|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА (ИУ12)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

**Отчет**

**Название**: Работа в Pandas на примере задачи «Titanic». Задача 2.

**Дисциплина:** Введение в искусственный интеллект

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ12-11М |  |  | Д.В. Кузнецов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Д.Ю. Евсюков |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цели практики:**

• обучение решающих деревьев

• нахождение наиболее важных для них признаков

**Решение задач**

Импорт необходимых библиотек

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

import matplotlib.pyplot as plt

from sklearn.tree import plot\_tree

from sklearn.tree import export\_graphviz

1. Загрузка выборки из файла titanic.csv с помощью пакета Pandas

data = pd.read\_csv('titanic.csv')

1. Выбор четырех признаков: класс пассажира (Pclass), цену билета (Fare), возраст пассажира (Age) и его пол (Sex).

selected\_features = ['Pclass', 'Fare', 'Age', 'Sex']

data = data[selected\_features + ['Survived']]

1. Учет строкового значения признака Sex и преобразование в числовое.

data['Sex'] = data['Sex'].apply(lambda x: 1 if x == 'female' else 0)

1. Выделение целевой переменной Survived.

y = X['Survived']

X = X.drop('Survived', axis=1)

1. Удаление объектов с пустыми признаками.

X = data.dropna()

1. Обучение решающего дерева с параметром random\_state=241.

clf = DecisionTreeClassifier(random\_state=241)

clf.fit(X, y)

1. Вычисление важности признаков и выделение двух наиболее важных.

importances = clf.feature\_importances\_

features\_indices = importances.argsort()

print("Признак, Вес")

top\_features\_indices = importances.argsort()[-2:][::-1]

top\_features = [selected\_features[i] for i in top\_features\_indices]

print(f'\nНаиболее важные признаки: {", ".join(top\_features)}')

Ответ:

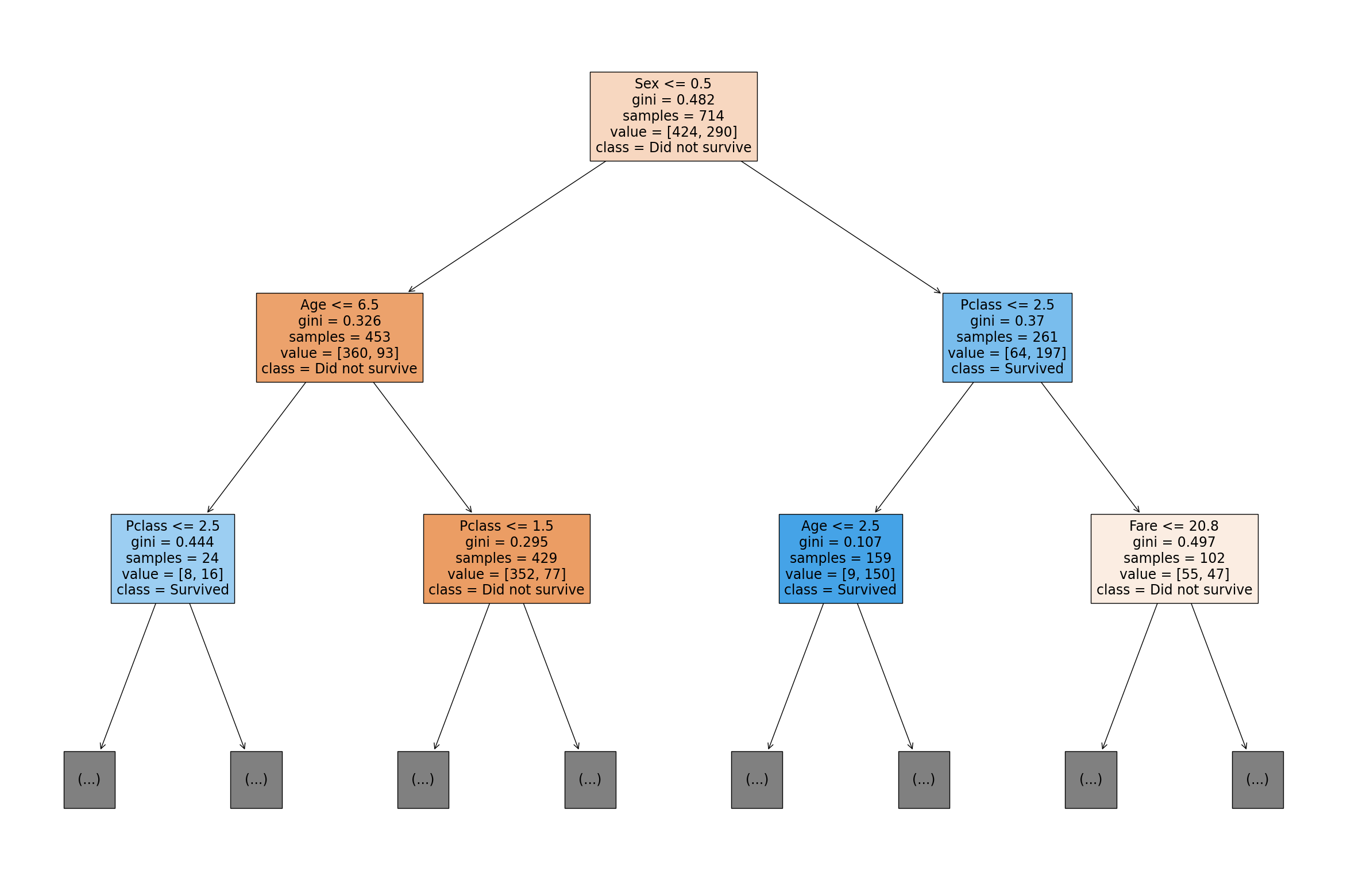
*Наиболее важные признаки: Sex, Fare*

1. Вывод решающего дерева.

plt.figure(figsize=(30,20))

plot\_tree(clf, feature\_names=selected\_features, filled=True, class\_names=['Did not survive', 'Survived'], max\_depth=2)

plt.show()



**Вывод**

В данной работе были использованы данные о пассажирах Титаника с применением библиотеки scikit-learn для решения задачи классификации - предсказания выживших после крушения корабля. С помощью решающих деревьев был разработан алгоритм, основанный на простых решающих правилах, что позволило получить интерпретируемую модель. В ходе работы были выбраны и обработаны четыре признака пассажиров, а также удалены записи с пропущенными значениями, что дало возможность обучить решающее дерево и вычислить важности признаков для предсказания выживших. Таким образом, работа позволила создать модель, способную предсказывать выживших пассажиров Титаника на основе их характеристик.