|  |  |
| --- | --- |
|  | ***Министерство науки и высшего образования Российской Федерации***  *Калужский филиал федерального государственного бюджетного образовательного учреждения высшего образования*  ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***Национальный исследовательский университет***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

Кириллов В.Ю.

**Инструментальные средства, подготовка эксперимента и анализ результатов**

**Методические указания   
по дисциплине  
 «Проектирование интеллектуальных систем»**

Калуга, 2022 г.

**Тема:**

Инструментальные средства, подготовка эксперимента и анализ результатов

**Цель:**

формирование практических навыков подготовки и настройки инструментальной среды анализа данных, умений подготовить эксперимент, представить и проанализировать полученные данные.

**Задачи**

* Подготовка и настройка среды исполнения Python
* Подготовка и предобработка данных эксперимента
* Простейшие алгоритмы анализа данных (получение статистических характеристик)
* Графическое представление результатов
* Написание отчета о работе

**Введение**

В настоящем курсе не ставится задача изучения языка программирования Python в чистом виде, цель - применять его в качестве эффективного инструмента для решения задач, требующихся для анализа данных при построении интеллектуальных систем. Применение Python для решения этих задач – один из современных трендов (не единственный, есть и R, и скрипты Azure ML, и другие инструментальные и визуальные средства, безусловно, заслуживающие внимания). Python имеет:

* изящный и краткий синтаксис;
* кроссплатформенную поддержку;
* обширную коллекцию мощных библиотек (IPython, NumPy, Pandas, Matplotlib, Scikit-Learn и др.);
* современные средства разработки.

Python используется на всех этапах рабочего процесса, обычно применяемых в моделировании машинного обучения:

* прием и обработка данных;
* создание компонентов;
* обучение моделей;
* проверка моделей;
* развертывание моделей.

Знакомство с языком Python, включая описание функций, присваивание переменных, вызов методов объектов, управление потоком выполнения программы и решение других простейших задач не входит в задачу лабораторной работы. Предполагается, что студенты, не знакомые с языком способны освоить его на должном уровне самостоятельно в процессе выполнения работ курса. Работа должна помочь пользователям языка Python научиться применять стек инструментов исследования данных языка Python для эффективной подготовки, хранения, манипуляции и понимания данных и результатов.

**Подготовка и настройка среды исполнения Python**

Возможные варианты

1. Выбор операционной системы не имеет принципиального значения
2. Установка Python (рекомендуется версия не ниже 3.5) и библиотек по необходимости при помощи pip
3. Установка сборки Anaconda. Преимущества: большинство библиотек уже входят в состав и актуально. Возможные дополнения Xgboost, Vowpal Wabbit по необходимости устанавливаются с помощью conda
4. Использование Azure ML
5. Использование Google Colab

**Подготовка и предобработка данных эксперимента**

Pandas — это библиотека Python, предоставляющая широкие возможности для анализа данных. Данные, с которыми работают, часто хранятся в форме табличек — например, в форматах .csv, .tsv или .xlsx. С помощью библиотеки Pandas такие табличные данные очень удобно загружать, обрабатывать и анализировать с помощью SQL-подобных запросов. А в связке с библиотеками Matplotlib и Seaborn Pandas предоставляет широкие возможности визуального анализа табличных данных.

Основными структурами данных в Pandas являются классы Series и DataFrame. Первый из них представляет собой одномерный индексированный массив данных некоторого фиксированного типа. Второй – это двухмерная структура данных, представляющая собой таблицу, каждый столбец которой содержит данные одного типа. Можно представлять её как словарь объектов типа Series. Структура DataFrame отлично подходит для представления реальных данных: строки соответствуют признаковым описаниям отдельных объектов, а столбцы соответствуют признакам.

Весьма часто для работы с данными необходима их предварительная обработка. Средства Python отлично подходят для этого, позволяя лаконично описать необходимые операции. Кроме того, весьма полезным является навых «пристрелки» к данным – интерактивный быстрый анализ вариантов, «что будет, если …»

**Простейшие алгоритмы анализа данных (получение статистических характеристик).**

Прежде всего имеет смысл научиться, пользуясь пакетом библиотек оценивать основные статистические характеристики данных: математическое ожидание, дисперсию, корреляцию признаков и др., а также изучить возможности по разнообразному графическому представлению результатов, предоставляемому средой.

