**Министерство образования и науки 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» (ИУ)

КАФЕДРА «Системы обработки информации и управления» (ИУ5)

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ*

## *НА ТЕМУ:*

**«Тематическое моделирование на основе учебных материалов»**

Студент группы ИУ5-32М _____
Колпаков М.О.

Руководитель _____ Гапанюк Ю.Е.

*2020 г.*

Оглавление



## ВВЕДЕНИЕ

В данной работе стояла задача исследовать датасет по звездам и классифицировать по набору данных является она пульсаром или нет – по сути классическая задача бинарной классификации. Для проверки результата

зададим точность модели в 80%, что является хорошим показателем для обучающей задачи. Решено применить метод к-ближайших соседей.

## ОСНОВНАЯ ЧАСТЬ

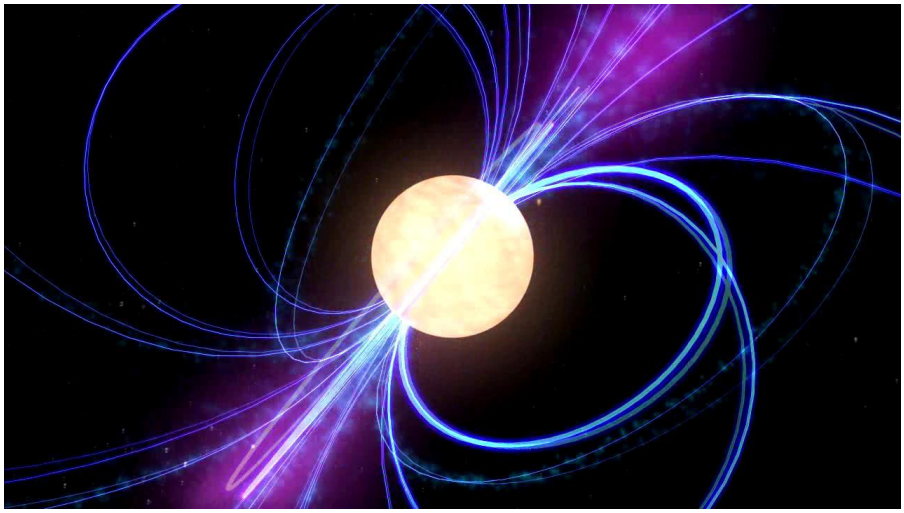


Рис.1 Пульсар

Пульсар (от pulse и -ar как в quasar) - сильно намагниченная вращающаяся компактная звезда (обычно нейтронные звезды, но также белые карлики), испускающая пучки электромагнитного излучения из своих магнитных полюсов. Это излучение можно наблюдать только тогда, когда луч излучения направлен на Землю (так же, как маяк можно увидеть, только когда свет направлен в направлении наблюдателя), и оно отвечает за импульсное появление излучения. Нейтронные звезды очень плотны и имеют короткие регулярные периоды вращения. Это дает очень точный интервал между импульсами, который составляет от миллисекунд до секунд для отдельного пульсара. Пульсары - один из кандидатов на роль источника космических лучей сверхвысокой энергии.

### История открытия

Для наблюдения компактных источников радиоизлучения Энтони Хьюиш и его коллеги в 1960-х годах создали радиотелескоп. Среди исследователей была аспирантка Джоселин Белл, которая собирала материал для своей диссертации. Джоселин Белл обработала наблюдения, обновила записывающие устройства телескопов и нашла сигналы от компактных

источников. Через 2 месяца работы она нашла сигналы, которые никак не могла классифицировать, так как их нельзя было отнести ни к помехам, ни к известным источникам. Поэтому предполагалось, что источником излучения является звезда. Но период излучения импульсов составлял 1,33 с, такие показатели не характерны для звезд и не могут быть следствием процессов, происходящих в звездах. Гипотеза о земном происхождении была отброшена, когда аспирант Энтони Хьюиша продолжал наблюдать странное излучение. К наблюдению присоединились и другие ученые. В результате того, что был обнаружен только один такой источник, стали возникать предположения, что периодический источник является следствием деятельности внеземной разумной цивилизации. По этой причине первый радиопульсар был назван «Зеленые человечки», сокращенно LGM-1. Позже Джоселин обнаружила еще три источника с такой маленькой периодичностью в совершенно разных областях неба. Тогда стало ясно, что этот источник представляет собой новый класс астрономических объектов.

Как оказалось, астрономы и раньше регистрировали подобные периодические радиосигналы, но их ошибочно приняли за помехи, вызванные деятельностью человека.

**Кандидаты в пульсары**

Предполагалось, что излучение приходит на Землю из относительно небольшого участка космоса - об этом свидетельствовал характер принимаемых сигналов. Также высокая стабильность пульсара указывает на то, что источником излучения является не скопление газа или плазмы, а жесткая система. Периодическое излучение можно объяснить разными способами: орбитальное вращение или колебания самого объекта-источника.

