Данила Борисович Шмаков 181-351

Задание 4

Нужно перевести качественные в количественные признаки.

```
le = LabelEncoder()
data['proto'] = le.fit_transform(data['proto'])
data['conn_state'] = le.fit_transform(data['conn_state'])
dtype = float
print(data)
```

Далее высчитываем доверительные интервалы для каждого количественного признака и удаляем аномалии (значения вне интервалов).

```
trust = dict()

for index in quantity:
     trust[index] = [np.mean(data[index]) - (3 * np.std(data[index])),
                     np.mean(data[index]) + (3 * np.std(data[index]))]
     data.loc[(data[index] > trust[index][1]) | (data[index] <
trust[index][0]), index] = np.nan

data = data.dropna()
```

Преобразуем метку в количественный признак

```
le = LabelEncoder()
le.fit(labels.label)
labels['label'] = le.transform(labels.label)
data = data.assign(label=labels['label'])
```

Нормировать данные нет необходимости.

Создаем и обучаем классификатор

```
points_train, points_test, labels_train, labels_test =
train_test_split(data.iloc[:, :-1], data['label'],

test_size=0.25,random_state=0)

dtc = DecisionTreeClassifier(criterion='entropy', max_depth=4)
dtc.fit(points_train, labels_train)
prediction = dtc.predict(points_test)
```

Создаем новую пробную точку

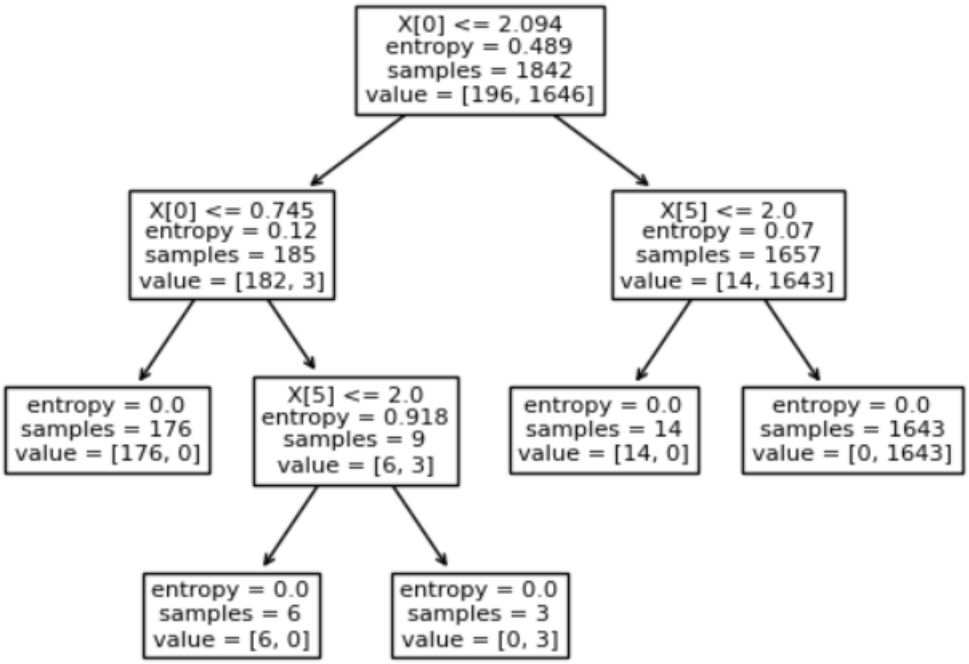
```
test_point = {'duration': [2.71], 'orig_bytes': [3], 'orig_pkts': [3],
test_point = {'duration': [2.71], 'orig_bytes': [20], 'orig_pkts': [3],
'resp_bytes': [21], 'resp_pkts': [3],
              'proto': [1], 'conn_state': [0]}

data_new = pd.DataFrame(test_point)
```

Проверяем входит ли в доверительные интервалы точка и скармливаем классификатору

```
if funcs.check_data(data_new, trust, quantity):
    prediction = dtc.predict(data_new)
    data_new = data_new.assign(label=prediction)
    data = data.append(data_new, ignore_index=True)
```

Рисуем красивую картинку дерева

```
plt.figure()
tree.plot_tree(dtc)
plt.show()
```



Выводим оценку

```
print('ОЦЕНКА: ', format(dtc.score(points_test, labels_test)))
```

```
ОЦЕНКА:  1.0
```