**Краткая характеристика объекта**

Исследуются наборы данных по оттоку клиентов телеком-оператора (18 признаков), демографические данные (15 признаков), анализ данных о публикациях (15 признаков)

**Литература**:

1. Открытый курс машинного обучения. Тема 1. Первичный анализ данных с Pandas - <https://habr.com/company/ods/blog/322626/>
2. Открытый курс машинного обучения. Тема 2: Визуализация данных c Python - <https://habr.com/company/ods/blog/323210/>

**Заготовки ноутбуков и домашних заданий:**

[**https://github.com/Yorko/mlcourse.ai/tree/main/jupyter\_russian**](https://github.com/Yorko/mlcourse.ai/tree/main/jupyter_russian)

**Данные для ноутбуков и домашних заданий:**

[**https://github.com/Yorko/mlcourse.ai/tree/main/data**](https://github.com/Yorko/mlcourse.ai/tree/main/data)

(можно скачать весь репозиторий – там все будет. Все задания размещаются в <https://github.com/Yorko/mlcourse.ai/tree/main/jupyter_russian/assignments_demo>

**Порядок выполнения работы**

1. Подготовить среду исполнения Python, включающую IPython, Pandas, Seaborn, Plotly.
2. Подготовить Jupiter Notebook для отчета по лабораторной работе, убедиться в его работоспособности и экспортируемости.
3. Выполнить загрузку данных для экспериментов. При необходимости выполнить фильтрацию признаков и/или нормализацию их значений. Выполнить просмотр head данных.
4. Получить статистические характеристики распределений признаков.
5. Получить графические отображения результатов, сопоставив наиболее удачные. Допустимо и даже похвально изменять предложенные в ноутбуках действия и делать самостоятельные выводы.
6. Обдумать и выполнить самостоятельные задания:  
   <https://github.com/Yorko/mlcourse.ai/blob/main/jupyter_russian/assignments_demo/assignment01_adult_pandas.ipynb>

и

<https://github.com/Yorko/mlcourse.ai/blob/main/jupyter_russian/assignments_demo/assignment02_habr_visual_analysis.ipynb>

(если Вам не нравится титаник, американская статистика и визуальный анализ публикаций - можете рассмотреть аналогичные задания на наборе данных из [Дополнение 2019](#bookmark=id.9ecnue27mekw)) и дописать вывод в Notebook.

1. Полученный результат и является отчетом. Отчет должен включать в себя описание выполненных Вами нетривиальных действий, возникших проблем; и результатов, которые иллюстрируют Ваши успехи и проблемы.

Отчеты сдаются исключительно в электронном виде ЭОС.

**Вопросы для самоконтроля**

*По демографическим данным*

1. Какова доля граждан Германии (признак native-country)?
2. Каковы средние значения и среднеквадратичные отклонения возраста тех, кто получает более 50K в год (признак salary) и тех, кто получает менее 50K в год?
3. Правда ли, что люди, которые получают больше 50k, имеют как минимум высшее образование? (признак education – Bachelors, Prof-school, Assoc-acdm, Assoc-voc, Masters или Doctorate)

*По анализу публикаций*

1. Когда лучше всего публиковать статью? Как это доказать?
2. Правда ли, что по субботам авторы пишут в основном днём, а по понедельникам — в основном вечером?

**Дополнение 2019.**

Возможно, данные и задачи самостоятельного задания покажутся Вам слишком скучными. Вот вам настоящие данные по результатам ЕГЭ 2019 года в одной из областей ([результаты](https://drive.google.com/open?id=13PlAXlgfJo333pULO0lG4rX7cbTj1tj2) и [коды предметов](https://drive.google.com/open?id=1grv0b_0QwCTDrm-o6FeTk3F0wcsyzGe8)) и настоящее жизненное задание:

(*на жаргоне министерства ОО - это образовательное учреждение)*

***Ранжирование ОО по интегральным показателям качества подготовки выпускников***

*(анализируется доля выпускников текущего года, набравших соответствующее количество тестовых баллов, полученных на ЕГЭ по трём предметам, кроме математики базового уровня)*

*Таблица 2*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *№ п/п* | *Наименование ОО* | *ВТГ, получившие суммарно по трём предметам соответствующее количество тестовых баллов* | | | | | | | |
| *до 160* | | *от 161 до 220* | | *от 221 до 250* | | *от 251 до 300* | |
| *чел.* | *%[1]* | *чел.* | *%* | *чел.* | *%* | *чел.* | *%* |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

***3.4. Выделение перечня ОО, продемонстрировавших наиболее высокие результаты ЕГЭ по предмету:*** *выбирается от 5 до 15% от общего числа ОО в субъекте РФ, в которых*

*o доля участников ЕГЭ,* ***получивших от 81 до 100 баллов,*** *имеет* ***максимальные значения*** *(по сравнению с другими ОО субъекта РФ);*

