В классе мы обсуждали анализ главных компонент (Principal Component Analysis) и его применение в качестве feature extractor'a.

Мы использовали PCA+LDA на задаче `mnist` и получили сходу неплохой результат в 3.6%.

Пользуясь случаем, напоминаю про дедлайн заданий `Advertising` и `Universities` в пятницу 31 октября

1 Домашнее задание

Задание 1.1 (Задание (теоретическое))**.** Вспомнить (или прочитать) про SVD-разложение и понять его связь с PCA (почему SVD дает нам решение).

Вспомнить про значение слов робастность и эффективность и как они связаны. Каки робастные оценки Вы знаете?

Задание 1.2. Для данных `mnist` попробовать применить известные вам подходы и улучшить мой результат. Сами данные можно взять здесь, а тут можно взять код для загрузки данных в `R` и визуализации.

Обратите внимание, что данные уже поделены на обучающее и тестовое подмножество. Тестовое подмножество используется только для проверки (правильнее было бы назвать его валидационным), все обучающие действия, включая кросс-валидацию, делать нужно только с обучающим подмножеством.

Напоминаю также, что наша цель не помериться количеством слоев нейронной сети и числом нодов кластера. Не стоит сразу пытаться использовать сложные методы, лучше сконцентрироваться на грамотном применении простых и правильной подготовке данных.

Какие идеи можно попробовать:

1. удалить предикторы с нулевой вариацией, чтобы считалось побыстрее и варнингов было поменьше

2. попробовать разные методы, использовать возможности отбора признаков (типа `stepAIC()`).

3. deskewing (симметризация). Простейший подход — это посчитать для каждого предиктора преобразование типа $\log(x+1)$ и сравнить ассиметрии для логарифмированного и исходного признака, оставив тот, у кого асимметрия меньше. Как вариант, можно попробовать использовать другое преобразование или сразу что-то вроде Бокса-Кокса.

4. посчитать главные компоненты не для всех данных, а отдельно по каждому классу, использовать полученный набор как новые фичи.

5. каким-то образом эксплуатировать пространственную структуру предикторов (у нас ведь не просто какие-то абстрактные данные, а точки на картинке). Можно использовать методы обработки изображений, например, сжать картинки, обрезать края, посчитать какие-то характеристики, фильтры, и т.п.

6. рассмотреть наиболее проблемно разделимые классы попарно и попробовать сдвинуть границу принятия решения с помощью ROC-кривых.

7. использовать попарную классификацию с голосованием в конце