Лабораторная работа №5

Задание

1. Ознакомиться с классификаторами библиотеки Scikit-learn
2. Выбрать для исследования не менее 3 классификаторов
3. Выбрать набор данных для задач классификации из открытых источников
4. Выбор классификаторов и набора данных утвердить у преподавателя (не должно быть полного совпадения с выбором другого студента)
5. Для каждого классификатора определить целевой столбец и набор признаков. Обосновать свой выбор. При необходимости преобразовать типы признаковых данных.
6. Подготовить данные к обучению.
7. Провести обучение и оценку моделей на сырых данных.
8. Провести предобработку данных.
9. Провести обучение и оценку моделей на очищенных данных.
10. Проанализировать результаты.
11. Результаты анализа представить в табличной и графической форме.
12. Сформулировать выводы.
13. Оформить отчет по л/р.

Порядок выполнения работы

1. Для лабораторной работы был выбран «Набор данных для анализа и прогнозирования сердечного приступа» dataset - heart.csv. Набор данных следующий:
    - **age** - возраст человека;
    - **sex** – пол человека;
        - значение: 0 – мужчина;
        - значение: 1 – женщина;
    - **cp** - тип боли в груди;

    - значение: 0 - бессимптомный;
    - значение: 1 - типичная стенокардия;
    - значение: 2 - атипичная стенокардия;
    - значение: 3 - неангинальная боль;
- **trtbps** - кровяное давление в состоянии покоя (в мм рт. ст.);
- **chol** - уровень холестерина в мг/дл определяется с помощью датчика ИМТ);
- **fbs** - уровень сахара в крови натощак > 120 мг/дл;
    - значение: 1 – да;
    - значение: 0 – нет;
- **restecg** - результаты электрокардиографии в состоянии покоя;
    - значение: 0 - обычный;
    - значение: 1 - наличие аномалии зубца ST-T (инверсии зубца T и/или подъем или депрессия ST > 0,05 мВ);
    - значение: 2 - наличие вероятной или определенной гипертрофии левого желудочка по критериям Эстеса;
- **thalachh** - достигнутая максимальная частота сердечных сокращений;
- **exng** - стенокардия, вызванная физической нагрузкой;
    - значение: 1 – да;
    - значение: 0 – нет;
- **oldpeak** - Предыдущий пик;
- **slp** – наклон сегмента ST пикового упражнения;
    - значение: 0 – спуск;
    - значение: 1 – пологий;
    - значение: 2 – подъем;
- **caa** - количество крупных капилляров (0-3);
- **thall** - thal rate;
    - значение: 1 – исправленный дефект;
    - значение: 2 – нормальный;

    - значение: 3 – обратимый дефект;
- **output** – шанс на сердечный приступ;
    - значение: 0 - меньше шансов на сердечный приступ;
    - значение: 1 - больше шансов на сердечный приступ;

Классификаторы были выбраны следующие (данный выбор утвержден преподавателем):

- Naive Bayes Classifier;
- Support Vector Machines;
- Stochastic Gradient Descent Classifier;
- Random Forest Classifier;

Данный выбор утвержден у преподавателя.

Наивный байесовский классификатор (Naive Bayes Classifier) – это самый простой алгоритм, который вы можете применить к своим данным. Как следует из названия, этот алгоритм делает предположение, что все переменные в наборе данных "наивные", т.е. не коррелируют друг с другом.

Плюсы:

- Алгоритм легко и быстро предсказывает класс тестового набора данных. Он также хорошо справляется с многоклассовым прогнозированием.
- Производительность наивного байесовского классификатора лучше, чем у других простых алгоритмов, таких как логистическая регрессия. Более того, вам требуется меньше обучающих данных.
- Он хорошо работает с категориальными признаками (по сравнению с числовыми). Для числовых признаков

предполагается нормальное распределение, что может быть серьезным допущением в точности нашего алгоритма.

