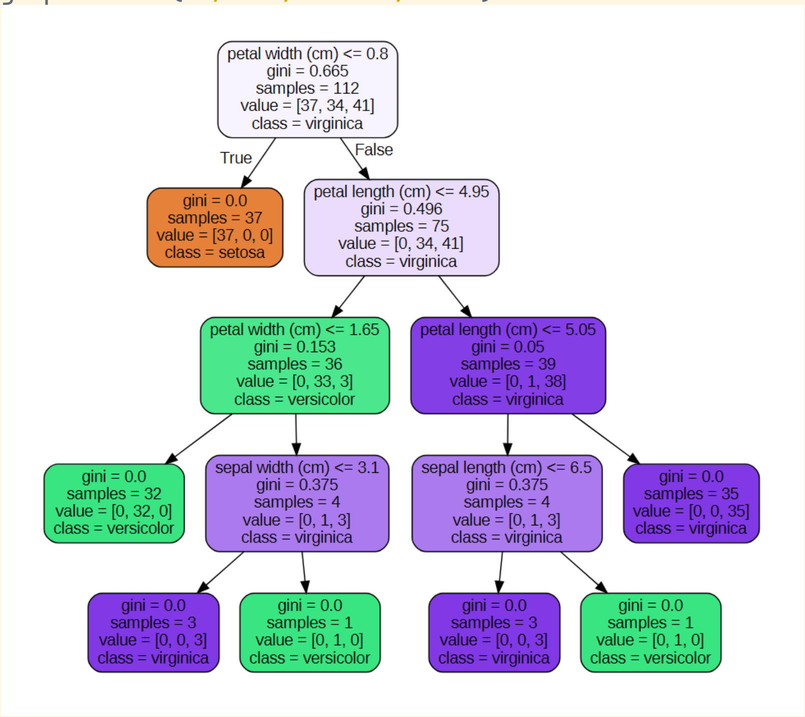
```
X = df.iloc[:, 0:4]
y = df.iloc[:, [4]]
columns = X.columns

# 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 = 0.25,
random_state = 0)
```

```
# Training the Decision Tree Classification model
model = tree.DecisionTreeClassifier(max_depth=4, criterion='gini')
model.fit(X_train, y_train)

mfi = model.feature_importances_

# Predicting the Test set results
y_pred = model.predict(X_test)
y_pred_pros = model.predict_proba(X_test)

# Making the Confusion Matrix and accuracy_score
cm = confusion_matrix(y_test, y_pred)

ax = plt.axes()
sns.heatmap(cm, annot=True, fmt='g', ax=ax)
ax.set_title("DT")
plt.show()

# Calculate accuracy score
score = accuracy_score(y_test, y_pred)
print(f"Accuracy score: {score}")
```



```
Accuracy score: 0.9736842105263158
```

```
# Create dot file for graphviz visualization
dot_data = export_graphviz(
            model,
            out_file =  None,
            feature_names = columns,
            class_names = df['Class'].unique(),
            filled = True,
            rounded = True)

graph = graphviz.Source(dot_data)
graph.render('./work/viikko8/iris')
```



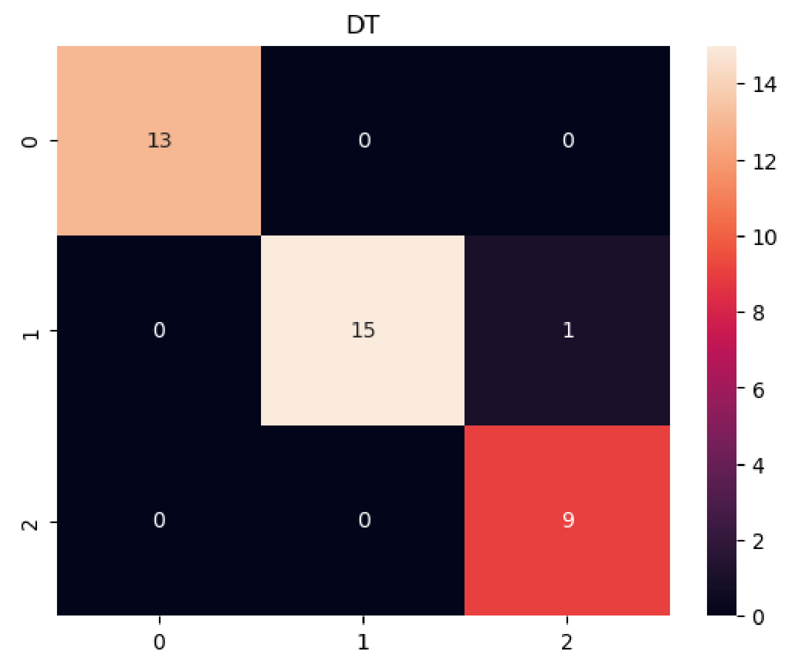
```
# predict new
df_new = pd.read_csv('./work/viikko8/datasets/new-iris.csv')

predictions = model.predict(df_new)
print(predictions)
```

```
[0. 2. 1.]
```

