Лабораторная работ 7.
Классификация (наивный бесовский метод, метод деревьев).

Цель: ознакомиться с методами классификации Sklearn (наивный бесовский метод и метод деревьев).
Задачи:

1. Загрузить пред обработанные данные;
2. Выделить метки и признаки, разбив на обучающую и тестовую выборки;
3. Провести классификацию наблюдений;
4. Отобразить дерево.

Пример выполнения лабораторной работы.

Выведем первые пять строк из подготовленного датасета.

```
import pandas as pd
import numpy as np
df = pd.read_csv('lab7.csv',encoding='cp1251')
print(df.head(5))
```

Результат:

	id	gender	age	income	spending_rating
0	1	1	19	15	39
1	2	1	21	15	81
2	3	0	20	16	6
3	4	0	23	16	77
4	5	0	31	17	40

Мужской пол – 1, женский – 0.
Выделим данные и метки признаков и разберём выборку на обучающую и тестовую пропорции 50/50 используя sklearn.model_selection.train_test_split и зафиксируем random_state=10.

```
df = pd.read_csv('lab7.csv',encoding='cp1251')
y = df['gender'].astype(int)
X = df.drop('gender', axis=1)
X_train, X_valid, y_train, y_valid = train_test_split(X, y,
train_size=0.5, random_state=10)
print(X_train.shape, X_valid.shape, y_train.shape, y_valid.shape)
```

Результат:

```
(99, 4) (99, 4) (99,) (99,)
```

Проведем классификацию наблюдений наивным байесовским методом. Выведем количество верных и не верных наблюдений.

```
df = pd.read_csv('lab7.csv',encoding='cp1251')
y = df['gender'].astype(int)
X = df.drop('gender', axis=1)
```

```
X_train, X_valid, y_train, y_valid = train_test_split(X, y,
train_size=0.5, random_state=10)
gnb = GaussianNB()
y_pred = gnb.fit(X_train, y_train).predict(X_valid)
print(('True:' , y_valid != y_pred).sum())
print(('True:' , y_valid == y_pred).sum())
```

Результат:

```
True: 41
Not true: 58
```

Проведем классификацию наблюдений методом деревьев. Выведем количество верных и не верных наблюдений.

```
df = pd.read_csv('lab7.csv',encoding='cp1251')
y = df['gender'].astype(int)
X = df.drop('gender', axis=1)
X_train, X_valid, y_train, y_valid = train_test_split(X, y,
train_size=0.5, random_state=10)
DT = DecisionTreeClassifier()
y_pred = DT.fit(X_train, y_train).predict(X_valid)
print('True:',(y_valid != y_pred).sum())
print('Not true:',(y_valid == y_pred).sum())
```

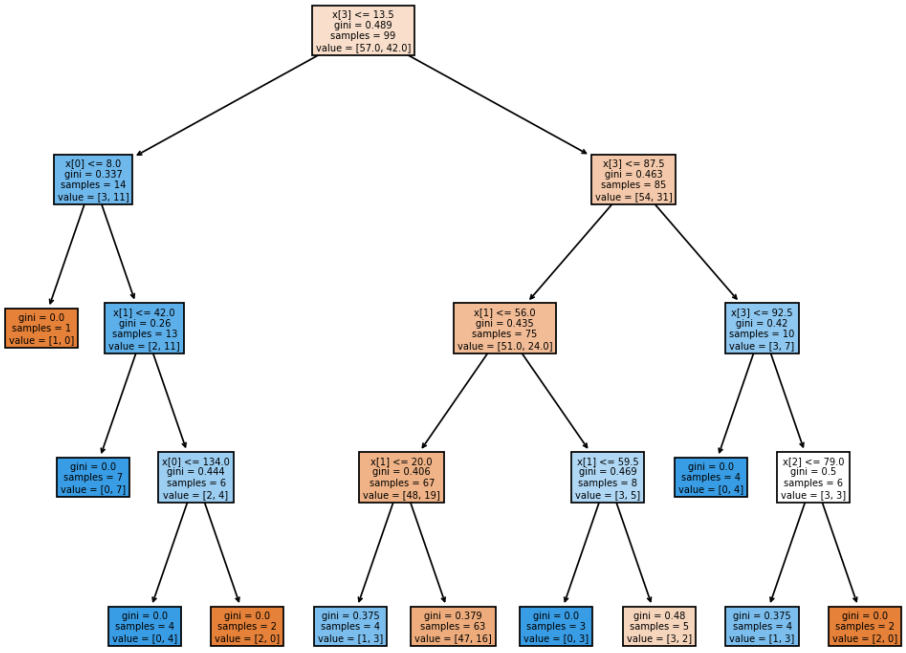
Результат:

```
True: 39
Not true: 60
```

И отобразим дерево с ограничением по максимальной глубине 4.

```
df = pd.read_csv('lab7.csv',encoding='cp1251')
y = df['gender'].astype(int)
X = df.drop('gender', axis=1)
X_train, X_valid, y_train, y_valid = train_test_split(X, y,
train_size=0.5, random_state=10)
DT = DecisionTreeClassifier(max_depth=4, random_state=10)
print(DT.fit(X_train, y_train))
plt.subplots(1,1,figsize = (10,10))
tree.plot_tree(DT, filled = True)
plt.show()
```

Результат:



В ходе лабораторной работы мы познакомились с методами классификации Sklearn. По данным результатов неправильно классифицированных результатов в зависимости от изменении пропорций выборки построили график зависимости, который показал… и отобразили дерево глубиной 3.

Задание.

1. Загрузить пред обработанные данные, вывести первые 5 строк из датасета;
2. Выделить данные и метки признаков, разбейте выборку на обучающую и тестовую в пропорции 75/25;
3. Провести классификацию наблюдений наивным байесовским методом и методом деревьев;
4. Указать точность наблюдений score() и измените пропорции выборки (85/15,75/25, 65/35, 55/45, 45/55, 35/65, 25/75, 15/85);
5. Построить график зависимости неправильно классифицированных результатов в зависимости от пропорции выборки.
6. Отобразите дерево максимальной глубиной 3.

Формат отчета.

Протокол лабораторной работы в формате PDF, который должен содержать, поэтапное выполнение всех задач с текстовым описанием ваших действий и экранными формами, отображающими данные действия.