# Метод k взвешенных ближайших соседей

Казаринов Андрей

316 группа, кафедра МС, ВМК МГУ

# Задача классификации

$X = \mathbb{R}^n$ — множество объектов

$Y = \{1, \ \ldots, \ M\}$ — множество ответов,

$X^l = \{(x_i, y_i), \ \mathrm{i} = \overline{1, n}\}$ — обучающая выборка, где $y_i = y(x_i)$,

y — неизвестная дискретнозначная функция.

Требуется найти решающую функцию $a$: $X \ \rightarrow Y$,

приближающую $y$ на всем множестве $X$.

# Математическое обоснование

**Гипотеза о компактности:**
"Близкие" объекты, как правило, лежат в одном классе. Понятие "близости" формализуется метрикой.

**Евклидова метрика:**

$$\rho(x_1, x_2) = \left( \sum_{j=1}^{n} |x_1^j - x_2^j|^2 \right)^{\frac{1}{2}}.$$

**Метрика Минковского:**

$$\rho(x_1, x_2) = \left( \sum_{j=1}^{n} |x_1^j - x_2^j|^p \right)^{\frac{1}{p}}.$$

**Косинусная метрика:**

$$\rho(x_1, x_2) = \frac{(x_1,\ x_2)}{||x_1||_2 * ||x_2||_2}.$$

$x_i = (x_i^1, \dots, x_i^n) -$ вектор признаков объекта $x_i,\ i = 1,2.$

# Метод k ближайших соседей

Для объектов выборки $x_1, \dots, x_l$ введём новую нумерацию $x^{(1)}, \dots, x^{(l)}$ в порядке возрастания их расстояния от объекта $x$:

$$\rho\left(x, x^{(1)}\right) \leqslant \rho\left(x, x^{(2)}\right) \leqslant \cdots \leqslant \rho\left(x, x^{(l)}\right).$$

Всё готово для определения классификатора $a$. Задаём его следующей формулой:

$$a(x; X^l) = \arg\max_{y \in Y} \sum_{i=1}^{l} \mathbb{1}_{\{y^{(i)}=y\}} w(i, x),$$

$w(i, x)$ — весовая функция (вес) объекта $x^{(i)}$.
Для метода k ближайших соседей: $w(i, x) = \mathbb{1}_{\{i \leqslant k\}}$.

# Способы введения весов для соседей

1 $w_i$ - вес, зависящий от номера i

$$w(i,x) = \mathbb{1}_{\{i \leqslant k\}} w_i$$

а) линейно убывающие веса

$$w_i = \frac{k+1-i}{k};$$

б) экспоненциально убывающие веса

$$w_i = q^i, \quad 0 < q < 1;$$

2 Вес равный расстоянию

$$w(i,x) = \rho\left(x, x^{(i)}\right);$$

3 Ядро ширины h

$$w(i,x) = K\left(\frac{\rho\left(x, x^{(i)}\right)}{h}\right).$$

# Постановка задачи

В качестве объектов взяты фильмы. Классификация производится на $2$ класса по жанру. Она определяет, относится фильм к выбранному классу или нет. Например: драматический фильм или не драматический, комедийный или не комедийный. Фильм будем считать не принадлежащим жанру $g$, если в датасете в столбце жанр у этого фильма нет жанра $g$.

# Описание данных

В качестве набора фильмов был взят открытый датасет с портала Kaggle из 85855 фильмов IMDb с такими атрибутами как название, описание, жанр, количество оценок, средняя оценка и т.д. Для создания меток ответов обучающей выборки будут использоваться жанры фильмов, а для признаков объектов – текстовые столбцы фильмов (название, описание, режиссёр, актёры, название киностудии…)

| | original_title | description | actors | director | writer | production_company | genre |
|---|---|---|---|---|---|---|---|
| 0 | Miss Jerry | The adventures of a female reporter in the 1890s. | Blanche Bayliss, William Courtenay, Chauncey D... | Alexander Black | Alexander Black | Alexander Black Photoplays | Romance |
| 1 | The Story of the Kelly Gang | True story of notorious Australian outlaw Ned ... | Elizabeth Tait, John Tait, Norman Campbell, Be... | Charles Tait | Charles Tait | J. and N. Tait | Biography, Crime, Drama |
| 2 | Den sorte drøm | Two men of high rank are both wooing the beaut... | Asta Nielsen, Valdemar Psilander, Gunnar Helse... | Urban Gad | Urban Gad, Gebhard Schätzler-Perasini | Fotorama | Drama |
| 3 | Cleopatra | The fabled queen of Egypt's affair with Roman ... | Helen Gardner, Pearl Sindelar, Miss Fielding, ... | Charles L. Gaskill | Victorien Sardou | Helen Gardner Picture Players | Drama, History |
| 4 | L'Inferno | Loosely adapted from Dante's Divine Comedy and... | Salvatore Papa, Arturo Pirovano, Giuseppe de L... | Francesco Bertolini, Adolfo Padovan | Dante Alighieri | Milano Film | Adventure, Drama, Fantasy |