Tehtävä 2

Sama koodi kuin tehtävässä 1. Vaihdetaan vain malli:

```
# Training the Decision Tree Classification model
model = ensemble.RandomForestClassifier(max_depth=5)
```



```
Accuracy score:
0.9736842105263158
```

```
# predict new
df_new = pd.read_csv('./work/viikko8/datasets/new-iris.csv')

predictions = model.predict(df_new)
print(predictions)
```

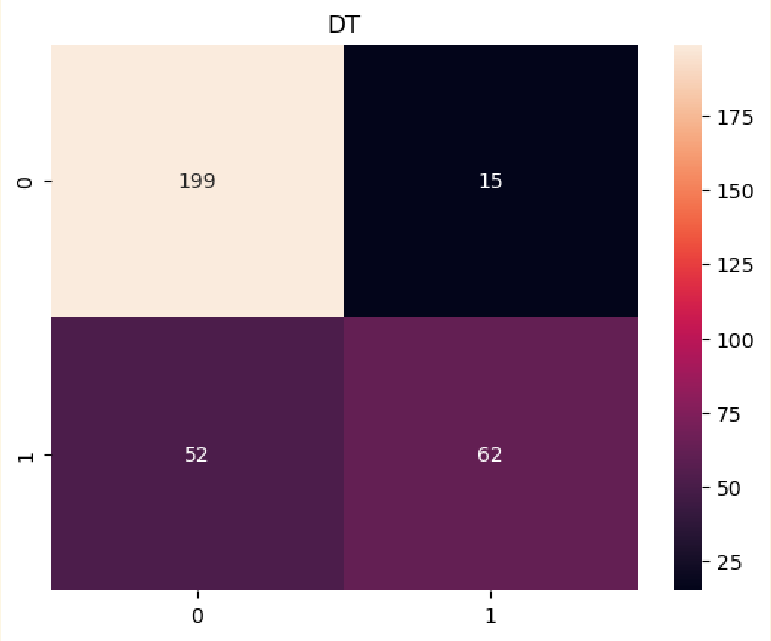
```
[0. 2. 1.]
```

Tehtävä 3

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
recall_score
import matplotlib.pyplot as plt
import seaborn as sns
```

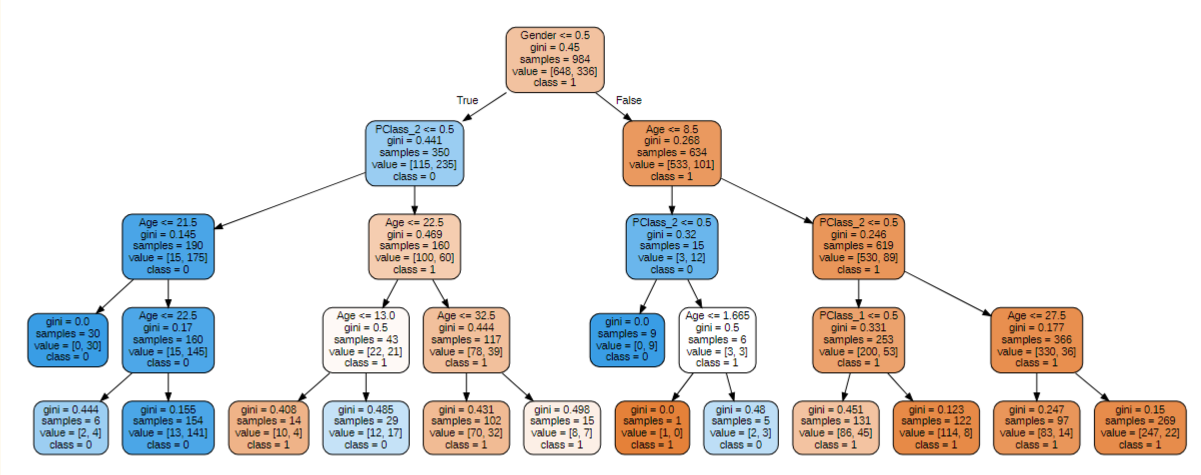
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.tree import export_graphviz
import graphviz
from sklearn import tree

df = pd.read_csv('./work/viikko8/datasets/titanic.csv')

X = df.iloc[:, 0:3]
y = df.iloc[:, [3]]
columns = ['PClass_1', 'PClass_2', 'Gender', 'Age']

print(columns)

# 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 = 0.25,
random_state = 0)

# Training the Decision Tree Classification model
model = tree.DecisionTreeClassifier(max_depth=4, criterion='gini')
model.fit(X_train, y_train)

mfi = model.feature_importances_

# Predicting the Test set results
y_pred = model.predict(X_test)
y_pred_pros = model.predict_proba(X_test)

# Making the Confusion Matrix and accuracy_score
```

```
cm = confusion_matrix(y_test, y_pred)


ax = plt.axes()
sns.heatmap(cm, annot=True, fmt='g', ax=ax)
ax.set_title("DT")
plt.show()


# Calculate accuracy score
score = accuracy_score(y_test, y_pred)
print(f"Accuracy score: {score}")
```

