

**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. 6304

\_\_\_\_\_

Антонов С.А.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2020

## Цель работы:

Ознакомиться с методами классификации модуля Sklearn.

## Ход работы:

### Загрузка данных

1. На данном этапе был скачан и загружен датасет в датафрейм Pandas.

```
data = pd.read_csv('iris.data', header=None)
data.head()
```

	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

Рисунок 1 Загруженный датасет

2. Были выведены данные и их метки, тексты меток были преобразованы к числам с помощью LabelEncoder:

```
X = data.iloc[:, :4].to_numpy()
labels = data.iloc[:, 4].to_numpy()
le = preprocessing.LabelEncoder()
Y = le.fit_transform(labels)
```

3. Выборка была разбита на обучающую и тестовую с помощью train\_test\_split:

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.5)
```

### Линейный дискриминантный анализ

1. Была проведена классификация данных методом LinearDiscriminantAnalysis и выявлено 3 неправильно классифицированных наблюдения.
2. С помощью метода score была получена точность классификации, которая составляет 0.98%.
3. Были построены графики зависимости неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки.

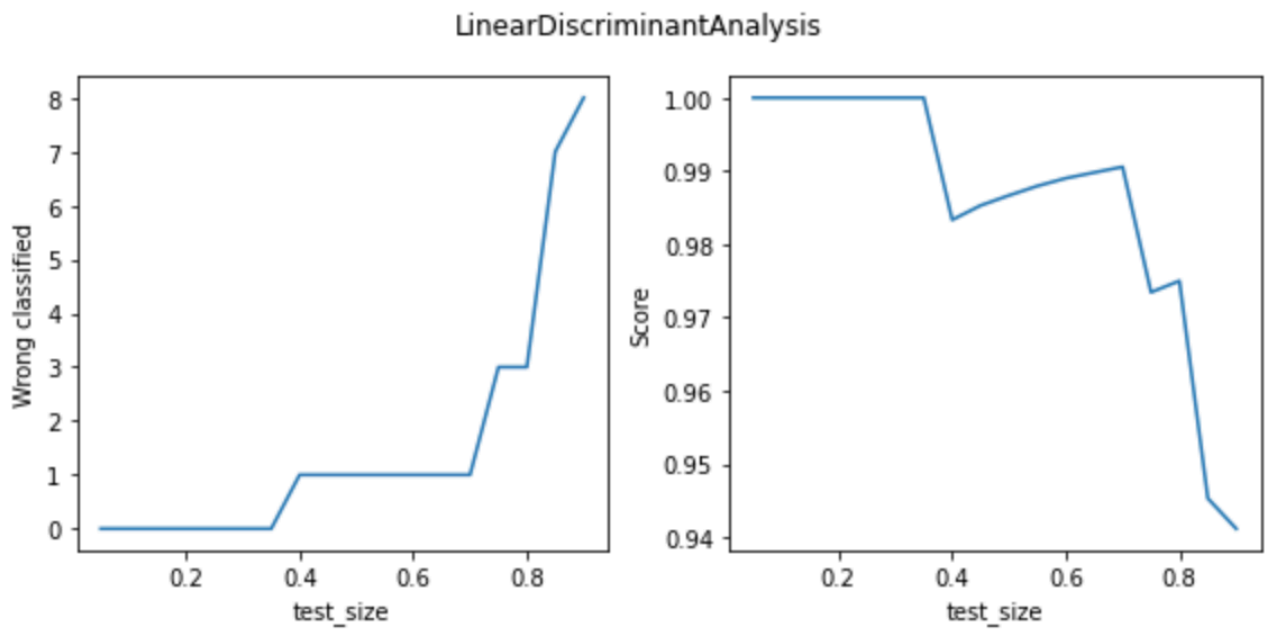


Рисунок 2 Графики для метода LDA

4. Функция transform может применяться для уменьшения размерности данных. Результаты представлены на рисунке 3.

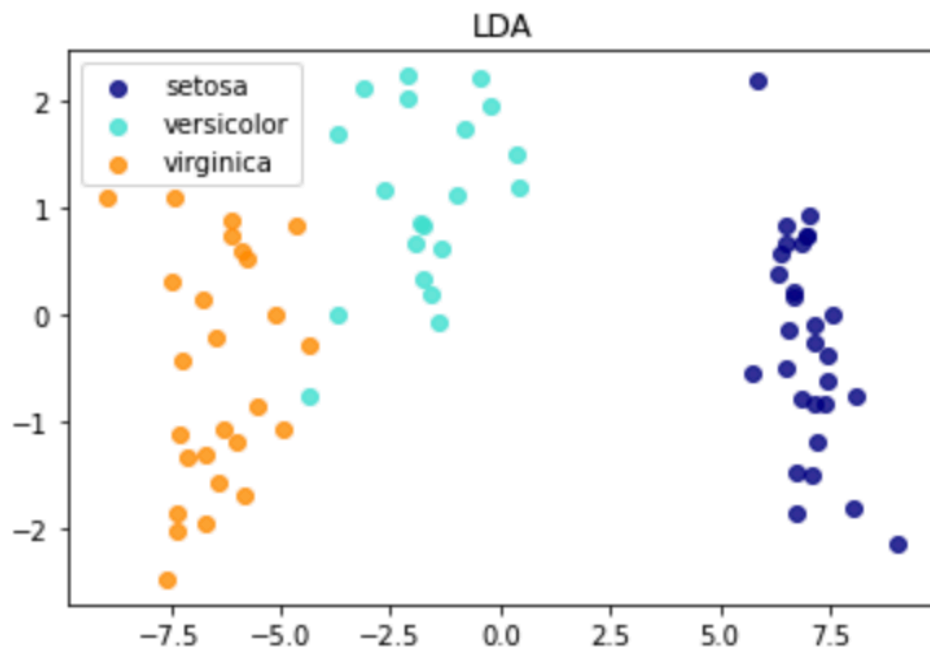


Рисунок 3 результаты transform

5. Работа классификатора была исследована при различных параметрах solver, shrinkage.

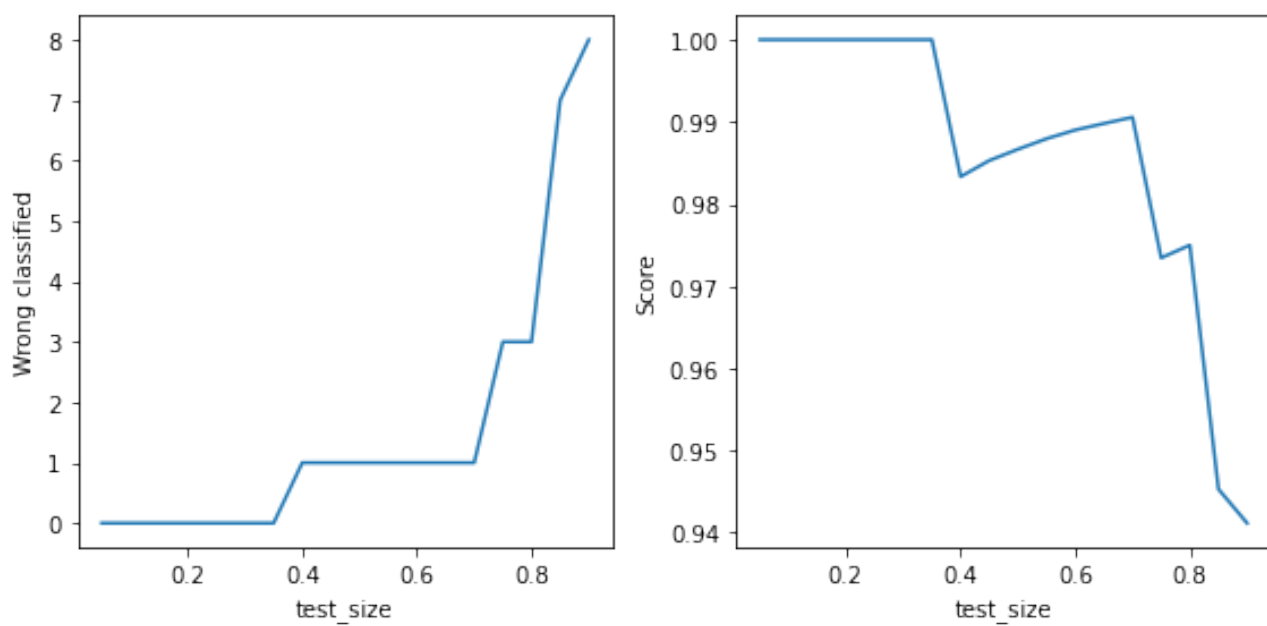


Рисунок 4 `solver=svd, shrinkage=no`