Орбитальное вращение источника периодического излучения - это взаимное вращение двух объектов, но такая система с таким малым периодом будет излучать такие мощные гравитационные волны, что они замедляют вращение объектов и приводят к их столкновению в течение одного года. Среди прочего, приближение привело к уменьшению периода излучения, тогда как у пульсаров он увеличивается со временем. Собственные пульсации такого объекта также привели бы к уменьшению периода. Остается только один вариант - с собственным вращением объекта.

Основные кандидаты на роль пульсаров - компактные объекты: белые карлики, черные дыры и нейтронные звезды. После того, как пульсары были обнаружены в течение примерно 30 миллисекунд, гипотеза о том, что они могут быть белыми карликами, была отвергнута. Дело в том, что у белых карликов не могло быть такого короткого периода вращения, так как они были бы разрушены под действием центробежной силы, то есть просто разлетелись бы. Черные дыры вообще не могут излучать сами по себе. В конечном итоге возможна конечная скорость вращения.

**Физика радиопульсаров**

Быстрое вращение нейтронной звезды приводит к потере части ее звездного вещества. То есть, быстро вращаясь, нейтронная звезда испускает элементарные частицы, образующие плазму.

Как оказалось, радиопульсары обладают сильными магнитными полями (1010-1013 Гс). Подобные поля наблюдаются у некоторых нейтронных звезд, что делает их кандидатами в радиопульсары. Внутри полярных шапок силовые линии электромагнитного поля направлены таким образом, что по отношению к излучаемой плазме они образуют продольное электрическое поле. Это поле имеет разность потенциалов между центром и краем полярной шапки, что приводит к ускорению упомянутых эмитированных элементарных частиц до ультрарелятивистских энергий. Достигая столь

высоких энергий, частицы выделяют часть энергии в виде излучения, в том числе в радиодиапазоне. Собирая все вышесказанное, мы можем представить радиопульсар как быстро вращающуюся нейтронную звезду с сильным магнитным полем, которое излучает плазму на своих полюсах, которая, в свою очередь, излучает электромагнитные волны.

**Основные характеристики**

Кроме координат, пульсары различают по их характеристикам:

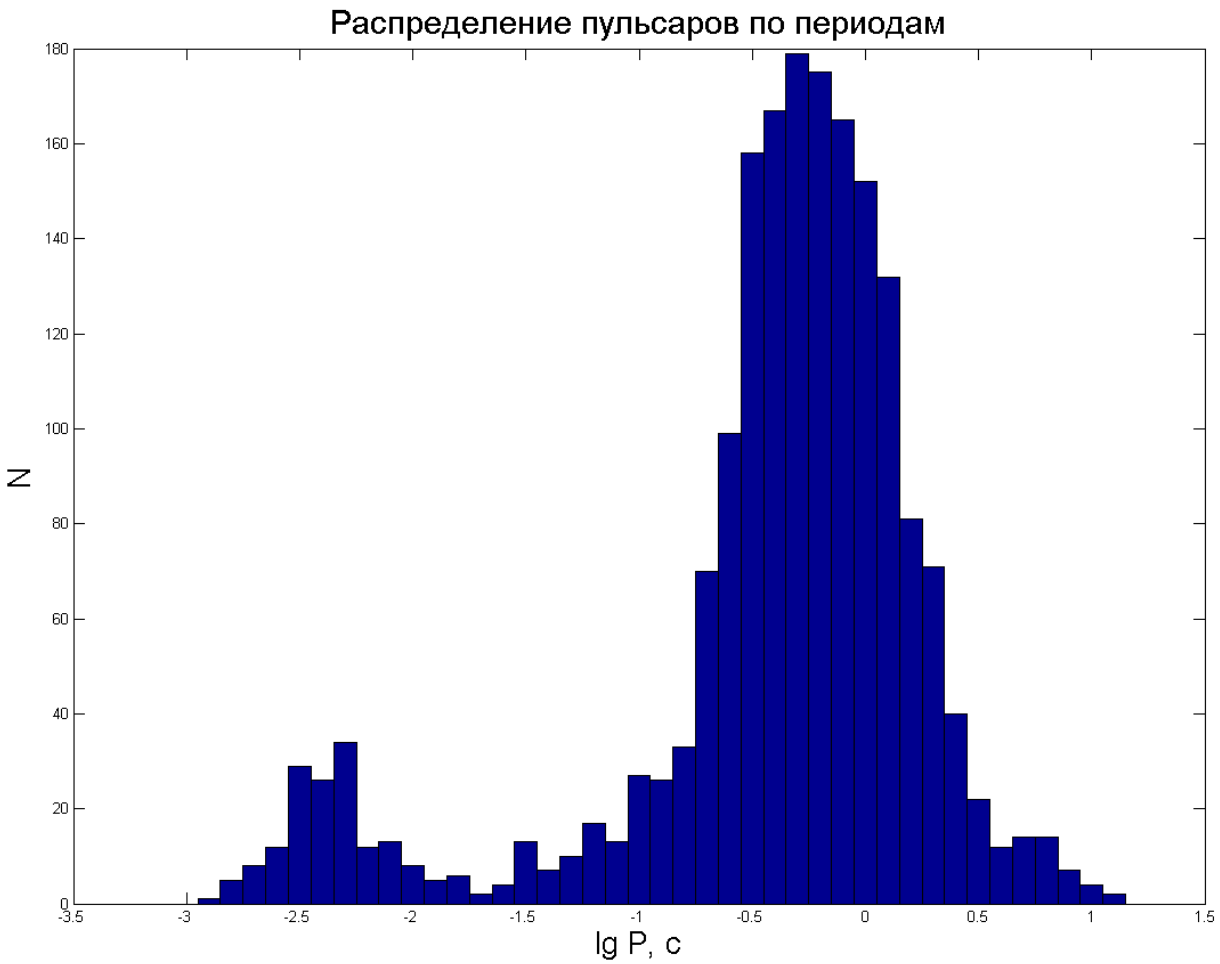



Рис.2 Распределение пульсара

- Период ротации. Распределение пульсаров по периоду дает максимум в районе 0,6 секунды. То есть у большинства пульсаров, называемых «нормальными», такой период вращения. Есть еще один ярко

выраженный максимум, в несколько раз меньший, и он находится в районе 4 мс, потому что пульсары этого типа называют «миллисекундными».

- Производная периода - это параметр, определяющий скорость увеличения периода вращения пульсара. Как известно, почти все наблюдаемые пульсары имеют период, монотонно увеличивающийся во времени, то есть вращение замедляется.

К 2011-му году количество открытых радиопульсаров перешло черту в 1970 объектов. Согласно теоретическим подсчетам в галактике Млечный Путь может находится порядка 240 000 радиопульсаров.

# МЕТОД K-БЛИЖАЙШИХ СОСЕДЕЙ

Метод k ближайших соседей (kNN) - очень популярный метод классификации, который также иногда используется в задачах регрессии. Если между примерами успешно введена метрика расстояния, то похожие примеры с большей вероятностью будут в одном классе, чем в разных.
Если при создании объявления вы не знаете, в какую категорию отнести товар для автомобильного зарядного устройства, вы можете найти 6 похожих зарядных устройств, и если 5 из них находятся в категории «Автомобильные аксессуары», а одно из них - в другой, то лучше выбрать первую категорию.
Для классификации образца выполняем следующие действия:

- Рассчитываем расстояние до каждого объекта по образцу
- Выберите k образцов объектов, расстояние до которых минимально.
- Класс с наибольшим количеством k ближайших соседей - это класс классифицируемого объекта.

Замечательная особенность такого подхода - его лень. Поскольку расчеты начинаются только в момент классификации примера и заранее, только при наличии обучающих примеров модель не строится.
В классификации k-NN выходом является принадлежность к классу. Объект классифицируется множеством голосов его соседей, причем объект назначается классу, наиболее распространенному среди его ближайших k соседей (k - положительное целое число, обычно небольшое). Если k = 1, то объект просто присваивается классу этого единственного ближайшего соседа. В регрессии k-NN выходом является значение свойства объекта. Это значение является средним из значений k ближайших соседей. k-NN - это тип обучения на основе экземпляров или ленивого обучения, при котором функция аппроксимируется только локально, а все вычисления откладываются до оценки функции. Поскольку этот алгоритм основан на расстоянии для классификации, если признаки представляют разные физические единицы или имеют совершенно разные масштабы, то

нормализация обучающих данных может значительно повысить его точность. Как для классификации, так и для регрессии полезным методом может быть присвоение весов вкладам соседей, чтобы более близкие соседи вносили больший вклад в среднее значение, чем более отдаленные. Например, обычная схема взвешивания заключается в присвоении каждому соседу веса 1 / d, где d - расстояние до соседа. Соседи берутся из набора объектов, для которых известен класс (для классификации k-NN) или значение свойства объекта (для регрессии k-NN). Это можно рассматривать как обучающий набор для алгоритма, хотя явного шага обучения не требуется. Особенность алгоритма k-NN в том, что он чувствителен к локальной структуре данных.