# Выбор параметров

- Векторизация текста производилась методом **TfidfVectorizer**. Tf означает частоту термина, а tf-idf означает частоту термина, умноженную на обратную частоту документа. Такая схема взвешивания терминов позволяет хорошо решать задачу классификации документов. Если термин встречается в большом числе описаний фильмов, то он менее информативен для определения жанра фильма и, наоборот.
- В качестве метрики близости выбираем **косинусную**. Косинусная близость лучше близости по евклидовой метрике, потому что в нашей задаче сонаправленность векторов встречаемости токенов важнее чем разность их величин.

```
tf_idf = TfidfVectorizer(max_df=0.8, min_df=10, stop_words='english')
```

# Описание алгоритма

Разбиение выборки на $75\%$ обучающую и $25\%$ тестовую.

**Этап 0**:

Выбираем какой вес будем использовать. Обучение проводится методом кросс валидации на разном количестве соседей. Кросс валидация на $4$ блоках из обучающей выборки. Затем смотрим при каком весе метрика качества оказалась выше и фиксируем этот вес.

**Этап 1**:

Смотрим как меняется качество метрики от количества соседей с уже выбранным нами весом. Количество соседей изменяем от $1$ до $1000$. Фиксируем количество соседей, для которого лучшее значение метрики.

**Этап 2** (финальный):

Смотрим какое качество показывает обученный классификатор на тестовой выборке.

# Метрики качества

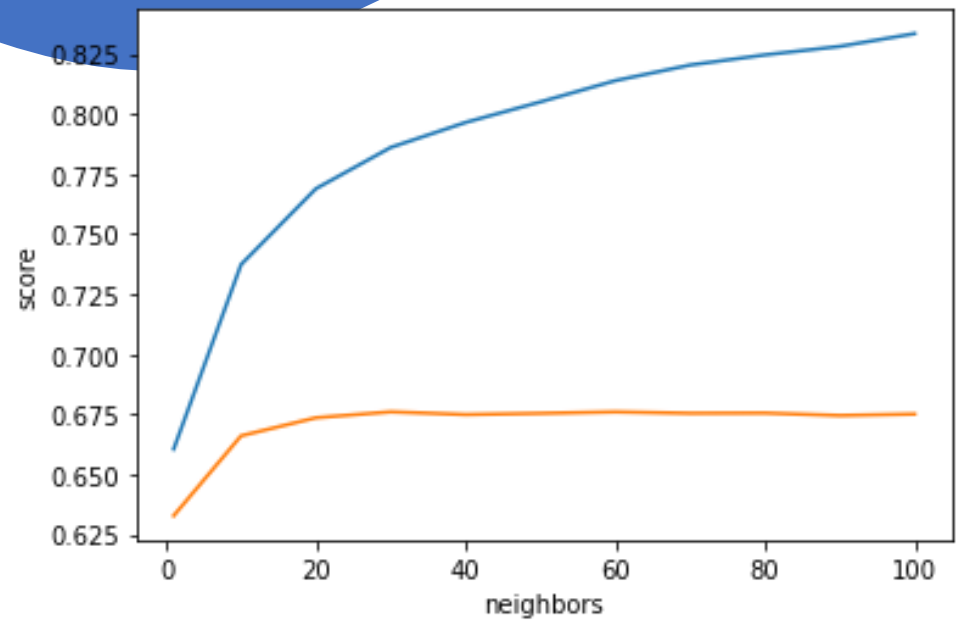
Для оценки качества будем в основном использовать полноту (recall), а также посмотрим на accuracy. Почему recall: в данной задаче false positive не всегда является ошибкой - фильм может иметь жанр драмы как неосновной, но это не указано в столбце жанры, а случай false negative грубая ошибка – в датасете помечено, что фильм драма, но к драматичным алгоритм его не отнёс. Лучше всего такую особенность отражает метрика качества recall.