Минусы:

- Значения спрогнозированных вероятностей, возвращенные методом *predict_proba*, не всегда являются достаточно точными.
- Ограничением данного алгоритма является предположение о независимости признаков. Однако в реальных задачах полностью независимые признаки встречаются крайне редко.

Метод опорных векторов (Support Vector Machines) — это набор контролируемых методов обучения, используемых
для классификации , регрессии и обнаружения выбросов .

Плюсы:

- Эффективен в пространствах больших размеров.
- По-прежнему эффективен в случаях, когда количество измерений превышает количество образцов.
- Использует подмножество обучающих точек в функции принятия решений (называемых опорными векторами), поэтому это также эффективно с точки зрения памяти.
- Универсальность: для функции принятия решения могут быть указаны различные функции ядра . Предоставляются общие ядра, но также можно указать собственные ядра.

Минусы:

- Если количество функций намного превышает количество выборок, избегайте чрезмерной подгонки при выборе функций ядра, и термин регуляризации имеет решающее значение.

- SVM не предоставляют напрямую оценки вероятностей, они рассчитываются с использованием дорогостоящей пятикратной перекрестной проверки (см. Оценки и вероятности ниже).

Стохастический градиентный спуск (Stochastic Gradient Descent Classifier**)** — это простой, но очень эффективный подход к подгонке линейных классификаторов и регрессоров под выпуклые функции потерь, такие как (линейные) Метод опорных векторов и логистическая регрессия . Несмотря на то, что SGD существует в сообществе машинного обучения уже давно, совсем недавно он привлек значительное внимание в контексте крупномасштабного обучения.

Плюсы:

- Эффективность.
- Простота реализации (множество возможностей для настройки кода).

Минусы:

- SGD требует ряда гиперпараметров, таких как параметр регуляризации и количество итераций.
- SGD чувствителен к масштабированию функций.

Случайный лес (Random Forest Classifier) — это алгоритм обучения с учителем. Его можно применять как для классификации, так и для регрессии. Также это наиболее гибкий и простой в использовании алгоритм. Лес состоит из деревьев. Говорят, что чем больше деревьев в лесу, тем он крепче. RF создает деревья решений для случайно выбранных семплов данных, получает прогноз от каждого дерева и выбирает наилучшее решение посредством голосования. Он также предоставляет довольно эффективный критерий важности показателей (признаков).