Стоит отметить, что метод ближайших соседей - хорошо изученный подход (в машинном обучении, эконометрике и статистике, возможно, больше известно только о линейной регрессии). Для метода ближайших соседей существует много важных теорем, утверждающих, что на «бесконечных» выборках это оптимальный метод классификации. Авторы классической книги «Элементы статистического обучения» считают kNN теоретически идеальным алгоритмом, применимость которого просто ограничена вычислительной мощностью и проклятием размеров.

**Класс KNeighborsClassifier в Scikit-learn**

sklearn.neighbors.KNeighborsClassifier основные пар-ры класса:

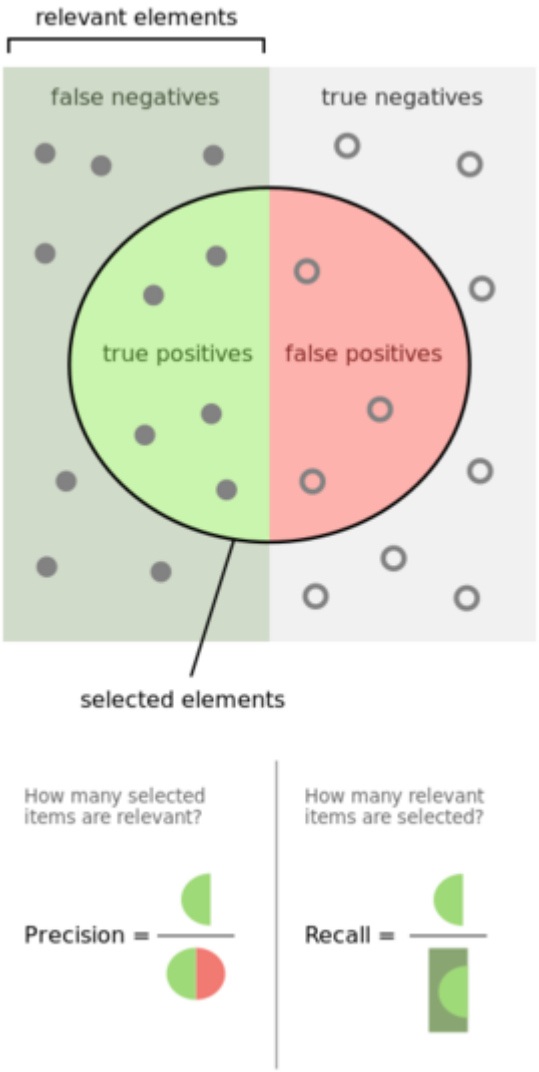
- weights: "uniform" (все веса равны), "distance" (вес обратного расстояния до тестового примера) или другая определенная функция
- algorithm (опционально): "brute", "ball_tree", "KD_tree", или "auto". В первом случае ближайшие соседи для каждого тестового примера считаются перечислением обучающей выборки. Во втором и третьем случае расстояние между примерами сохраняется в дереве, что ускоряет поиск ближайших соседей. Если указан параметр «auto», подходящий

способ поиска соседей будет выбран автоматически на основе обучающей выборки.

- leaf_size (опционально): полный перебор в случае выбора BallTree или KDTree для нахождения соседей
- metric: "minkowski", "manhattan", "euclidean", "chebyshev" и другие

# МЕТРИКИ КАЧЕСТВА И ПОДБОР ПАРАМЕТРОВ

## Accuracy, precision и recall




Прежде чем перейти к метрикам, необходимо использовать концепцию описания метрик в терминах ошибок - матрицу ошибок.

У нас есть два класса и алгоритм предсказывает принадлежность каждого объекта к одному из классов, тогда матрица ошибок будет выглядеть так:

| | y=1 | y=0 |
|---|---|---|
| y^=1 | True Positive (TP) | False Positive (FP) |
| y^=0 | False Negative (FN) | True Negative (TN) |

Таб.3 Метрики качества

Здесь y^ — это ответ алгоритма на объекте, а y — истинная метка класса на этом объекте.

Таким образом, ошибки классификации бывают двух видов: False Negative (FN) и False Positive (FP).

