**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №8**

**по дисциплине «Машинное обучение»**

**Тема: Классификация (Линейный дискриминантный анализ, метод опорных векторов)**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8303 |  | Гришин К. И. |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2021

# Цель работы

Ознакомиться с методами кластеризации из библиотеки *Sklearn*.

# Ход выполнения работы

## Загрузка данных

1. Скачать датасет: <https://archive.ics.uci.edu/ml/datasets/iris>
2. Загрузить датасет в датафрейм

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** |
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |

1. Выделены данные и метки
2. Метки преобразованы к числам
3. Данные разбиты на обучающую и тестовую выборки

## Линейный дискриминантный анализ

1. Проведена классификация *LDA (Linear Discriminant Analysis)*

Тестовая и обучающая выборки представляют собой исходные данные, поделенные пополам.

Неправильно классифицировано 3 значения.

|  |  |
| --- | --- |
| **Параметр** | **Описание** |
| *solver* | Метод поиска компонент  *`svd`* - поиск сингулярных значений без знаний о матрице ковариации  *`lsqr`* - метод наименьших квадратов  *`eigen`* - метод собственных чисел |
| *shrinkage* | Сжатие матрицы ковариации. При *`auto`* - используется лемма Ледуа-Вольфа |
| *priors* | Априорные вероятности. Изначальное считается по результатам обучения |
| *n\_components* | Количество компонентов разбиения. Параметр влияет только на transform |

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| *coef\_* | Весовые вектора дискриминанта Фишера. |
| *covariance\_* | Взвешенная матрица ковариации |
| *explained\_variance\_ratio\_* | Объясненная дисперсия каждой компоненты |
| *means\_* | Мат. ожидания каждого класса |
| *priors\_* | Вероятности классов |
| *classes\_* | Метки классов |

1. Точность классификации
2. График зависимости количества неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки. *random\_state = 830303*. (рис. 1)

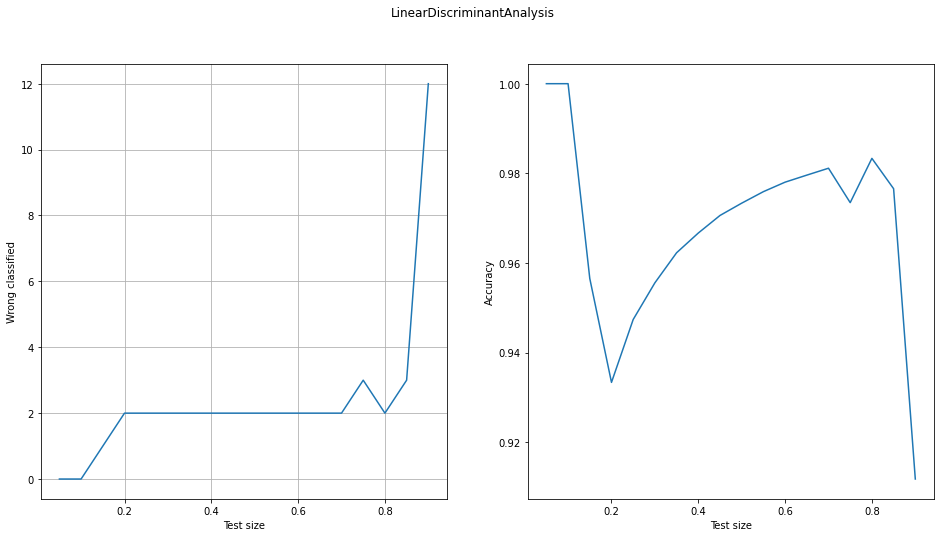


Рисунок 1. Зависимость количества неправильно классифицированных данных и точности классификации от размера тестовой выборки *LDA default*.

1. Описание метода *`transform`*

Проведено понижение размерности до 2 при помощи методов *PCA* и *LDA*. (рис. 2).

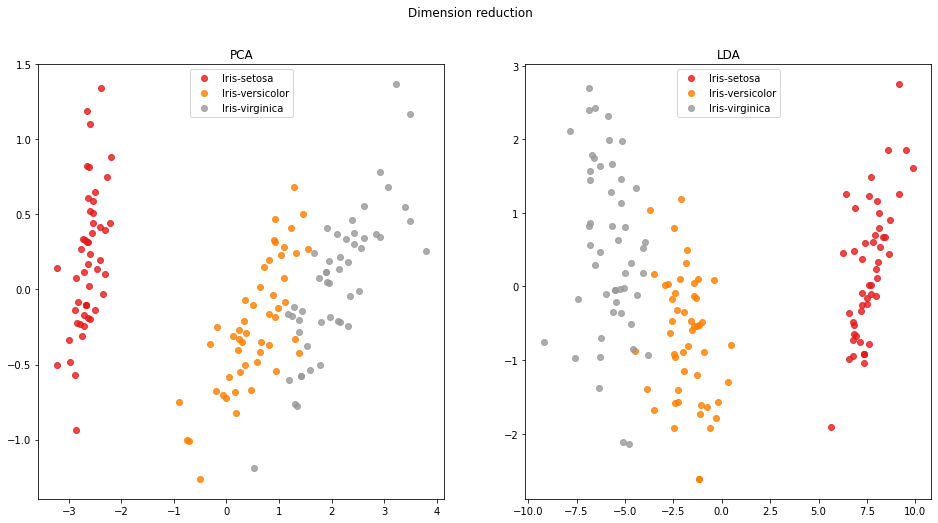


Рисунок 2. Результат понижения размерности с помощью PCA и LDA.

Результат получился похож. Связано это с тем, компонента наибольшего разброса дисперсии совпадает с компонентой разделения классов.

1. Классификация при различных параметрах *solver* и *shrinkage*

solver = SVD (рис. 3, 4)

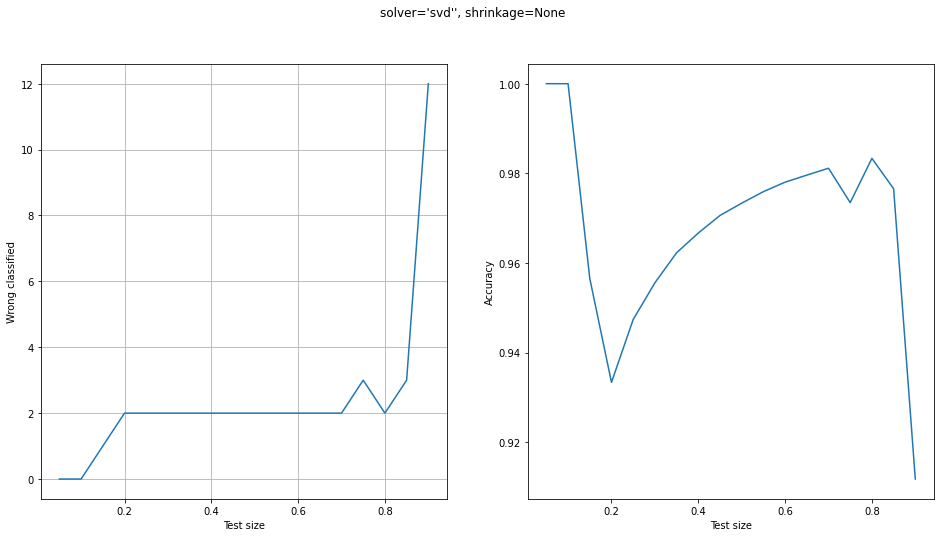


Рисунок 3. *LDA(solver="svd", shrinkage=None)*

solver = LSQR (рис. 4, 5, 6, 7, 8, 9, 10)

