WUM_P2

March 30, 2021

[77]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
```

[93]:
```
df  = pd.read_csv('congressional_voting_dataset.csv')
```

[94]:
```
change_dict = {"y": 1, "n": -1, "?": 0, "republican": 0, "democrat": 1 }
df.replace(change_dict, inplace=True)
```

0.1 Wybór column i wierszy

Usuwam 6 kongresmenów którzy zabrali głos mniej niż 11 razy (na 16 głosowań). Oraz 3 głosowania które słabo dzielą nasz zbiór

[95]:
```
df_count = np.apply_along_axis(sum, 1, abs(df))
indexes = df[df_count < 11].index
indexes = np.array(indexes)
print(df.shape)
df.drop(indexes, axis=0, inplace=True)
print(df.shape)
```

```
(435, 17)
(429, 17)
```

[96]:
```
df.columns
```

[96]:
```
Index(['handicapped_infants', 'water_project_cost_sharing',
       'adoption_of_the_budget_resolution', 'physician_fee_freeze',
       'el_salvador_aid', 'religious_groups_in_schools',
       'anti_satellite_test_ban', 'aid_to_nicaraguan_contras', 'mx_missile',
       'immigration', 'synfuels_corporation_cutback', 'education_spending',
       'superfund_right_to_sue', 'crime', 'duty_free_exports',
       'export_administration_act_south_africa', 'political_party'],
      dtype='object')
```

[97]:
```
col_ls = ['water_project_cost_sharing', 'immigration',␣
 ↪'export_administration_act_south_africa', ]
df.drop(col_ls, axis=1, inplace=True, errors='ignore')
print(df.shape)
```

```
(429, 14)
```

0.2 Modelowanie

[152]:
```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier,plot_tree
from sklearn.model_selection import cross_val_score
from sklearn.naive_bayes import GaussianNB
```

[105]:
```
X = df.drop('political_party', axis=1)
y = df['political_party']
```

[106]:
```
X_train, X_val, y_train, y_val = train_test_split(
    X, y, stratify=y, test_size=0.3, random_state=42
)
X_val, X_test, y_val, y_test = train_test_split(
    X_val, y_val, stratify=y_val, test_size=0.3, random_state=42
)
print(X_train.shape, X_val.shape, X_test.shape)
```

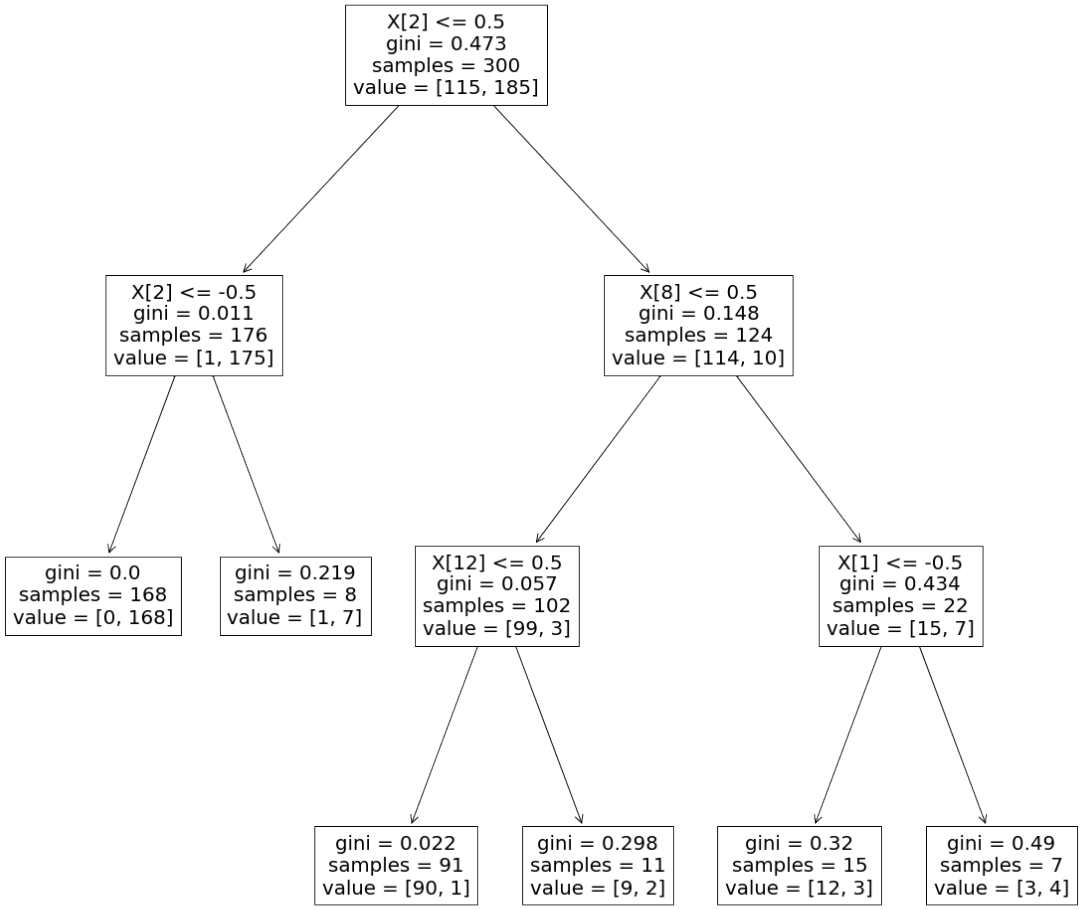
```
(300, 13) (90, 13) (39, 13)
```

0.2.1 Tree

[146]:
```
tree = DecisionTreeClassifier(min_samples_leaf=5, max_depth=3)
tree.fit(X_train, y_train)
tree.score(X_test, y_test)
```

[146]:
```
0.9487179487179487
```

[147]:
```
plt.figure(figsize=(20,20))
plot_tree(tree)
plt.show()
```



[149]:
```
X_tt = pd.concat((X_train, X_test))
y_tt = np.concatenate((y_train, y_test), axis=0)
cross_val_score(tree, X_tt, y_tt)
```

[149]:
```
array([0.97058824, 0.94117647, 0.94117647, 0.97058824, 0.95522388])
```

[150]:
```
tree.score(X_val, y_val)
```

[150]:
```
0.9777777777777777
```

0.2.2 Bayes

[154]:
```
nb = GaussianNB()
nb.fit(X_train, y_train)
nb.score(X_test, y_test)
```

[154]:
```
0.9230769230769231
```

[155]:
```
cross_val_score(nb, X_tt, y_tt)
```

[155]:
```
array([0.94117647, 0.97058824, 0.92647059, 0.92647059, 0.94029851])
```