Обучение алгоритма и построение матрицы ошибок

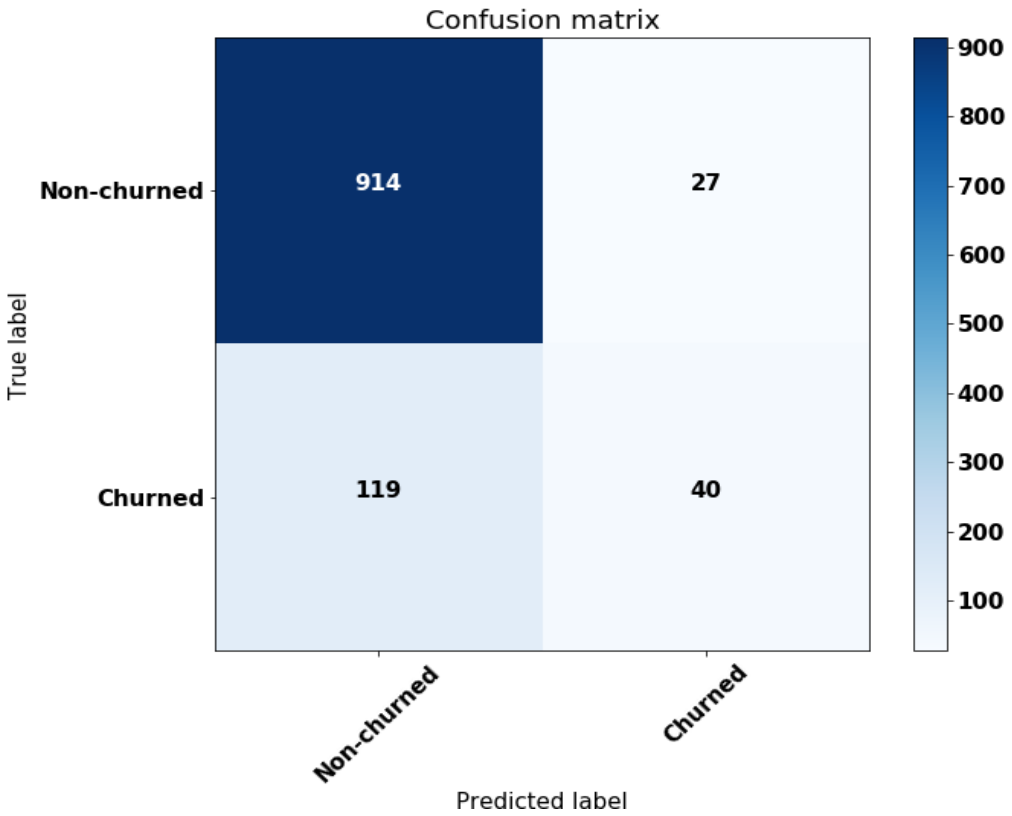



Рис.4 Матрица классификации

**Accuracy**

Интуитивно понятной, очевидной и почти неиспользуемой метрикой является accuracy — доля правильных ответов алгоритма:

$$accuracy=TP+TNTP+TN+FP+FN$$

Эта метрика бесполезна в задачах с неравными классами, и это легко показать на примере.

В то же время наша модель не имеет абсолютно никакой предсказательной силы, поскольку изначально мы хотели идентифицировать спам-сообщения. Преодолеть это нам поможет переход от общей метрики для всех классов к отдельным показателям качества классов.

**Precision, recall и F-мера**

Для оценки качества работы алгоритма на каждом из классов по отдельности введем метрики precision (точность) и recall (полнота).

$$precision=TPTP+FP$$

$$recall=TPTP+FN$$

Точность можно интерпретировать как долю объектов, названных классификатором как положительные и в то же время действительно положительные, а напоминание показывает, какая доля объектов положительного класса от всех объектов положительного класса была найдена алгоритмом.

Есть несколько разных способов объединить точность и обратную связь в агрегированный показатель качества. F-мера (в общем случае $F\beta$) — среднее гармоническое precision и recall

$$F\beta=(1+\beta2)\cdot precision\cdot recall(\beta2\cdot precision)+recall$$

β в данном случае определяет вес точности в метрике, и при β=1 это среднее гармоническое (с множителем 2, чтобы в случае precision = 1 и recall = 1)

## ПРАКТИЧЕСКАЯ ЧАСТЬ

Задачу было решено решать в Jupyter Notebook на python 3 с помощью библиотек scikit-learn и метода к-ближайших соседей, как самый простой и надежный способ классификации.

| | Mean of the integrated profile | Standard deviation of the integrated profile | Excess kurtosis of the integrated profile | Skewness of the integrated profile | Mean of the DM-SNR curve | Standard deviation of the DM-SNR curve | Excess kurtosis of the DM-SNR curve | Skewness of the DM-SNR curve | target_class |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 140.562500 | 55.683782 | -0.234571 | -0.699648 | 3.199833 | 19.110426 | 7.975532 | 74.242225 | 0 |
| 1 | 102.507812 | 58.882430 | 0.465318 | -0.515088 | 1.677258 | 14.860146 | 10.576487 | 127.393580 | 0 |
| 2 | 103.015625 | 39.341649 | 0.323328 | 1.051164 | 3.121237 | 21.744669 | 7.735822 | 63.171909 | 0 |

Исходный датасет имел сдеуюущие атрибуты

Рис. 5 Исходный датасет

Описание полей:

- # Среднее значение интегрированного профиля float64
- # Стандартное отклонение интегрированного профиля float64
- # Лишний эксцесс интегрированного профиля float64
- # Асимметрия интегрированного профиля float64
- # Среднее значение кривой DM-SNR float64
- # Стандартное отклонение кривой DM-SNR float64
- # Избыточный эксцесс кривой DM-SNR float64
- # Асимметрия кривой DM-SNR float64

**Считывание данных из файла**

```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:90% !important; }</style>"))
from IPython.core.interactiveshell import InteractiveShell  #to run all statements in cell, not only the last
InteractiveShell.ast_node_interactivity = "all"
import warnings
warnings.filterwarnings('ignore')
```

Исходное кол-во классов 2, примеров:

0 16259

1 1639

```
import pandas as pd
data = pd.read_csv('pulsar_stars.csv', sep=",")
data.head(3)
data.dtypes
data.count()
```

**Очистка данных**

```
data.target_class.value_counts()
# data = data[(data.target_class == 0) | (data.target_class == 1)]   #Очистка таргета от мусора, если бы понадобилась
# data.target_class.value_counts()
```

**Разделение данных на атрибуты для обучения и проверки**

```
features = data.drop(['target_class'], axis=1)
target_tmp = data['target_class']

target = pd.DataFrame({'target_class':target_tmp.index,
'target_class':target_tmp.values})

features.head(3)
type(features)
features.shape
target.head(3)
type(target)
target.shape
```

**Разделение датасета на обучающую и тестовые выборки**

```
PULSAR_X_train, PULSAR_X_test, PULSAR_y_train, PULSAR_y_test =
train_test_split(features, target, test_size=0.2, random_state=1)
PULSAR_X_train.shape
PULSAR_X_test.shape
PULSAR_y_train.shape
PULSAR_y_test.shape
```

```
(14318, 8)
```