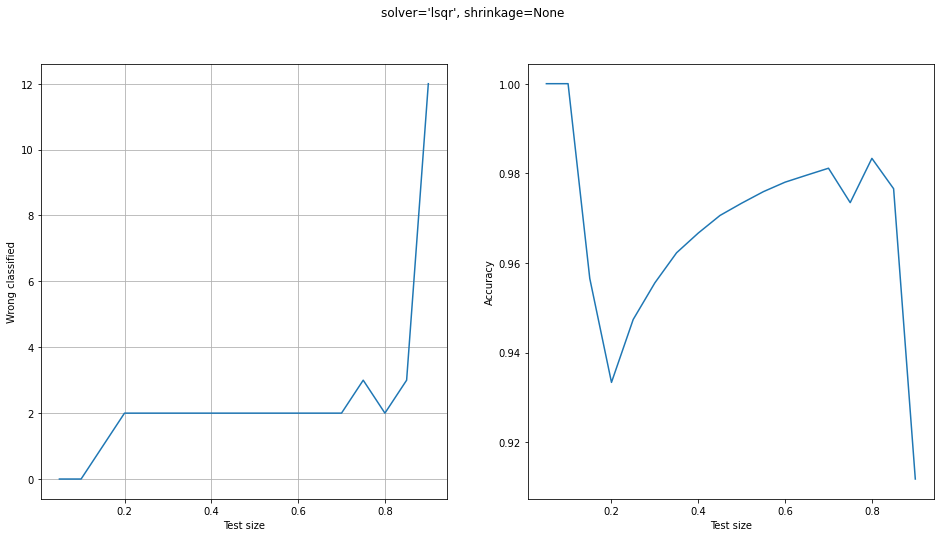


Рисунок 4. *LDA(solver="lsqr", shrinkage=None)*

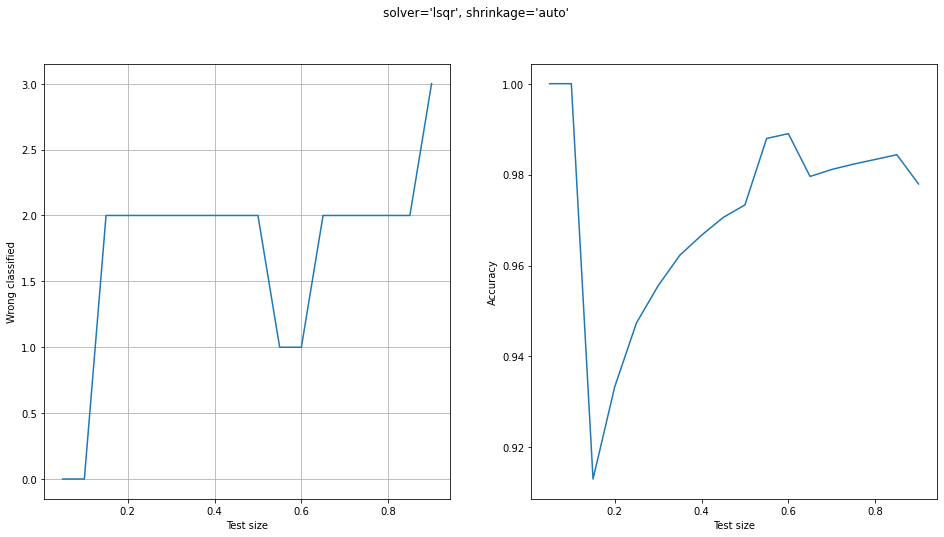


Рисунок 5. *LDA(solver="lsqr", shrinkage="auto")*

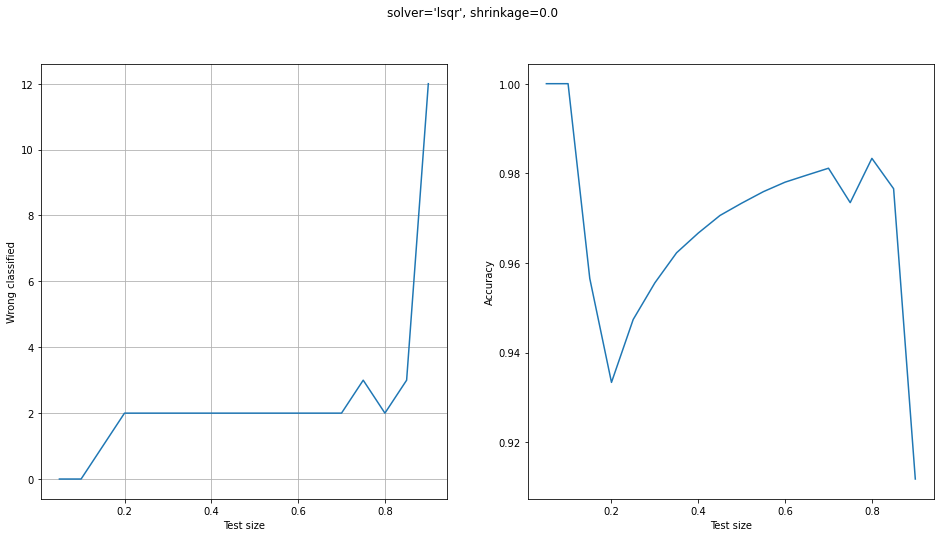


Рисунок 6. *LDA(solver="lsqr", shrinkage=0)*

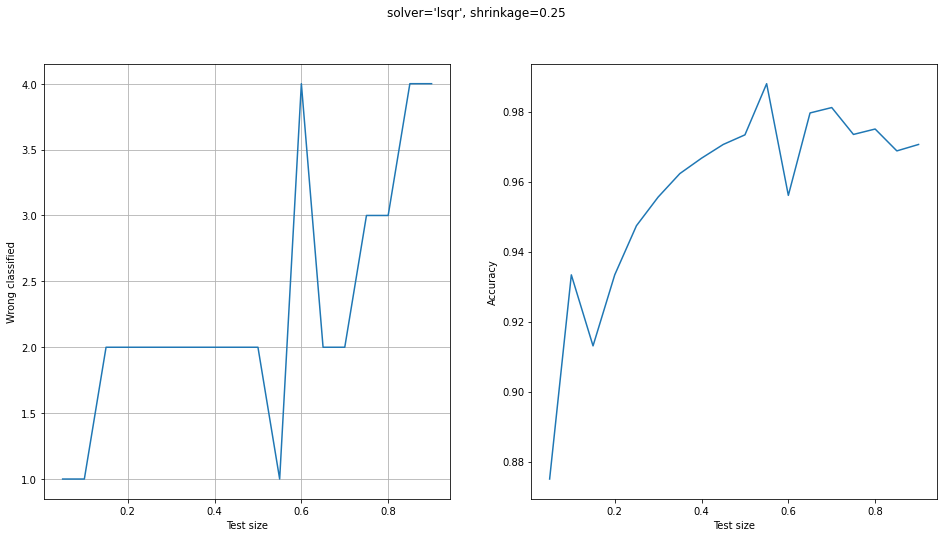


Рисунок 7. *LDA(solver="lsqr", shrinkage=0.25)*