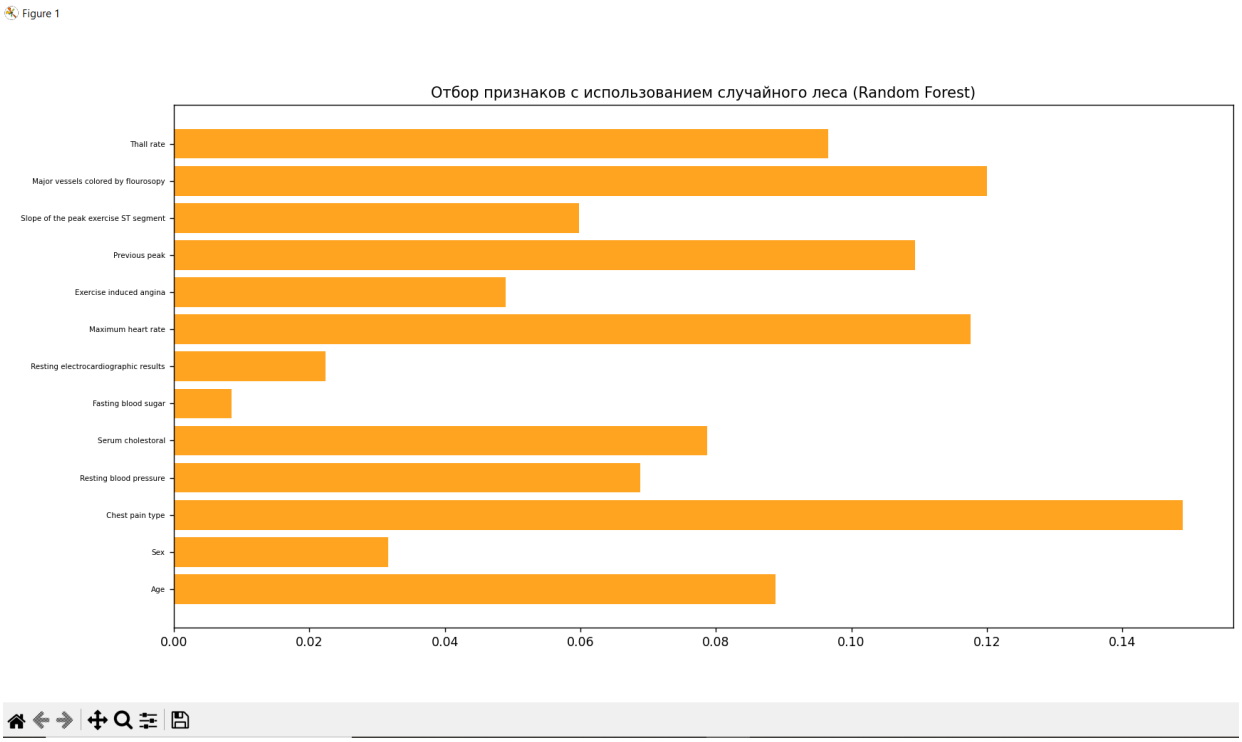
Случайный лес имеет множество применений, таких как механизмы рекомендаций, классификация изображений и отбор признаков. Он лежит в основе алгоритма Борута, который определяет наиболее значимые показатели датасета.

2. Для лучшего понимания, имена признаков были изменены на те, что лучше характеризуют признак:

```
train.rename(columns=
    {
        'age': 'Age',
        'sex': 'Sex',
        'cp': 'Chest pain type',
        'trtbps': 'Resting blood pressure',
        'chol': 'Serum cholestoral',
        'fbs': 'Fasting blood sugar',
        'restecg': 'Resting electrocardiographic results',
        'thalachh': 'Maximum heart rate',
        'exng': 'Exercise induced angina',
        'oldpeak': 'Previous peak',
        'slp': 'Slope of the peak exercise ST segment',
        'caa': 'Major vessels colored by flourosopy',
        'thall': 'Thall rate',
        'output': 'Risk of heart attack'
    }, inplace=True)
```

Для классификаторов определен целевой столбец – «Risk of heart attack» (Шанс на сердечный приступ).

Для определения набора признаков, была написана функция, где при помощи «Случайного леса», можно рассчитать, насколько важен признак, рассчитав степень уменьшения «шумов» за счёт этого признака. Важность признаков была визуализирована на диаграмме ниже:



Таким образом, больше наибольшее влияние на целевой столбец оказывают признаки: «Chest pain type (тип боли в груди)» «Maximum heart rate (достигнутая максимальная частота сердечных сокращений)» «Previous peak (Предыдущий пик)» «Major vessels colored by flourosopy (количество крупных капилляров)» «Thall rare». Поэтому данные признаки были оставлены в dataset, а остальные удалены.

3. Обучение моделей на сырых данных произошло без ошибок, так как пропусков значений в столбцах не было, что было проверено выводя информацию о datafream. Единственное, выводя информацию о дубликатах было найдено 13 повторяющих записей, которые были в последующем удалены.
4. Затем производилась оценка моделей, отчет и визуализация представлены ниже на скриншотах консоли и диаграммах:

Naive Bayes Classifier

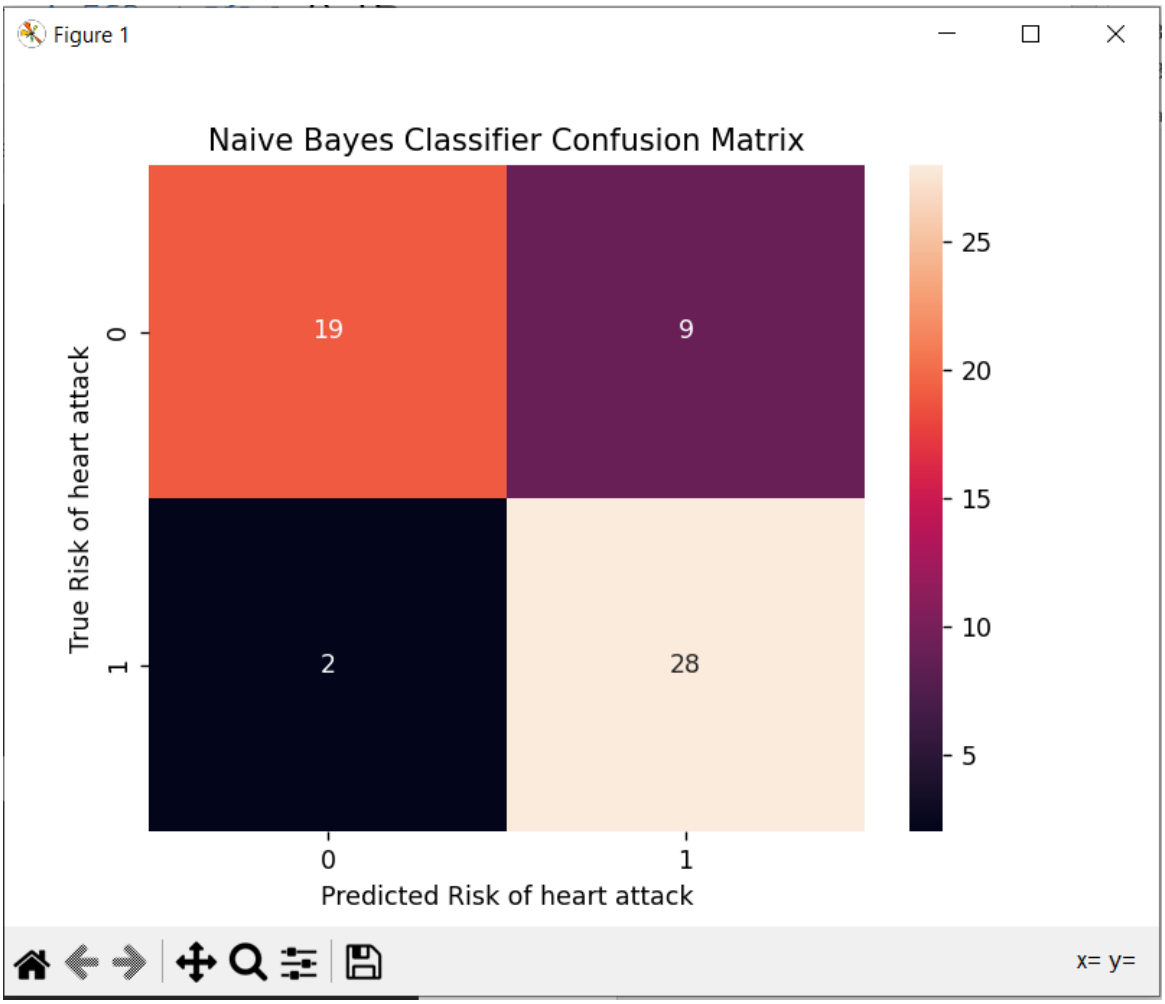
```
Отчет о наивной Байесовской классификации:

              precision    recall  f1-score   support

           0       0.90      0.68      0.78        28
           1       0.76      0.93      0.84        30

    accuracy                           0.81        58
   macro avg       0.83      0.81      0.81        58
weighted avg       0.83      0.81      0.81        58


Точность Наивного Байесовского классификатора: 0.8103448275862069
```