Out[103]:

```
(3580, 8)
```

Out[103]:

```
(14318, 1)
```

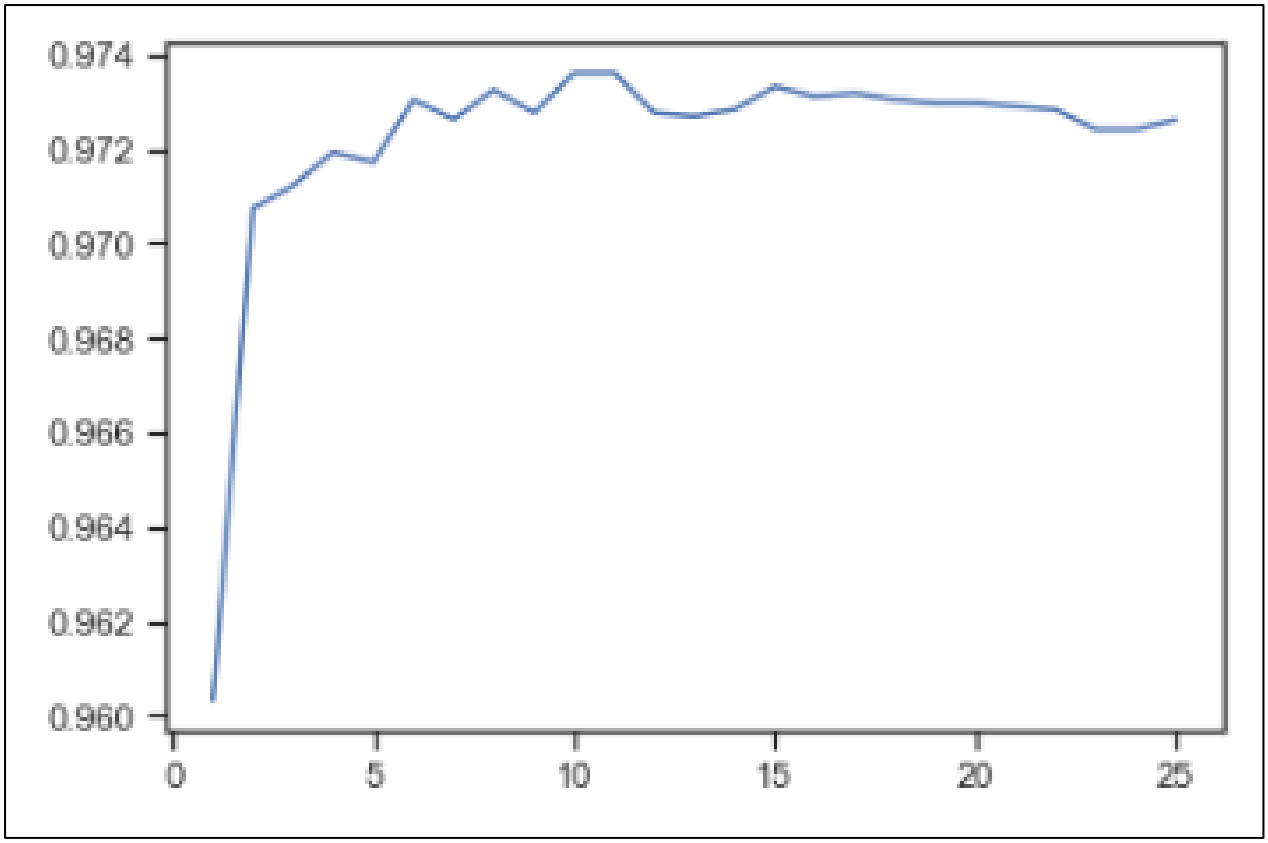
Out[103]:

```
(3580, 1)
```

**Поиск оптимального параметра для метода**

```
n_range = np.array(range(1,26,1))
tuned_parameters = [{'n_neighbors': n_range}]
tuned_parameters
clf_gs = GridSearchCV(KNeighborsClassifier(), tuned_parameters, cv=5,
scoring='accuracy')
clf_gs.fit(PULSAR_X_train, PULSAR_y_train)
clf_gs.best_estimator_
clf_gs.best_score_
clf_gs.best_params_
```

**В итоге имеем оптимальное значение: 10 соседей**




{'n_neighbors': 10}

Рис.6 График точность обучения от кол-ва соседей

**Обучение, тест и результаты**

```
clf_gs.best_estimator_.fit(PULSAR_X_train, PULSAR_y_train)
target2_0 = clf_gs.best_estimator_.predict(PULSAR_X_train)
target2_1 = clf_gs.best_estimator_.predict(PULSAR_X_test)

accuracy_score(PULSAR_y_train, target2_0)
accuracy_score(PULSAR_y_test, target2_1)

balanced_accuracy_score(PULSAR_y_test, target2_1)
```

На выходе имеем отличную метрику точности нашей модели:

```
0.9753457186757927
0.9743016759776536
0.8830569555695558
```

**Построение графика зависимости точности обучения от кол-ва обучающей выборки:**

plot_learning_curve(KNeighborsClassifier(n_neighbors=10), 'n_neighbors=10', features, target, cv=10)