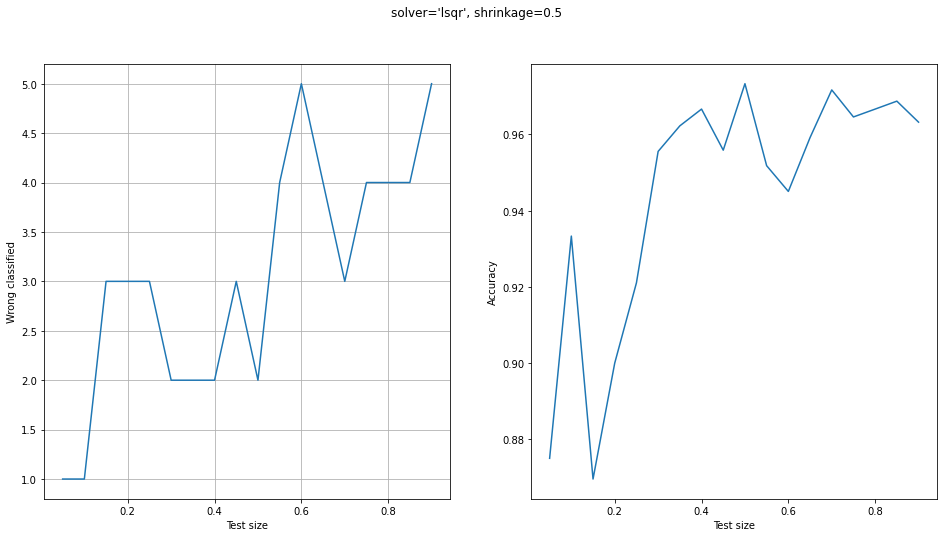


Рисунок 8. *LDA(solver="lsqr", shrinkage=0.5)*

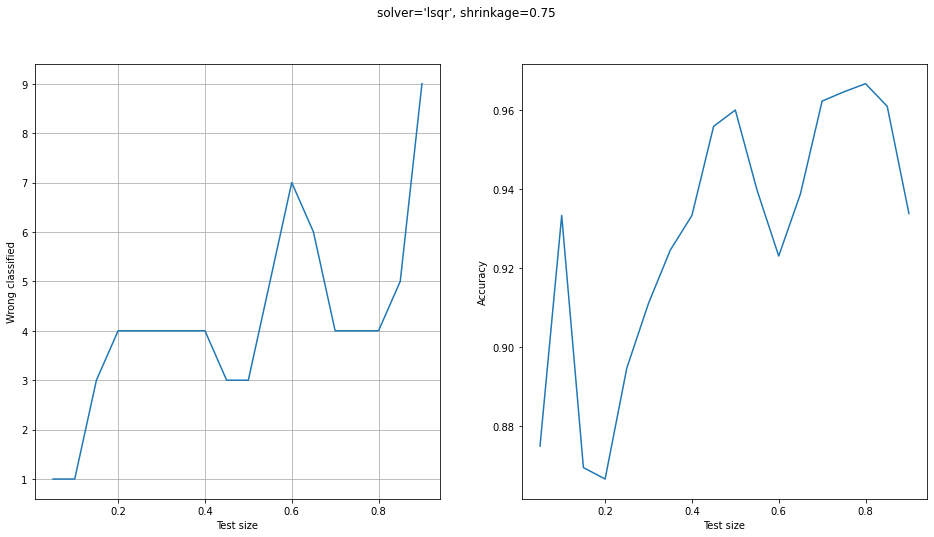


Рисунок 9. *LDA(solver="lsqr", shrinkage=0.75)*

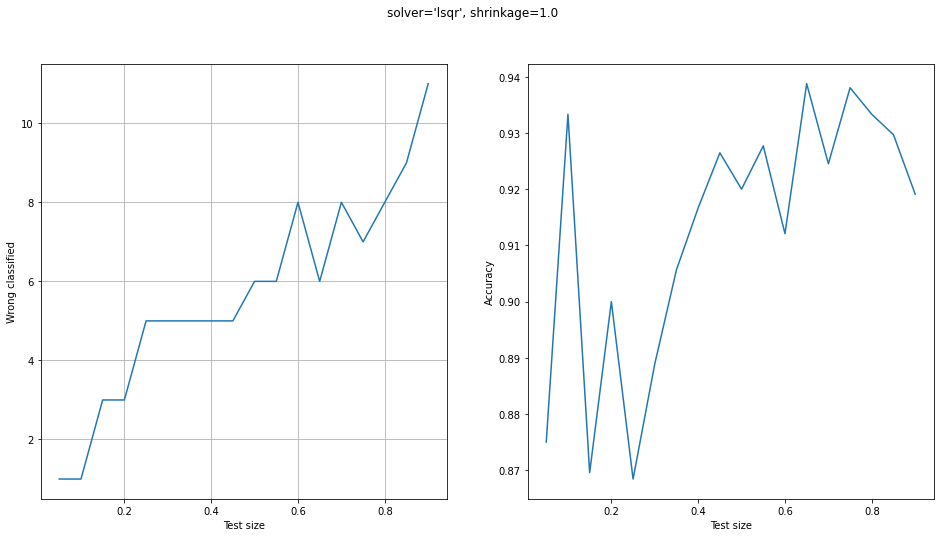


Рисунок 10. *LDA(solver="lsqr", shrinkage=1)*

solver = eigen (рис. 11, 12, 13, 14, 15, 16, 17)