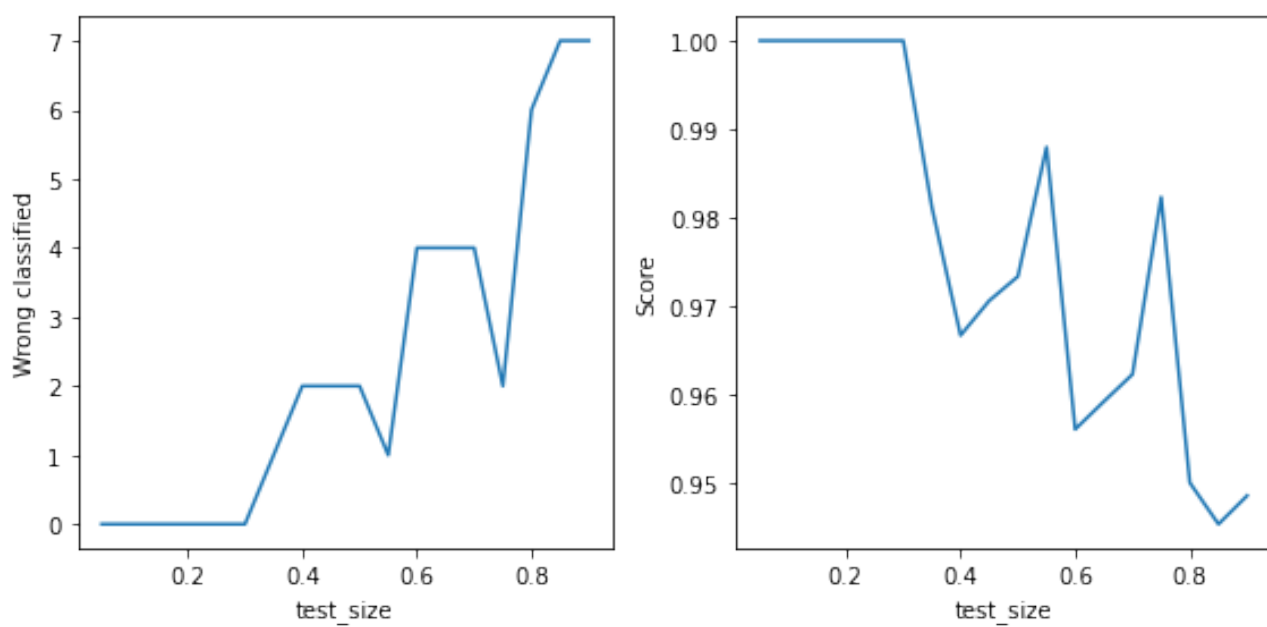


Рисунок 5 `solver=lsqr, shrinkage=no`

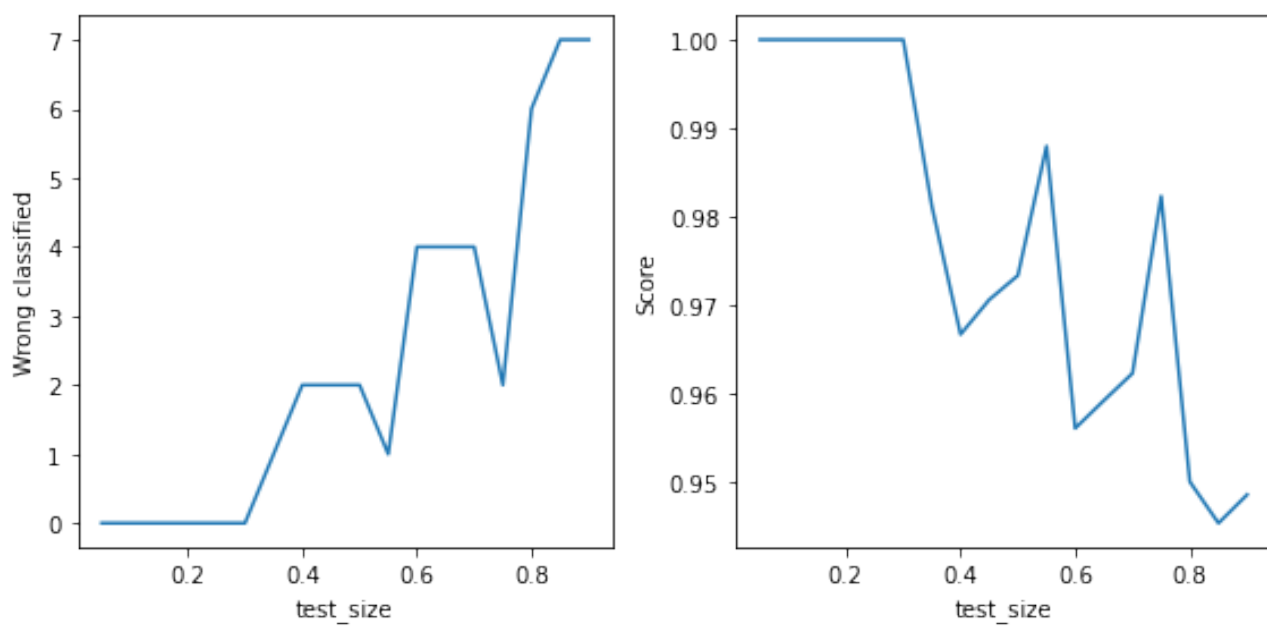


Рисунок 6 solver=lsqr, shrinkage=auto

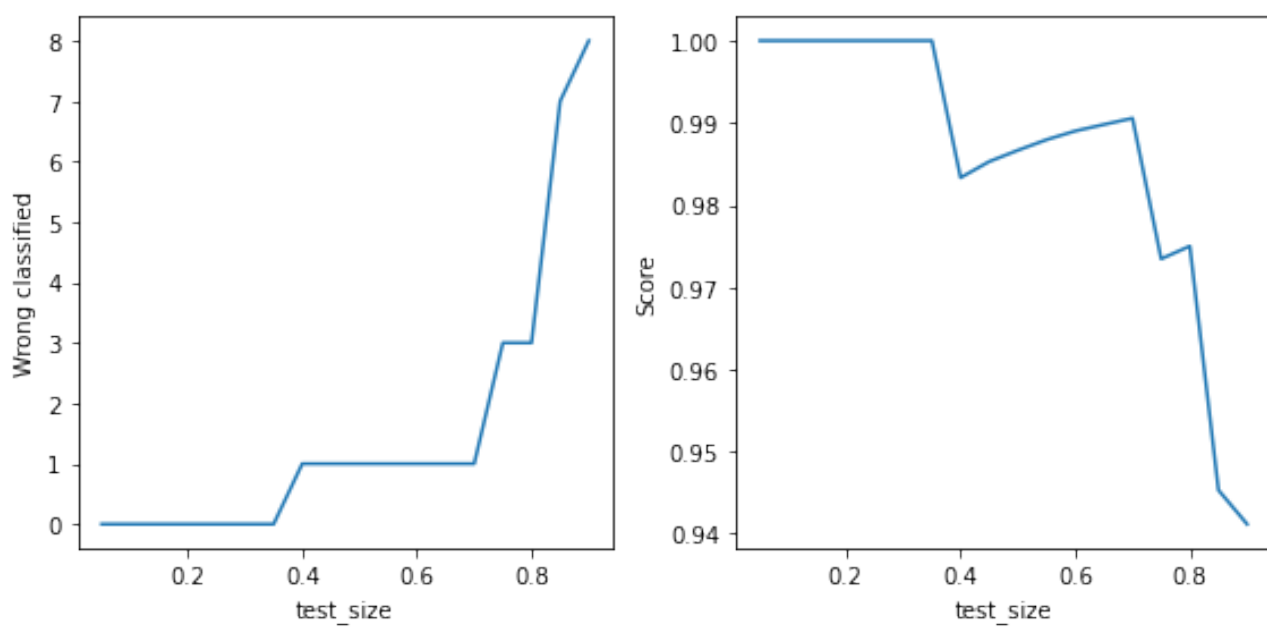


Рисунок 7 solver=eigen, shrinkage=auto

6. Классу с номером 1 задана априорная вероятность классу 0.7, остальным классам заданы равные априорные вероятности.

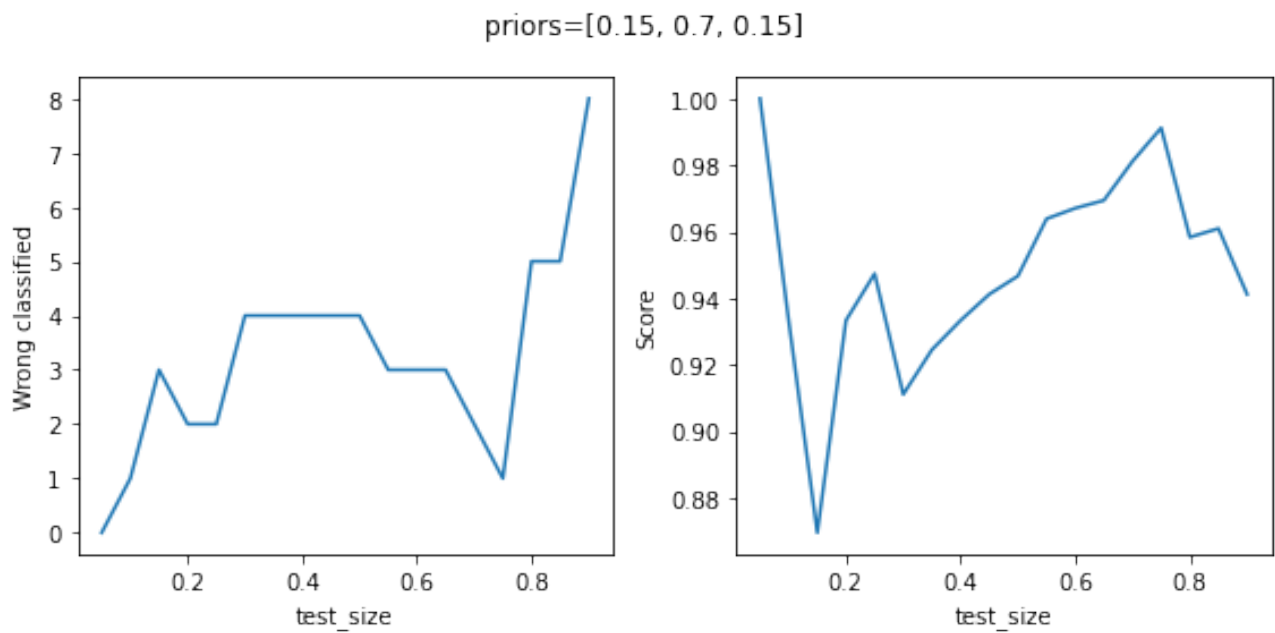


Рисунок 8 Графики зависимости от размера тестового набора

### Метод опорных векторов

1. Была проведена классификация SVM на тех же данных и выведено количество неправильно классифицированных наблюдений:  