# Полученные результаты. Драма.

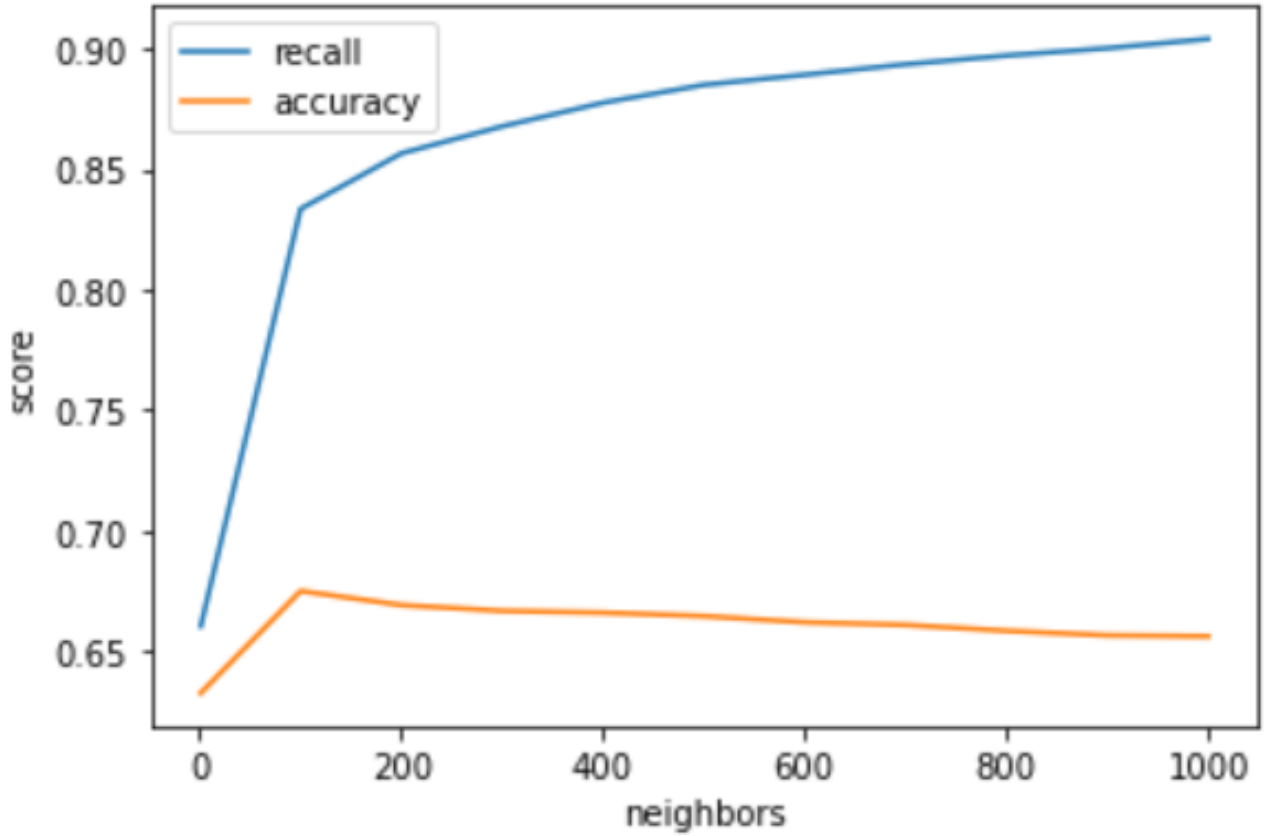
## Этап 0

```
{(1, 'uniform'): 0.6605275499731177,
 (1, 'linear'): 0.6605275499731177,
 (1, 'my_exp'): 0.6605275499731177,
 (1, 'distance'): 0.6605275499731177,
 (5, 'uniform'): 0.7117852697398503,
 (5, 'linear'): 0.7013384940690495,
 (5, 'my_exp'): 0.6605275499731177,
 (5, 'distance'): 0.7119996149410621,
 (10, 'uniform'): 0.6583977193951804,
 (10, 'linear'): 0.7314971866261734,
 (10, 'my_exp'): 0.6605275499731177,
 (10, 'distance'): 0.737352771252132}
```

# Полученные результаты. Драма.




Этап 1



# Полученные результаты. Драма.

Этап 2

```
neigh = neighbors.KNeighborsClassifier(n_neighbors=best_par[0],
                                       metric='cosine', weights=best_par[1])
scaler = TfidfVectorizer(max_df=0.8, min_df=10, stop_words='english')
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
neigh.fit(X_train, y_train)
score_test1_acc = accuracy_score(y_test, neigh.predict(X_test))
score_test1_rec = recall_score(y_test, neigh.predict(X_test))
score_test1_acc, score_test1_rec
```

✓ 1m 26.3s

```
(0.6835068381302306, 0.8350945494994438)
```

Время

Этап 0: ~50 минут
Этап 1: ~2.5 часа
Этап 2: 1 мин 26 сек

# Пример. Драма.

```
neighbors.KNeighborsClassifier(n_neighbors=best_par[0], metric='cosine', weights=best_par[1])
```

✓ 0.5s

```
KNeighborsClassifier(metric='cosine', n_neighbors=100, weights='distance')
```

| | original_title | genre | drama |
|---|---|---|---|
| 15528 | The Godfather | Crime, Drama | 1.0 |
| 34127 | The Lord of the Rings: The Return of the King | Action, Adventure, Drama | 1.0 |
| 31279 | The Lord of the Rings: The Fellowship of the Ring | Action, Adventure, Drama | 1.0 |
| 32229 | The Matrix | Action, Sci-Fi | 1.0 |
| 28066 | Forrest Gump | Drama, Romance | 1.0 |
| 28381 | Pulp Fiction | Crime, Drama | 1.0 |
| 32487 | Fight Club | Drama | 1.0 |
| 57475 | Inception | Action, Adventure, Sci-Fi | 0.0 |
| 48078 | The Dark Knight | Action, Crime, Drama | 0.0 |
| 28453 | The Shawshank Redemption | Drama | 1.0 |