Support Vector Machines

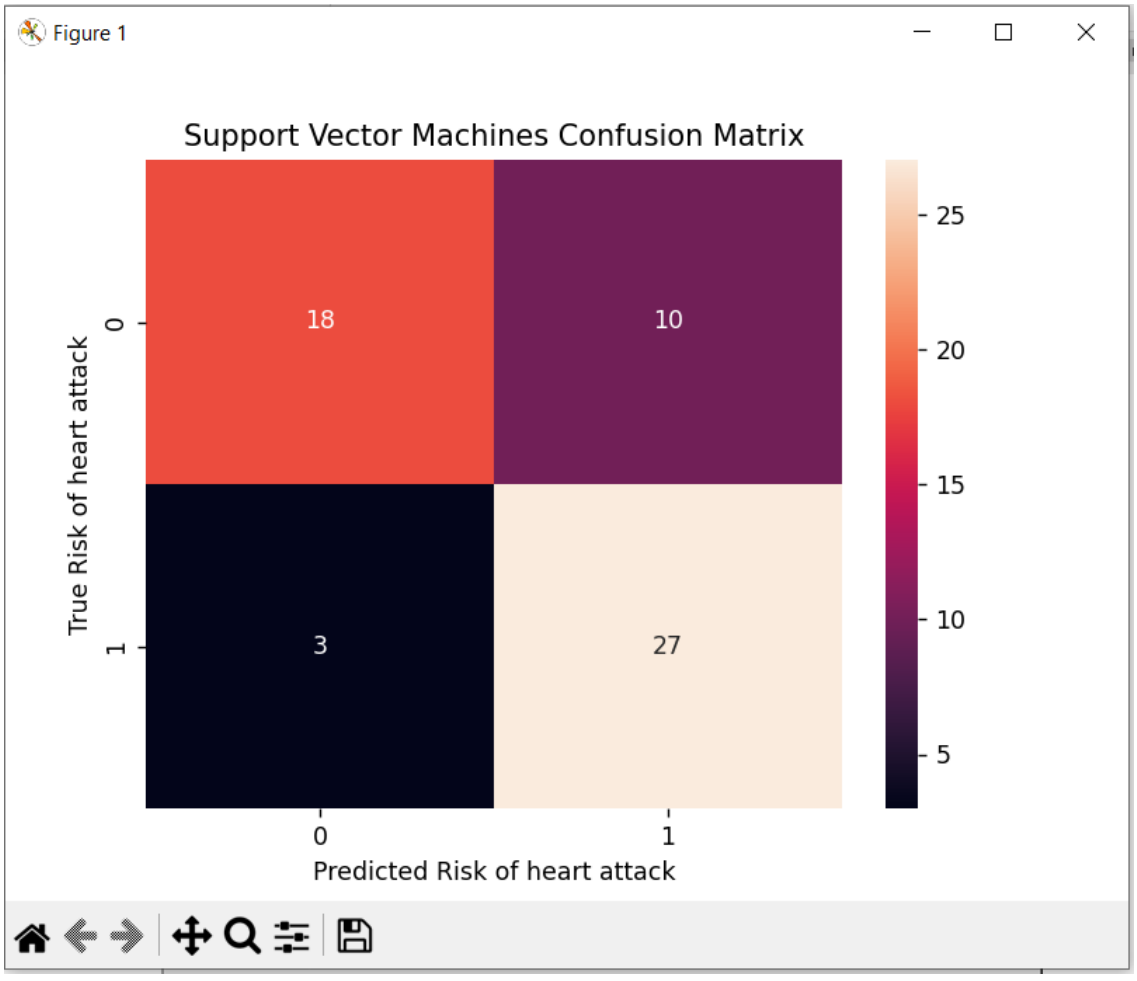
```
Отчет о классификации опорных векторов:

              precision    recall  f1-score   support

           0       0.86      0.64      0.73        28
           1       0.73      0.90      0.81        30

    accuracy                           0.78        58
   macro avg       0.79      0.77      0.77        58
weighted avg       0.79      0.78      0.77        58


Точность классификатора Опорные вектора: 0.7758620689655172
```