Wrong classified: 4
2. С помощью метода score была получена точность классификации, которая составляет 0.96%.
3. Атрибут support\_ хранит индексы опорных векторов, support\_vectors\_ сами опорные вектора, n\_support\_ количество опорных векторов для каждого класса.
4. Построен график неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки.

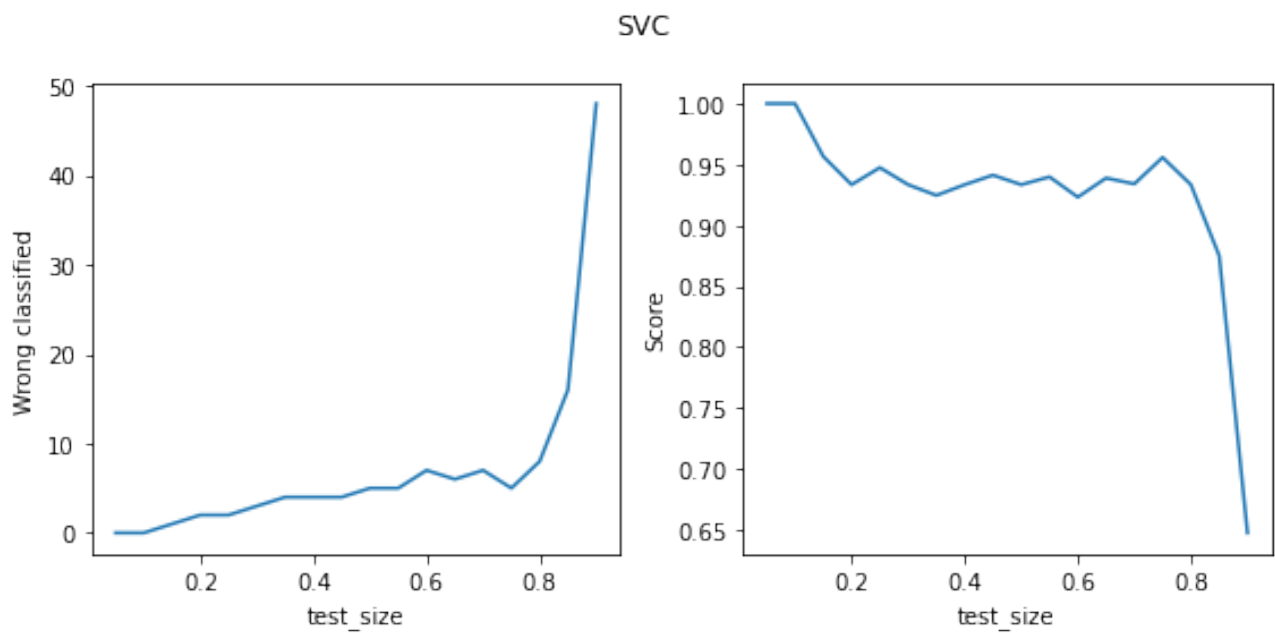


Рисунок 9 График зависимости от размера тестовой выборки

5. Исследована работа метода опорных векторов при различных значениях параметров kernel, degree, max\_iter.