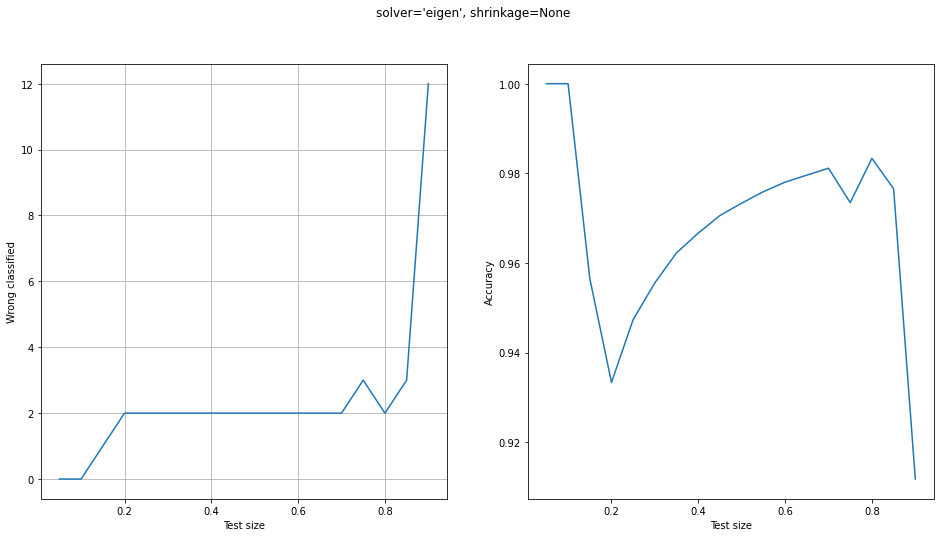


Рисунок 11. *LDA(solver="eigen", shrinkage=None)*

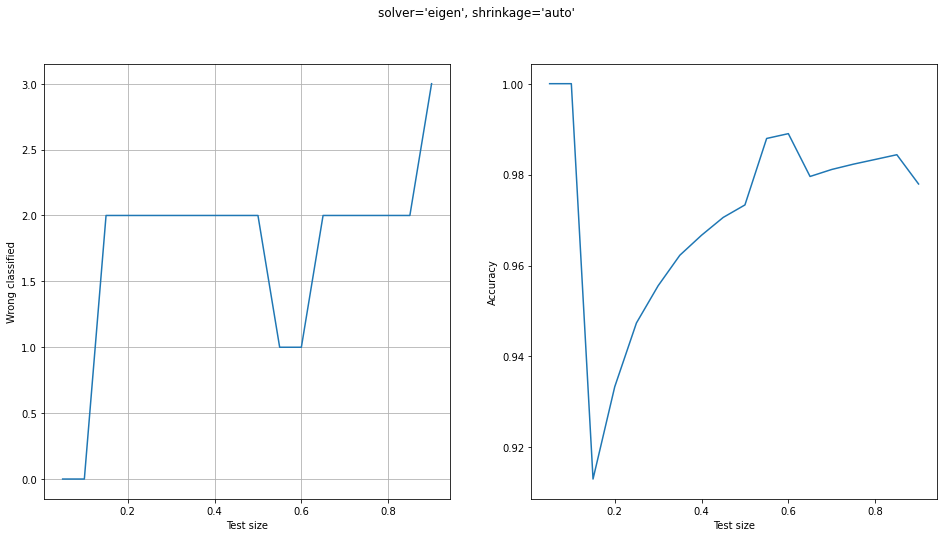


Рисунок 12. *LDA(solver="eigen", shrinkage="auto")*

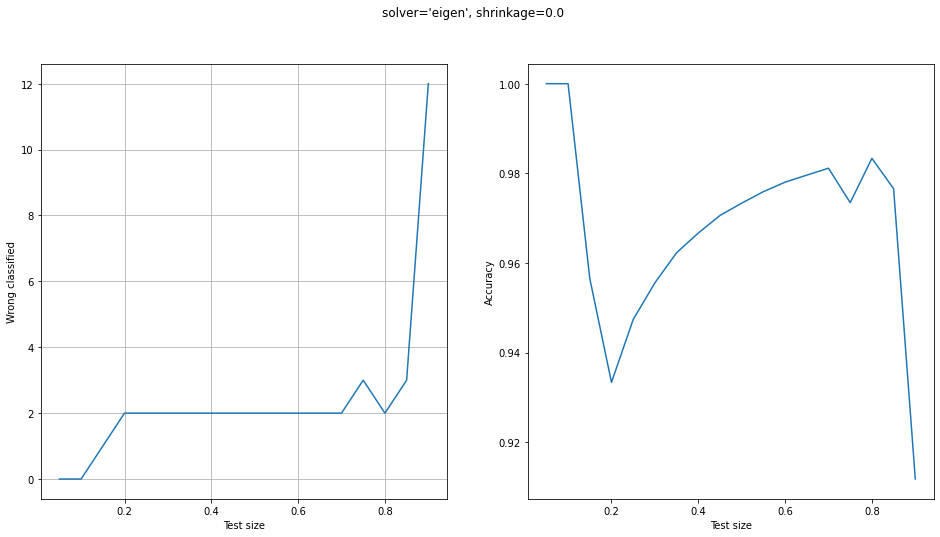


Рисунок 13. *LDA(solver="eigen", shrinkage=0)*

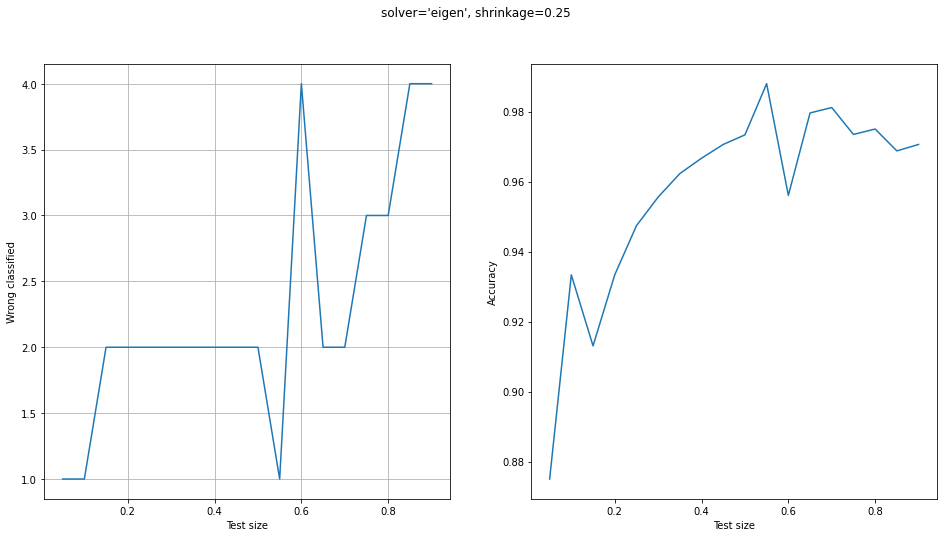


Рисунок 14. *LDA(solver="eigen", shrinkage=0.25)*

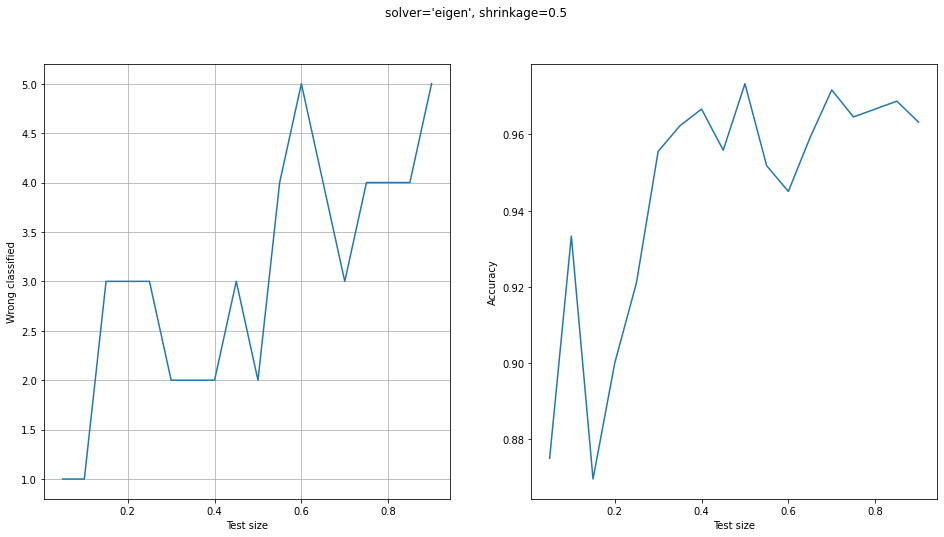


Рисунок 15. *LDA(solver="eigen", shrinkage=0.5)*

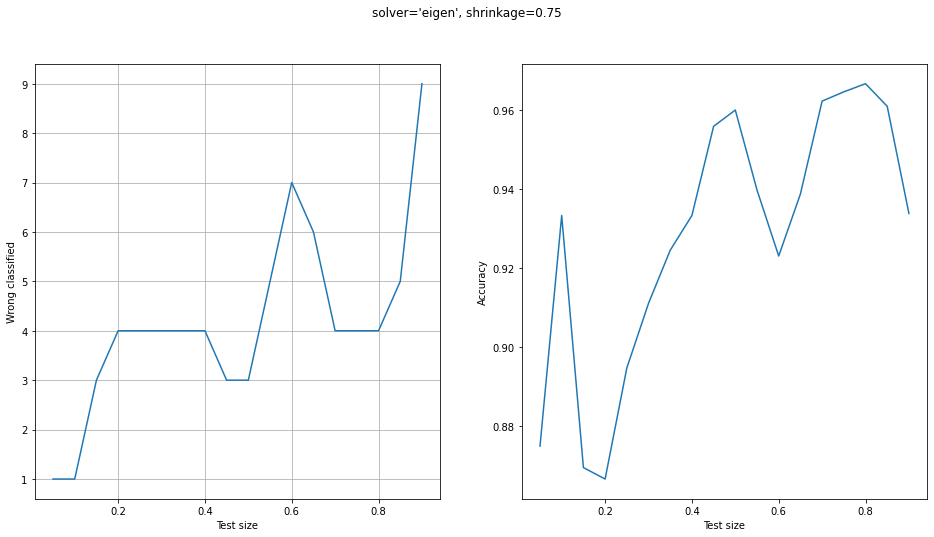


Рисунок 16. *LDA(solver="eigen", shrinkage=0.75)*

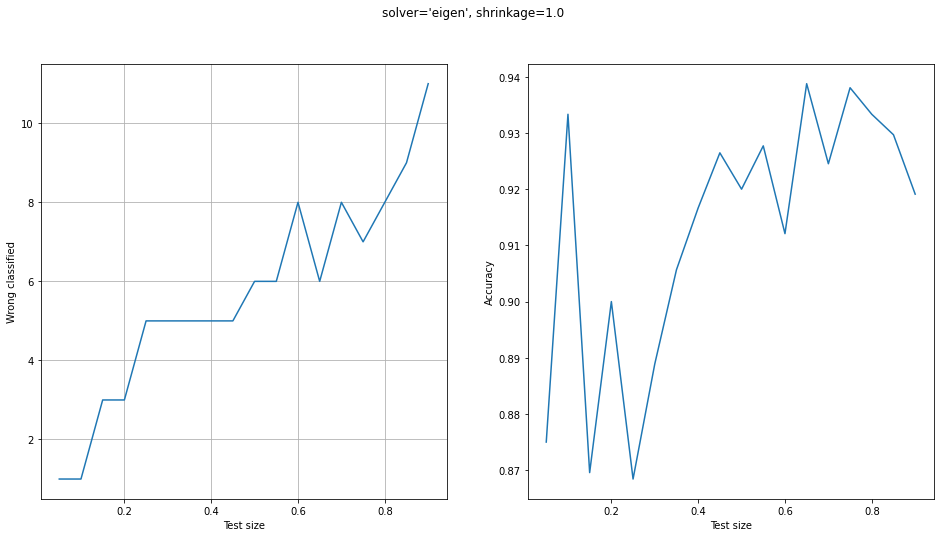


Рисунок 17. *LDA(solver="igen", shrinkage=1)*

1. Установка априорной вероятности 0.7 классу 1.