Stochastic Gradient Descent Classifier

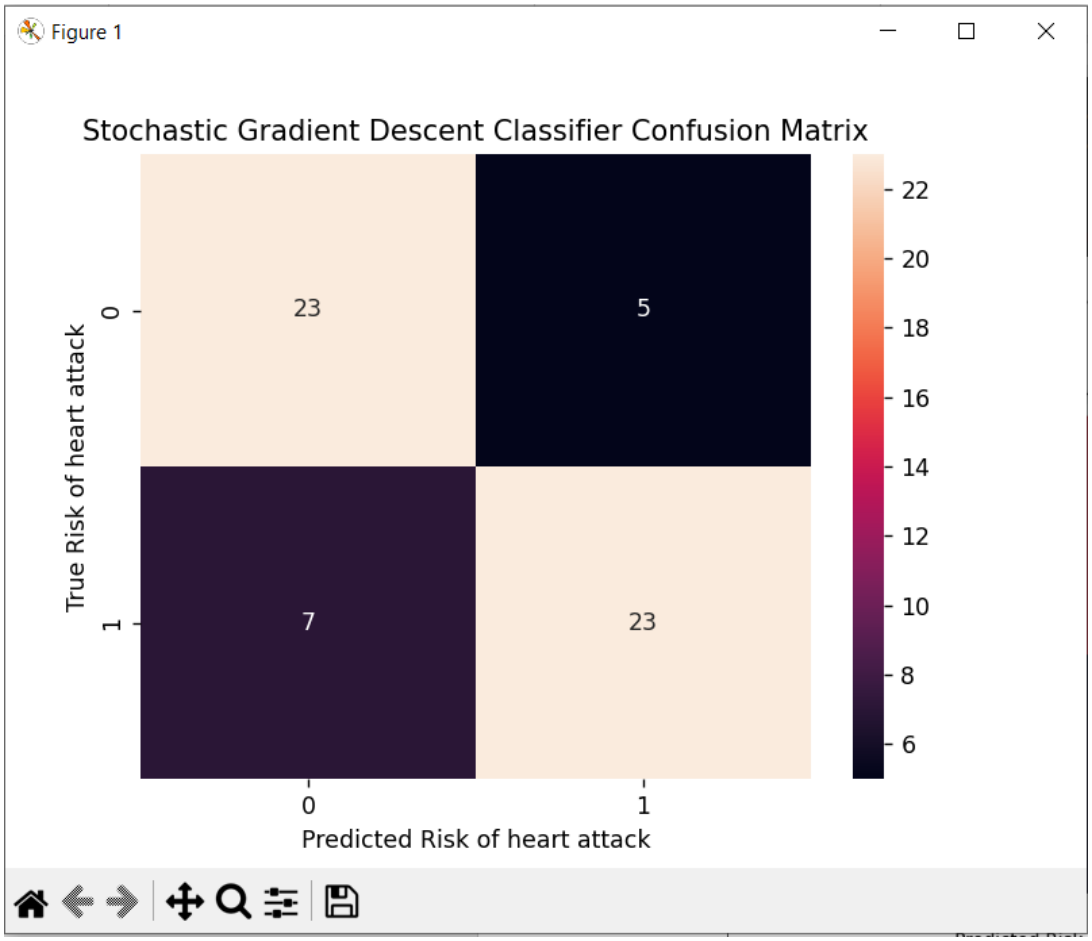
```
Отчет о классификации стохастического градиентного спуска:

              precision    recall  f1-score   support

           0       0.77      0.82      0.79        28
           1       0.82      0.77      0.79        30

    accuracy                           0.79        58
   macro avg       0.79      0.79      0.79        58
weighted avg       0.79      0.79      0.79        58


Точность классификатора Стохастический Градиентный Спуск: 0.7931034482758621
```