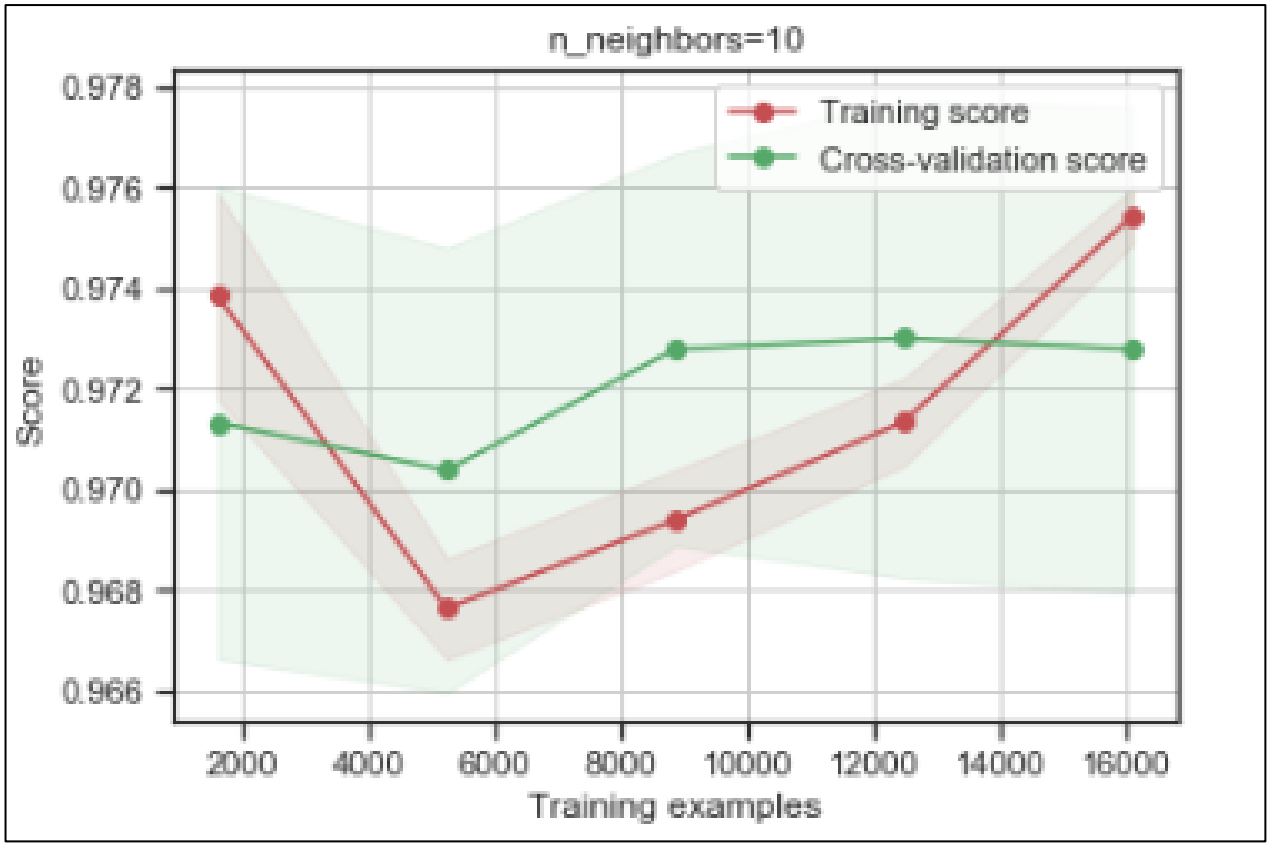



Рис.7 График точность обучения от кол-ва обучающей выборки

**Построение графика точности обучния и теста**

```
n_range2 = np.array(range(1,16,1))
plot_validation_curve(KNeighborsClassifier(), 'knn',
                      features, target,
                      param_name='n_neighbors', param_range=n_range2, cv=5,
scoring="accuracy")
```

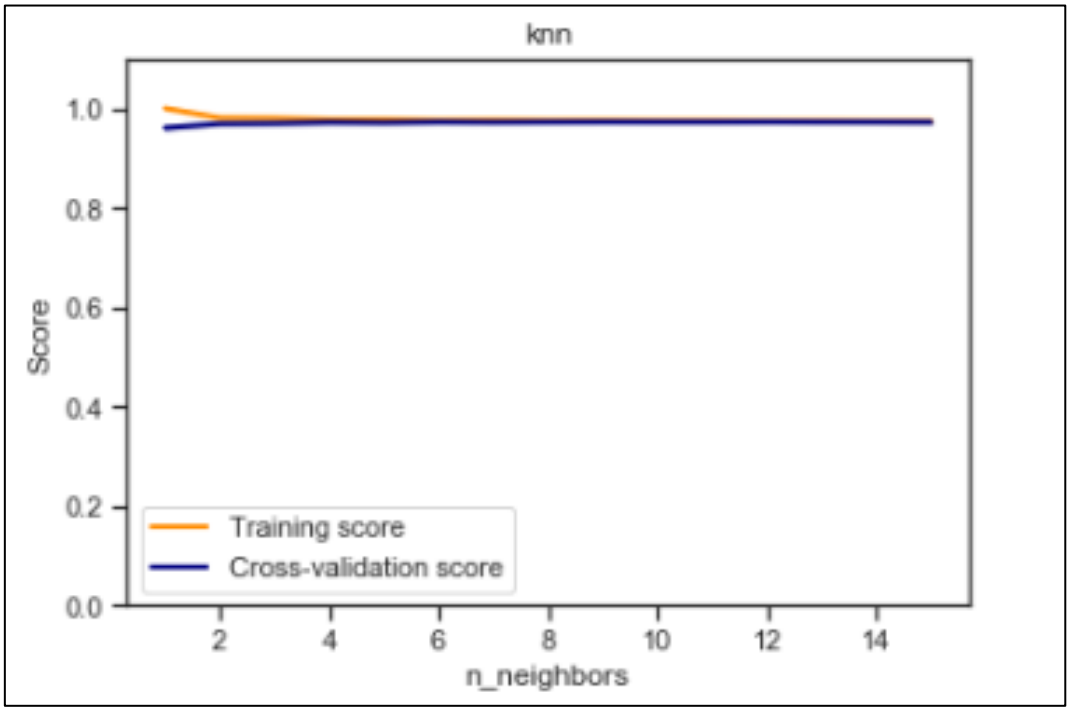



Рис.8 График точности обучения и теста

**Итоговая комплексная метрика качества:**

f1_score(PULSAR_y_test, target2_1)

0.8461538461538461

# ЗАКЛЮЧЕНИЕ

В итоге решили поставленную задачу и сделали модель с точностью более 80% для класиификации звезды является она пульсаром или нет

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


- https://habr.com/ru/company/ods/blog/322534/#metod-blizhayshih-sosedey
- https://nbviewer.jupyter.org/github/ugapanyuk/ml_course/blob/master/common/notebooks/metrics/metrics.ipynb
- https://www.kaggle.com/pavanraj159/predicting-a-pulsar-star
- https://ru.wikipedia.org/wiki/%D0%9F%D1%83%D0%BB%D1%8C%D1%81%D0%B0%D1%80
- https://habr.com/ru/company/ods/blog/322534/#metod-blizhayshih-sosedey