Неправильно классифицировано 4

График зависимости количества неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки представлен на рисунке 18.

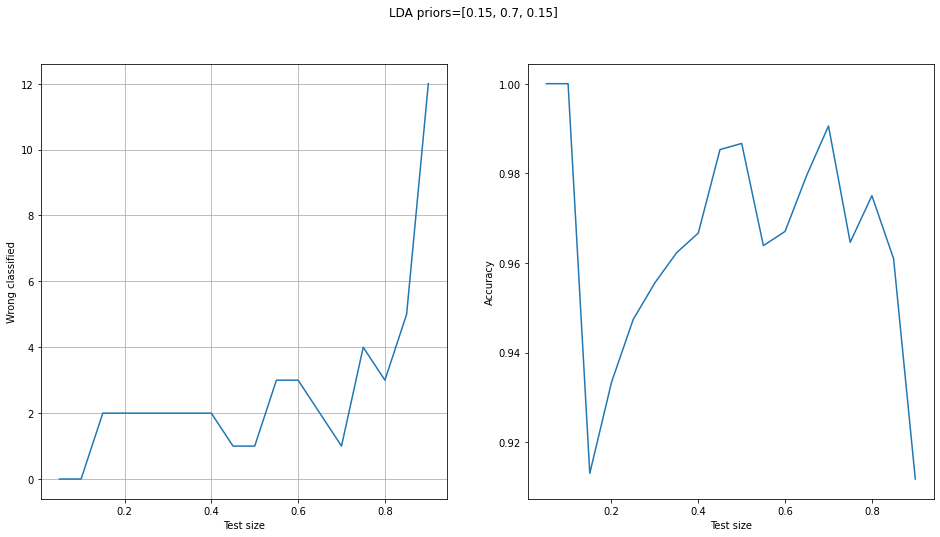


Рисунок 18. Зависимость количества неправильно классифицированных данных и точности классификации от размера тестовой выборки LDA prior [0.15, 0.75, 0.15].

При использовании априорных вероятностей, результат изменился в лучшую сторону, хоть и не сильно. Однако стал менее равномерным.

## Метод опорных векторов

1. Проведена классификация *SVM (Support Vector Machine)*

Тестовая и обучающая выборки представляют собой исходные данные, поделенные пополам.

Неправильно классифицировано 3 значения.

1. Точность классификации
2. Значение атрибутов классификации

|  |  |
| --- | --- |
| **Аттрибут** | **Значение** |
| *support\_* | Индексы опорных векторов |
| *support\_vectors\_* | Опорные вектора |
| *n\_support\_* | Количество опорных векторов для каждого класса |

1. График зависимости количества неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки. *random\_state = 830303*.

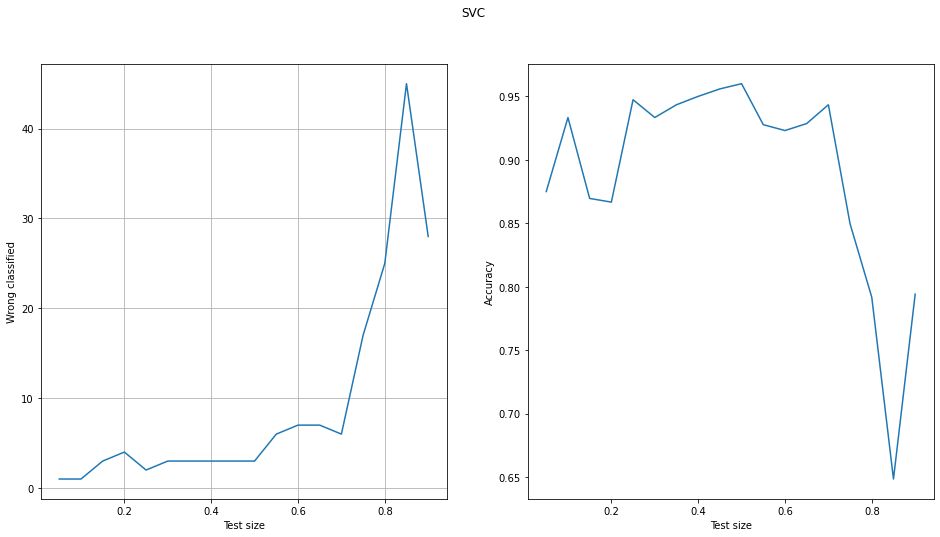


Рисунок 19. Зависимость количества неправильно классифицированных данных и точности классификации от размера тестовой выборки *SVC default*.

1. Классификация при различных параметрах *kernel*, *degree* и *max\_shrinkage*

*kernel* (рис. 20, 21, 22, 23)