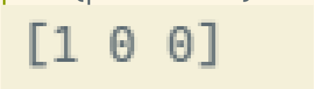


```
Accuracy score: 0.7957317073170732


# Create dot file for graphviz visualization
dot_data = export_graphviz(
            model,
            out_file =  None,
            feature_names = columns,
            class_names = df['Survived'].astype(str).unique(),
            filled = True,
            rounded = True)


graph = graphviz.Source(dot_data)
graph.render('./work/viikko8/titanic')
```

```
# predict new
df_new = pd.read_csv('./work/viikko8/datasets/titanic-new.csv')
df_new = ct.transform(df_new)
```



```
predictions = model.predict(df_new)
print(predictions)
```



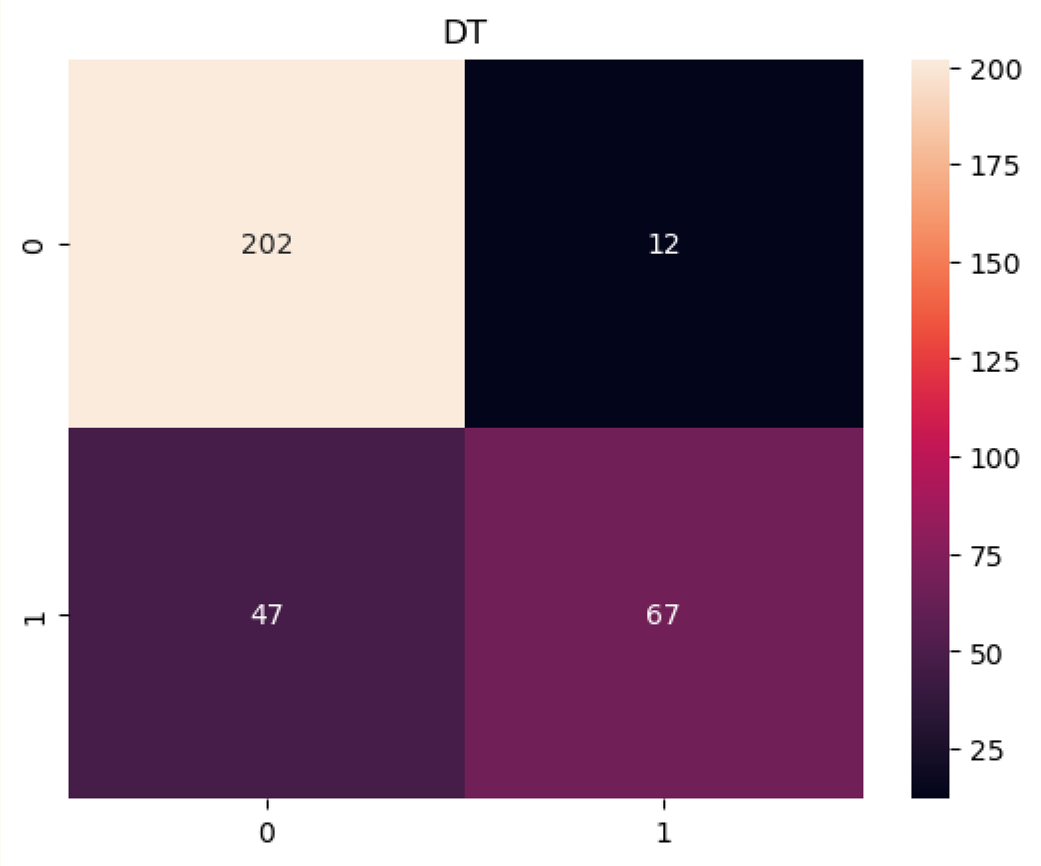
```
[1 0 0]
```

Tehtävä 4

Sama koodi kuin tehtävässä 3. Vaihdetaan vain malli:

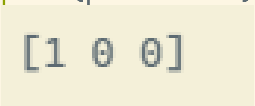
```
# Training the Decision Tree Classification model
model = ensemble.RandomForestClassifier(max_depth=5)

# Making the Confusion Matrix and accuracy_score
cm = confusion_matrix(y_test, y_pred)

ax = plt.axes()
sns.heatmap(cm, annot=True, fmt='g', ax=ax)
ax.set_title("DT")
plt.show()

# Calculate accuracy score
score = accuracy_score(y_test, y_pred)
```

```
print(f"Accuracy score: {score}")
```



```
Accuracy score: 0.8201219512195121
```

```
# predict new
df_new = pd.read_csv('./work/viikko8/datasets/titanic-new.csv')
df_new = ct.transform(df_new)

predictions = model.predict(df_new)
print(predictions)
```



```
[1 0 0]
```

Satunnaismetsällä saadaan hieman parempi accuracy score, mutta ennustukset ovat kuitenkin samat uudelle datalle.