Random Forest Classifier

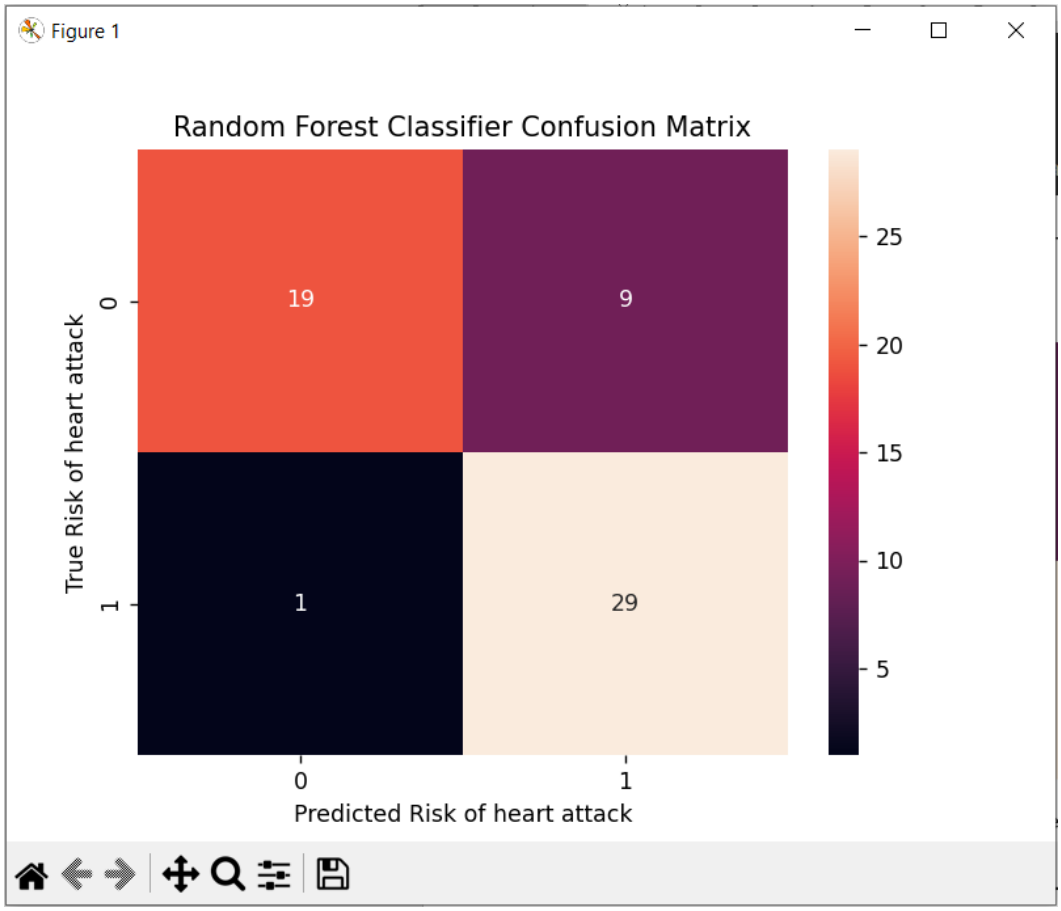
```
Отчет о классификации случайного леса:

              precision    recall  f1-score   support

           0       0.95      0.68      0.79        28
           1       0.76      0.97      0.85        30

    accuracy                           0.83        58
   macro avg       0.86      0.82      0.82        58
weighted avg       0.85      0.83      0.82        58


Точность классификатора Случайный Лес: 0.8275862068965517
```



Вывод: проведя оценку работ классификаторов было выявлено, что на данном наборе данных с классификацией лучше всего справились Random Forest Classifier и Naive Bayes Classifier.