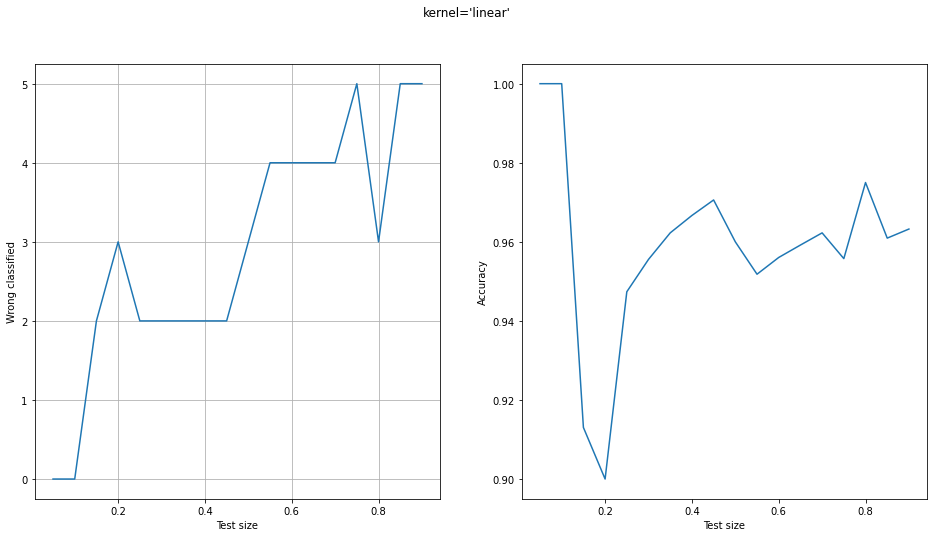


Рисунок 20. *SVC(kernel="linear")*

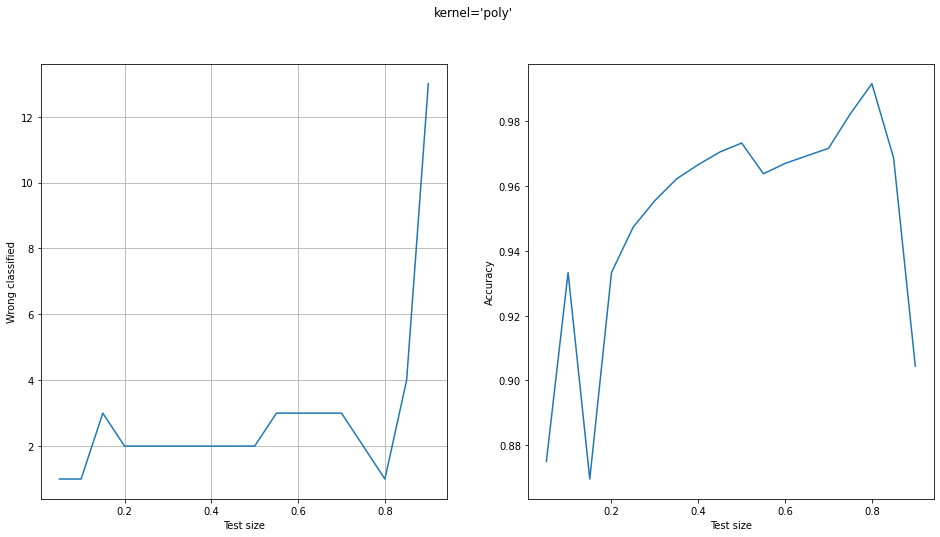


Рисунок 21. *SVC(kernel="poly")*

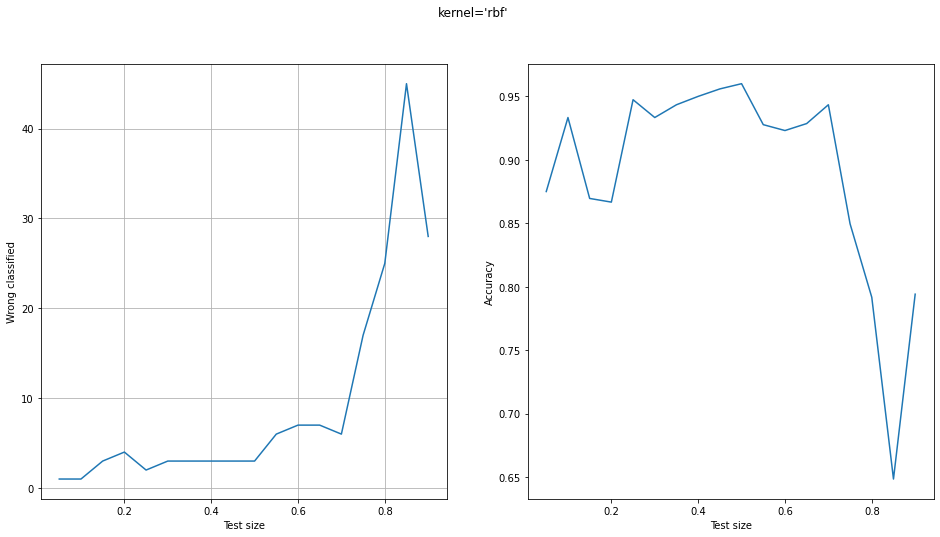


Рисунок 22. *SVC(kernel="rbf")*

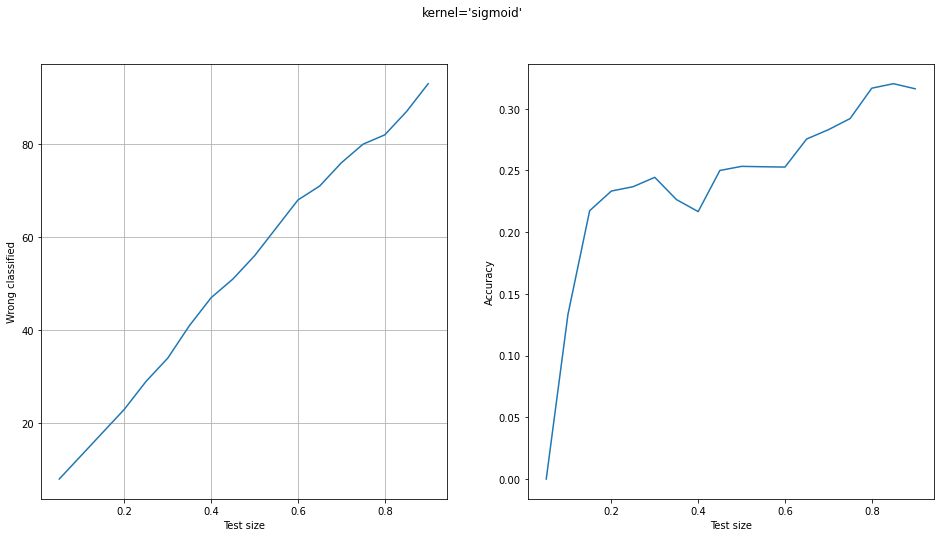


Рисунок 23. *SVC(kernel="sigmoid")*

*degree* (рис. 24, 25, 26, 27, 28)