a) Kernel – тип ядра, который будет использоваться внутри алгоритма

kernel	Wrong classified	Score
linear	2	0.97
poly	6	0.95
rbf	4	0.96
sigmoid	54	0.38

b) Degree – степень полиномиальной функции ядра (при kernel = poly)

degree	Wrong classified	Score
1	5	0.96
2	6	0.96
3	6	0.97
4	5	0.98
5	3	0.98
6	3	0.98
7	3	0.98

с) Max\_iter ограничение на количество итераций.

Max_iter	Wrong classified	Score
1	9	0.96
2	8	0.98
3	5	0.97
4	3	0.97
5	1	0.97
6	3	0.97
7	3	0.97

6. Исследована работа методов:

а) NuSVC – имеет параметр для управления количеством опорных векторов.

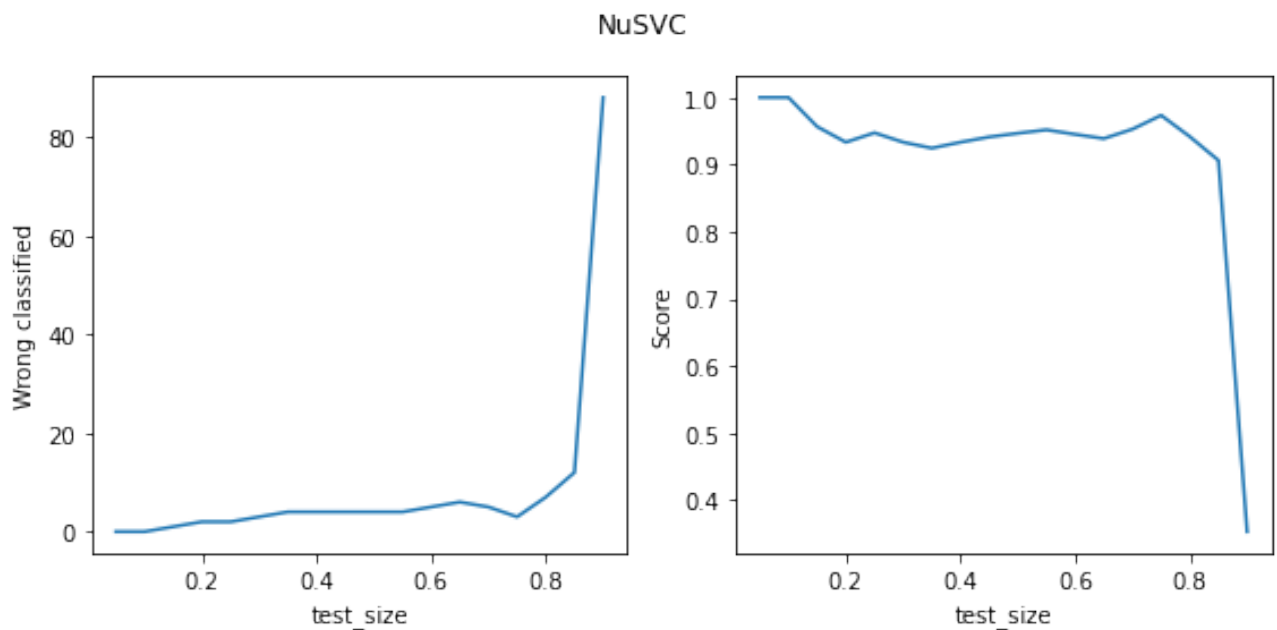


Рисунок 10 Графики зависимости от размера тестовой выборки NuSVC

б) LinearSVC – аналогичен SVC с kernel=linear, но лучше масштабируется.



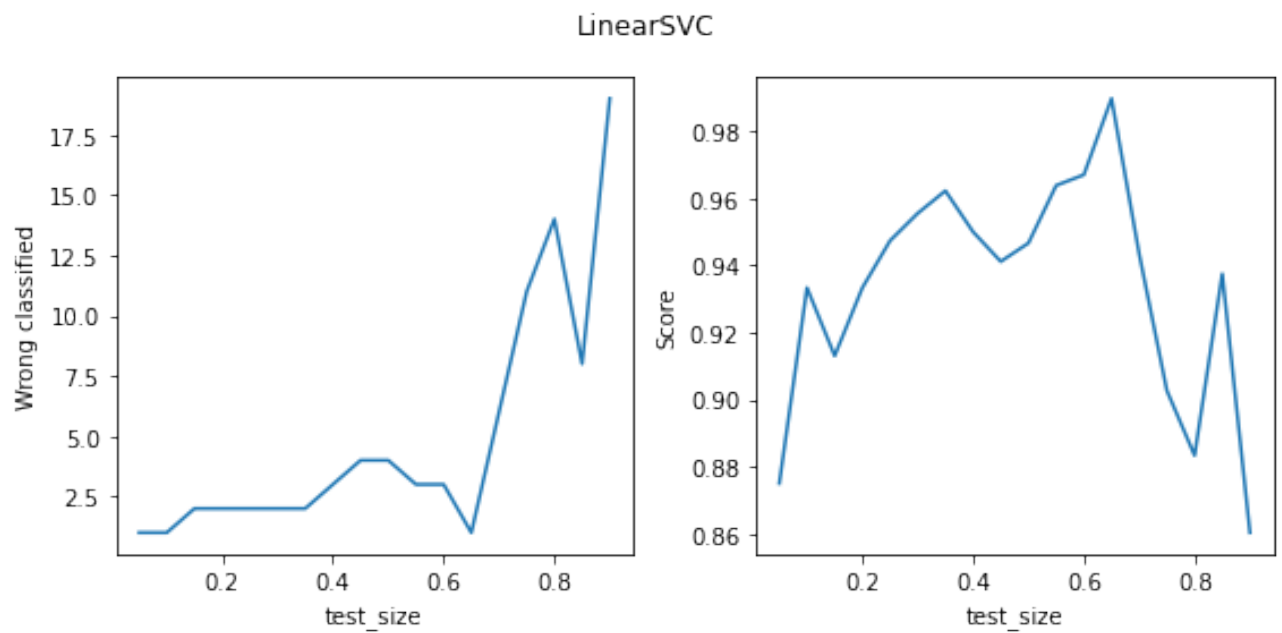


Рисунок 11 Графики зависимости от размера тестовой выборки LinearSVC

### Выводы:

В ходе выполнения лабораторной работы было произведено знакомство с классификацией методами GaussianNB, MultinomialNB, ComplementNB, BernoulliNB и DecisionTreeClassifier модуля Sklearn.