*Инструментарий машинного обучения и анализа данных*

**“Распознавание рукописных цифр”**

*Описание:* Работа направлена на разработку инструмента для проверки на ботов в соц. сетях с помощью методов машинного обучения путем сравнения исходного изображения с цифрами и цифрами, введенными пользователем.

**1.Словесное описание набора данных.**

Наборы данных содержат в себе черно-белые изображения рукописных цифр от нуля до девяти.

Каждое изображение имеет размерность 28 на 28 пикселей, что в целом дает 784 пикселя. Каждый пиксель имеет по одному связанному с ним значению, которое обозначает оттенок пикселя: светлый или темный (часть цифры обозначают темные пиксели). Значение пикселя - целое число от 0 до 255.

Таблица тренировочных данных имеет 785 колонок. Первая колонка - это сама цифра, которая изображена на картинке. Последующие 784 колонки содержат значения пикселей изображения.

Каждый пиксель колонки тренировочной таблицы имеет имя наподобие pixelX, где X - целое число между 0 и 783. Для сопоставления значения пикселя с конкретным пикселем на изображении используется формула x = i \* 28 + j, где i и j - целые числа в диапазоне от 0 до 27, обозначающие высоту и ширину изображения, соответственно.

**2. Словесное описание преобразований над данными и их обоснование.**

Входные данные представлены в удобном для выполнения над ними предсказывающих алгоритмов виде и не требуют преобразований.

Значения пикселей тестовых изображений уже распределены по таблице, каждый пиксель в своей ячейке, соответственно. Это позволяет удобно работать со всеми входными данными и способствует обеспечению оптимальной скорости работы алгоритмов.

**3. Словесное описание применяемых методов машинного обучения.**

Для реализации программного кода приложения и, в частности, алгоритмов машинного обучения, планируется использовать классификаторы

* **Naive Bayes**
* **Random Forest**

По окончанию реализации для обоих методов - Naive Bayes и Random Forest - будут произведены тестовые запуски программы на различных входных данных с целью выявления наиболее производительного и точного алгоритма для данной задачи - распознавания рукописных цифр.

Для данной задачи нам необходимо классифицировать каждый новый набор входных данных (картинку, представленную в виде набора пикселей) - определить его принадлежность к одному из 10 классов, отображающих цифры от 0 до 9 соответственно.

**Naive Bayes** — простой вероятностный [классификатор](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8), основанный на применении [Теоремы Байеса](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D0%91%D0%B0%D0%B9%D0%B5%D1%81%D0%B0) со строгими (наивными) предположениями о [независимости](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D1%8C).

Достоинством наивного байесовского классификатора является малое количество данных для обучения, необходимых для оценки параметров, требуемых для классификации.

Её смысл на обывательском уровне можно выразить следующим образом. Теорема Байеса позволяет переставить местами причину и следствие. Зная с какой вероятностью причина приводит к некоему событию, эта теорема позволяет рассчитать вероятность того что именно эта причина привела к наблюдаемому событию.

Цель классификации состоит в том чтобы понять к какому классу принадлежит изображение, поэтому нам нужна не сама вероятность, а наиболее вероятный класс. Байесовский классификатор использует [оценку апостериорного максимума](http://ru.wikipedia.org/wiki/%D0%9E%D1%86%D0%B5%D0%BD%D0%BA%D0%B0_%D0%B0%D0%BF%D0%BE%D1%81%D1%82%D0%B5%D1%80%D0%B8%D0%BE%D1%80%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BC%D0%B0%D0%BA%D1%81%D0%B8%D0%BC%D1%83%D0%BC%D0%B0) (Maximum a posteriori estimation) для определения наиболее вероятного класса. Грубо говоря, это класс с максимальной вероятностью. Следовательно, нам надо рассчитать вероятность для всех классов и выбрать тот класс, который обладает максимальной вероятностью.

**Random Forest** - для полученных данных алгоритм создает множество деревьев принятия решений и потом усредняет результат их предсказаний. Важным моментом тут является элемент случайности в создании каждого дерева. Ведь понятно, что если мы создадим много одинаковых деревьев, то результат их усреднения будет обладать точностью одного дерева.Предположим, у нас есть некие данные на входе. Каждая колонка соответствует некоторому параметру, каждая строка соответствует некоторому элементу данных. Мы можем выбрать, случайным образом, из всего набора данных некоторое количество столбцов и строк и построить по ним дерево принятия решений. Дальше мы можем повторить эту процедуру много-много раз и получить множество различных деревьев. Алгоритм построения дерева очень быстр. И поэтому нам не составит большого труда сделать столько деревьев, сколько будет нужно. При этом, все эти деревья в некотором смысле случайны, ведь для создания каждого из них мы выбирали случайное подмножество данных.

**4. Описание оценки качества Вашего алгоритма.**

Алгоритм будет проверяться с помощью алгоритма ROC-curves, который позволяет оценить качество [бинарной классификации](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%B8%D0%BD%D0%B0%D1%80%D0%BD%D0%B0%D1%8F_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F&action=edit&redlink=1), отображает соотношение между долей объектов от общего количества носителей признака, верно классифицированных, как несущих признак и долей объектов от общего количества объектов, не несущих признака, ошибочно классифицированных, как несущих признак при варьировании порога решающего правила.

Также алгоритм известен как кривая ошибок. Анализ классификаций с применением ROC-кривых называется ROC-анализом.

Количественную интерпретацию ROC дает показатель AUC ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) area under ROC curve, площадь под ROC-кривой) — площадь, ограниченная ROC-кривой и осью доли ложных положительных классификаций. Чем выше показатель AUC, тем качественнее классификатор, при этом значение 0,5 демонстрирует непригодность выбранного метода классификации (соответствует случайному гаданию). Значение менее 0,5 говорит, что классификатор действует с точностью до наоборот: если положительные назвать отрицательными и наоборот, классификатор будет работать лучше.

В задаче проверки принадлежности картинки с изображением рукописной цифры к одному из классов от 0 до 9, соответствующих цифрам, будет использован следующий подход проверки качества алгоритма.  
После каждого нового получения результатов работы алгоритма на одной из предложенных картинок, будет производиться проверка полученных результатов с помощью алгоритма ROC-curves: для каждого из 10 классов будет производиться проверка качества бинарной классификации - принадлежит изображение к данному классу или нет.

**5. Вывод из работы.**

В результате выполнения всех этапов работ проекта, будет получена утилита, позволяющая определить какая цифра изображена на картинке.

Это станет основой будущего сервиса по проверке пользователей соц. сетей на ботов.

Пользователю будет предлагаться ввести число, изображенное на картинке. Далее с помощью разработанной утилиты сервер будет также определять число, изображенное на картинке и сравнивать с числом, введенным пользователем. При совпадении будет возвращен ответ “не бот”, в противном случае - “бот”.