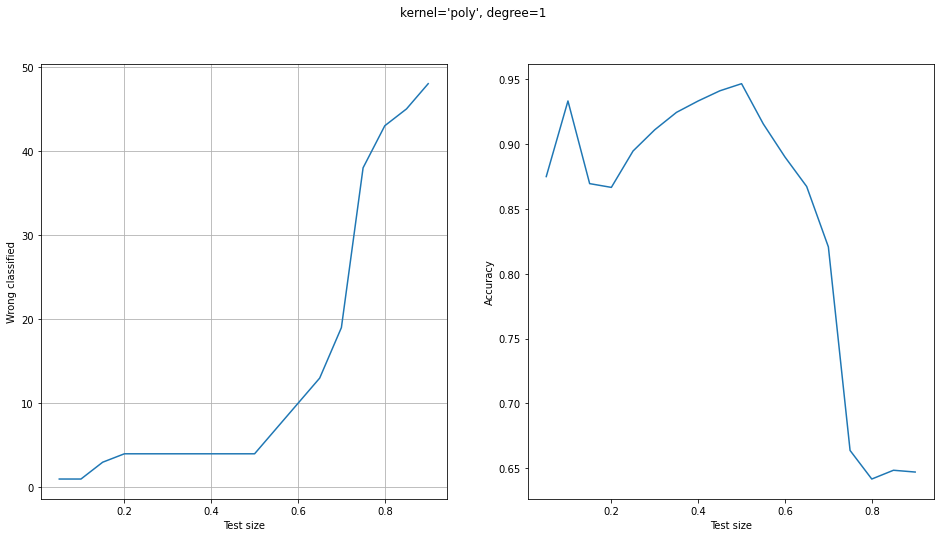


Рисунок 24. *SVC(kernel="poly", degree=1)*

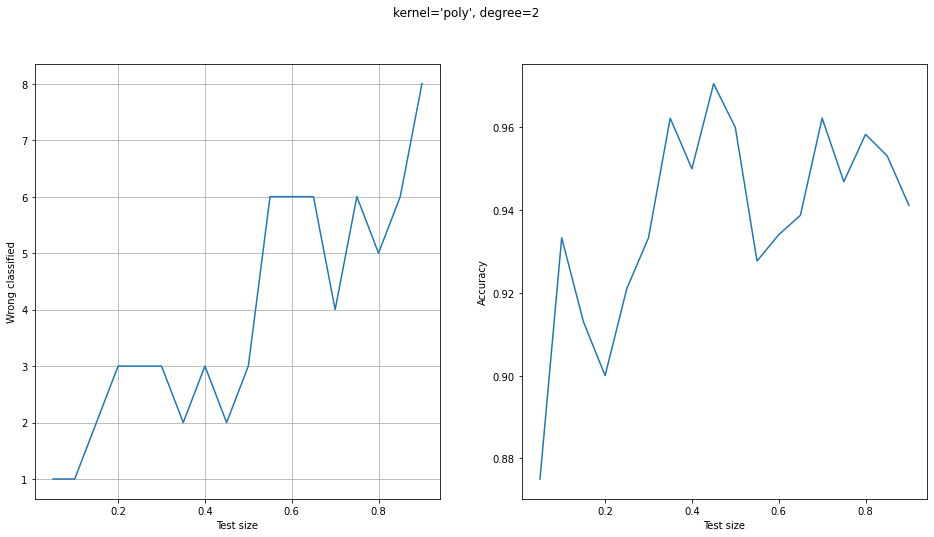


Рисунок 25. *SVC(kernel="poly", degree=2)*

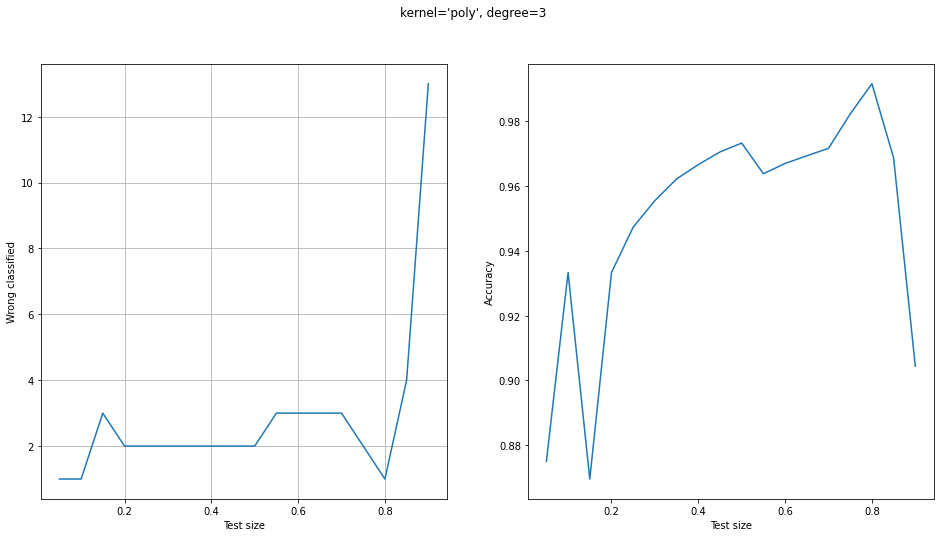


Рисунок 26. *SVC(kernel="poly", degree=3)*

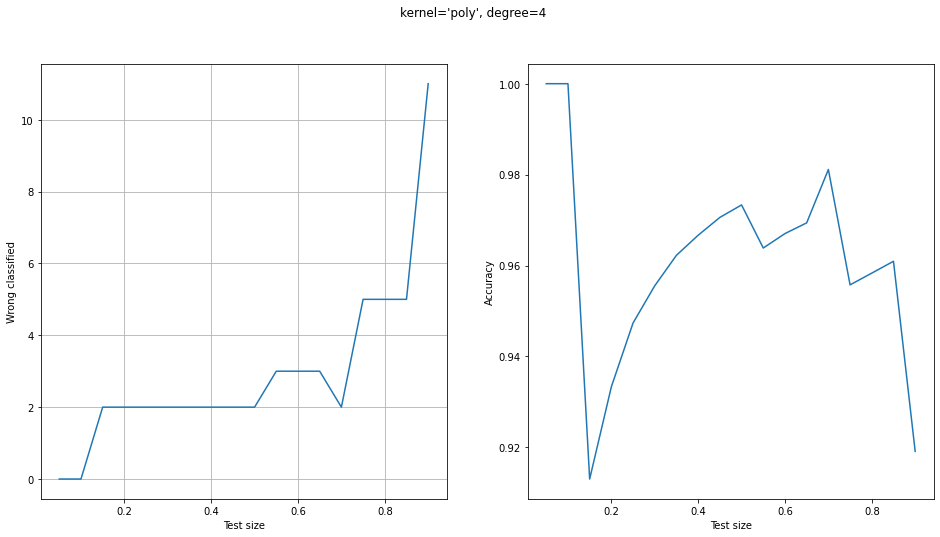


Рисунок 27. *SVC(kernel="poly", degree=4)*

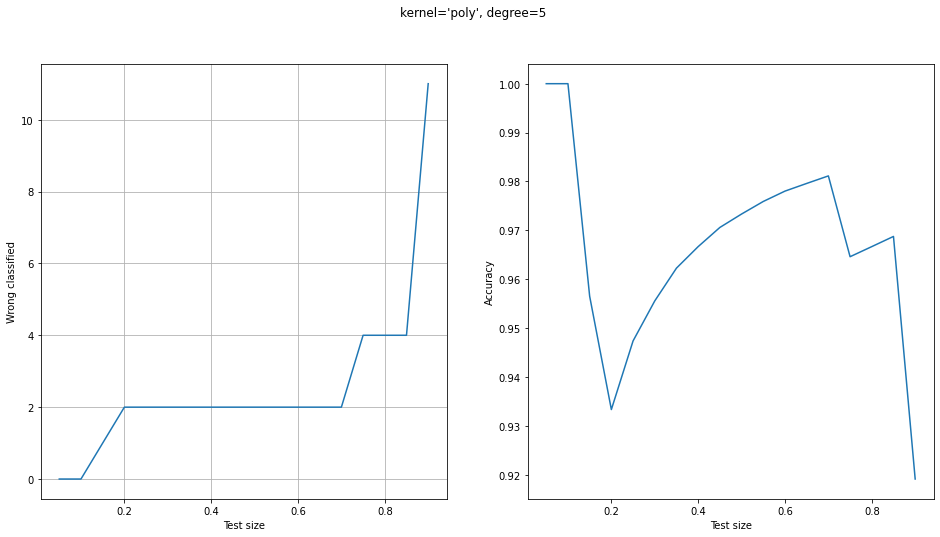


Рисунок 28. *SVC(kernel="poly", degree=1)*

*max\_iter* (рис. 29, 30, 31, 32).

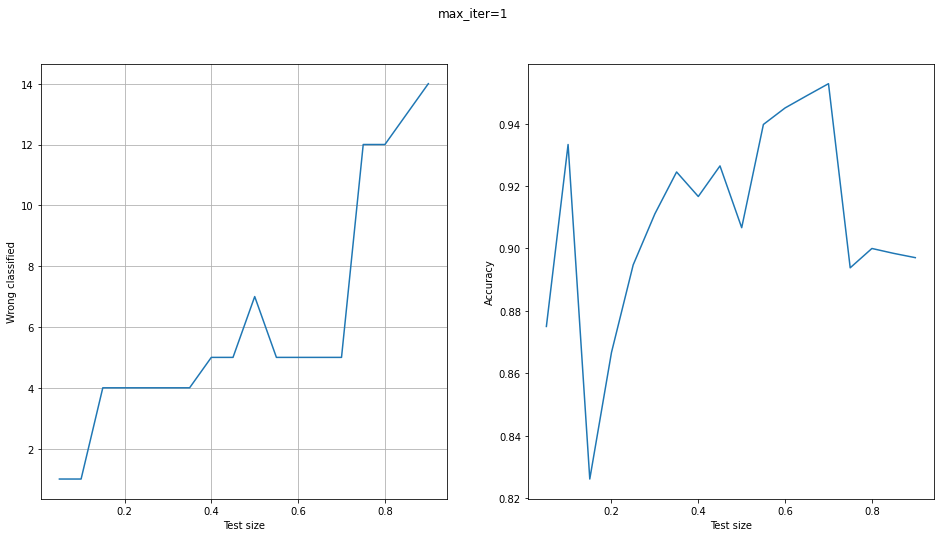


Рисунок 29. *SVC(max\_iter=1)*

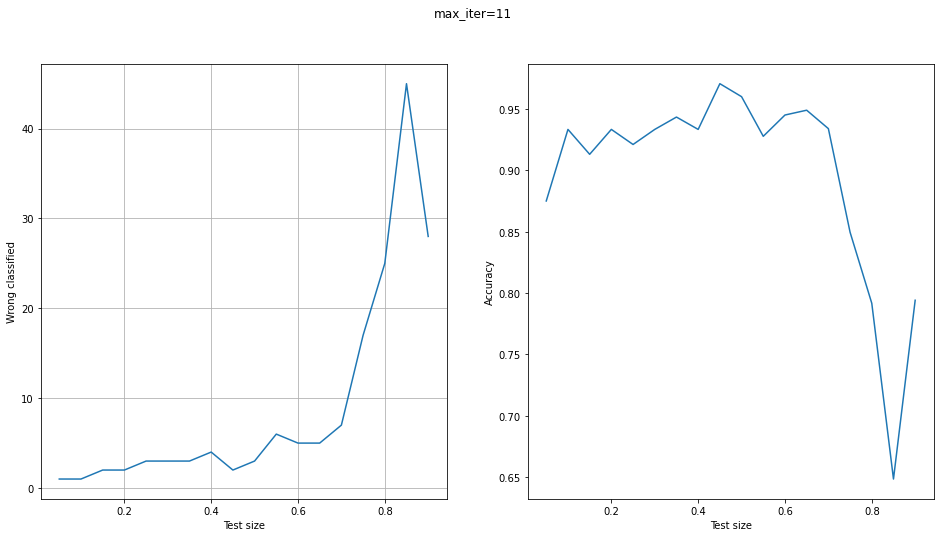


Рисунок 30. *SVC(max\_iter=11)*

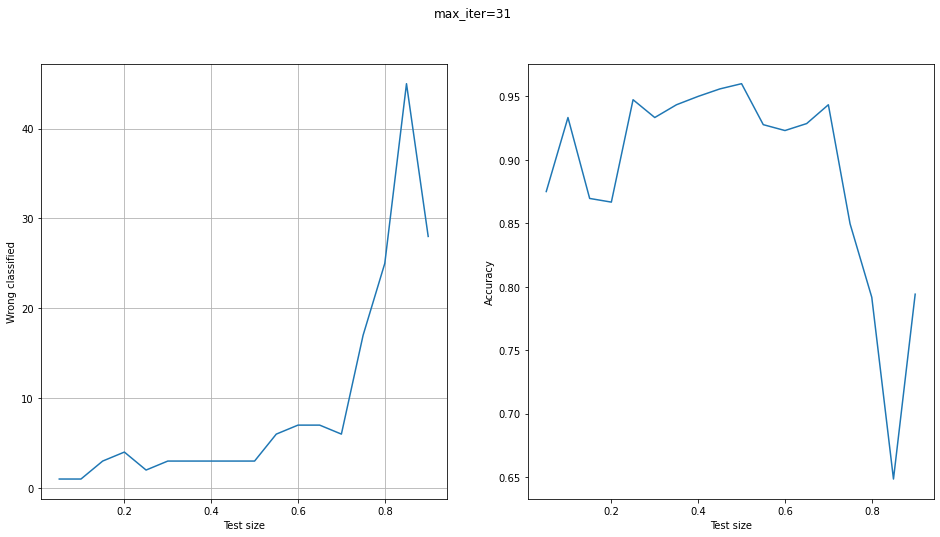


Рисунок 31. SVC(max\_iter=31)

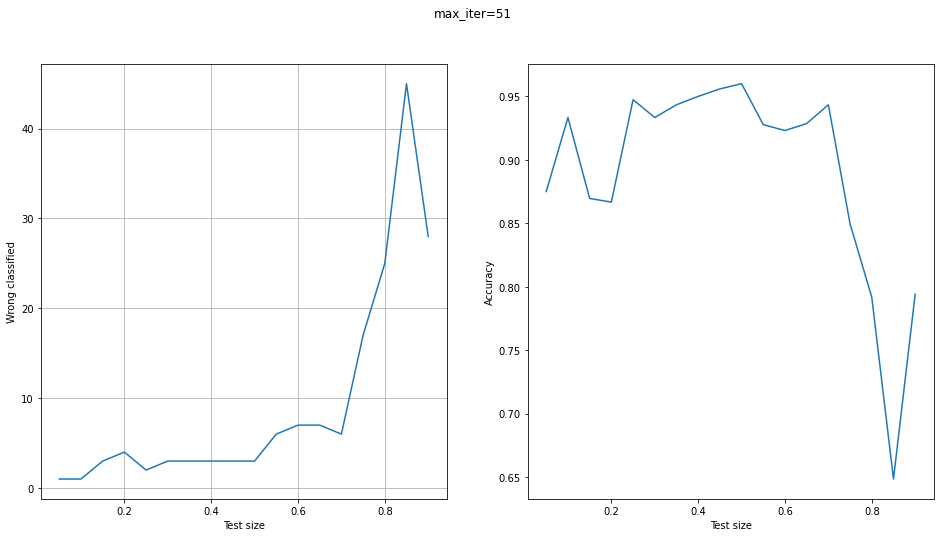


Рисунок 32. *SVC(max\_iter=51)*

1. Исследование методов *NuSVC*, *LinearSVC*

График зависимости количества неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки для методов *NuSVC* и *LinearSVC* представлены на рисунках 33 и 34 соответственно.

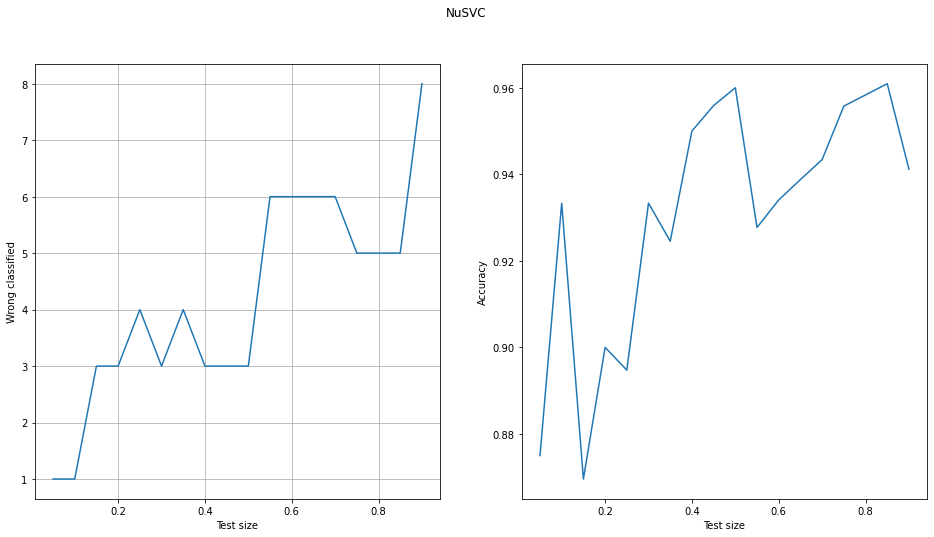


Рисунок 33. Зависимость количества неправильно классифицированных данных и точности классификации от размера тестовой выборки *NuSVC*.

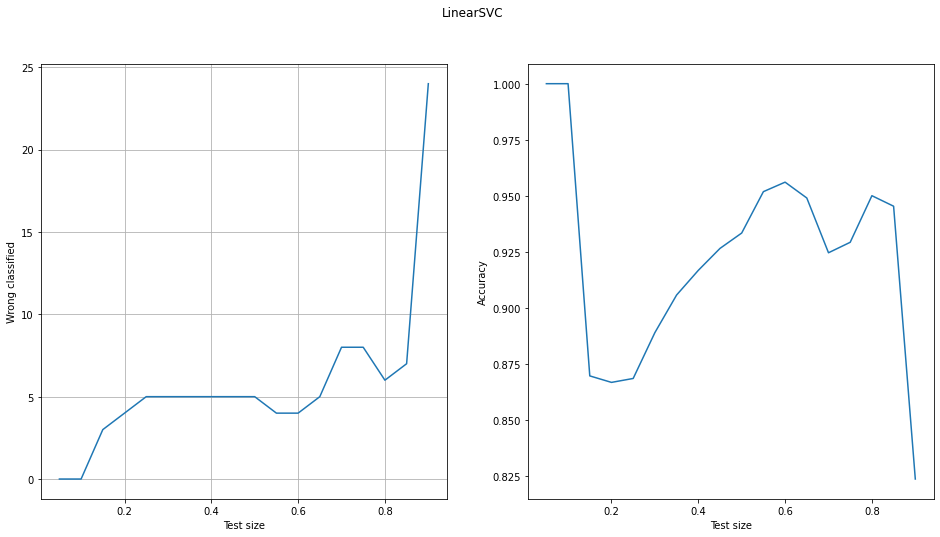


Рисунок 34. Зависимость количества неправильно классифицированных данных и точности классификации от размера тестовой выборки *LinearSVC*.

# Вывод

В ходе лабораторной работы исследованы методы классификации: *LinearDiscriminantAnalysis* и *SupportVectorMachines*.

*LinearDiscriminantAnalysis* – представляет собой метод понижения размерности, основанный на расстоянии между классами. Выбираются компоненты, в которых классы находятся наиболее далеко друг от друга, а сами при этом максимально сжаты. Поскольку метод сводится к нахождению классовых компонент, его можно использовать для классификации.

*SupportVectorMachines* – метод классификации данных, основанный на линейном разделении пространства наблюдений. Несмотря на линейное разделение, разделяется пространство повышенной, с помощью ядра, размерности, что может приводить к нелинейным границам классов.