*Примечание: при необходимости по отдельным предметам можно сравнивать и доли участников, получивших от 61 до 80 баллов.*

*o доля участников ЕГЭ,* ***не достигших******минимального балла****, имеет* ***минимальные значения*** *(по сравнению с другими ОО субъекта РФ)*

*Примечание. Сравнение результатов по ОО проводится при условии не менее 10 количества участников ОО.*

*Таблица 13*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *№* | *Наименование ОО* | *Доля участников, получивших*  *от 81 до 100 баллов* | *Доля участников, получивших*  *от 61 до 80 баллов* | *Доля участников,*  *не достигших минимального балла* |
|  |  |  |  |  |
|  |  |  |  |  |

***3.5. Выделение перечня ОО, продемонстрировавших низкие результаты ЕГЭ по предмету:*** *выбирается от 5 до15% от общего числа ОО в субъекте РФ, в которых*

*o доля участников ЕГЭ,* ***не достигших минимального балла****, имеет* ***максимальные значения*** *(по сравнению с другими ОО субъекта РФ);*

*o доля участников ЕГЭ,* ***получивших от 61 до 100 баллов****, имеет* ***минимальные значения*** *(по сравнению с другими ОО субъекта РФ).*

*Примечание. Сравнение результатов по ОО проводится при условии не менее 10 количества участников ОО.*

*Таблица 14*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *№* | *Наименование ОО* | *Доля участников,*  *не достигших минимального балла* | *Доля участников, получивших*  *от 61 до 80 баллов* | *Доля участников, получивших*  *от 81 до 100 баллов* |
|  |  |  |  |  |
|  |  |  |  |  |

*[1] от количества ВТГ данной ОО*

***2.3.2. Статистический анализ выполняемости заданий и групп заданий КИМ ОГЭ в 2019 году***

*Для заполнения таблицы 12 используется обобщенный план КИМ по предмету с указанием средних процентов выполнения по каждой линии заданий в регионе*

*Таблица 12*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Обознач.*  *задания в работе* | *Проверяемые элементы содержания / умения* | *Уровень сложности задания* | *Средний процент выполнения[1]* | *Процент*  *выполнения по региону в группах,*  *получивших отметку* | | | |
| *«2»* | *«3»* | *«4»* | *«5»* |
| *…* |  |  |  |  |  |  |  |

*[1] Для политомических заданий (максимальный первичный балл за выполнение которых превышает 1 балл), средний процент выполнения задания вычисляется как сумма первичных баллов, полученных всеми участниками, выполнявшими данное задание, отнесенная к количеству этих участников\*на максимальный балл, который можно получить за выполнение задания.*

*рекомендуется использовать формулу p = N/(n\*m)\*100, где N – сумма первичных баллов, полученных всеми участниками группы за выполнение задания, n – количество участников в группе, m – максимальный первичный балл, который можно получить за выполнение задания.*

**Классификация, деревья решений и метод ближайших соседей**

**Тема:**

Классификация, деревья решений и метод ближайших соседей

**Цель:**

формирование практических навыков применения простейших методов классификации и регрессии

**Задачи**

* разработать решение задачи классификации методом решающих деревьев
* разработать решение задачи классификации методом ближайших соседей.
* обосновать выбор правил и метрик.
* выполнить задачу кросс-валидации.
* рассмотреть решение задачи регрессии на полученных моделях

**Введение**

В настоящей работы исследуются простейшие алгоритмы классификации и регрессии.

**Классификация** – отнесение объекта к одной из категорий на основании его признаков.

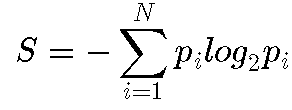
**Регрессия** – прогнозирование количественного признака объекта на основании прочих его признаков.

**Деревья решений (decision trees)**

Дерево решений в алгоритмах машинного обучения – это объединение логических правил вида "Значение признака a меньше x И Значение признака b меньше y ... => Класс 1" в древовидную структуру данных, где логические правила представляются внутренними вершинами дерева, а выводы-классификаторы являются листьями. Огромное преимущество деревьев решений в том, что они легко интерпретируемы, понятны человеку. Классификация может быть объяснена однозначно интерпретируемым проходом по дереву от листа к вершине. В связи с этой "понятностью" деревьев решений и их сходством с моделью принятия решений человеком (можно легко объяснять свою модель), деревья решений получили огромную популярность.

Критерием эффективности построения дерева решений для конкретной предметной области является минимизация его информационной емкости, или, выражаясь более строго: энтропии.

**Энтропия Шеннона** определяется для системы с N возможными состояниями следующим образом:



где *pi* - вероятности нахождения системы в *i*-ом состоянии. Состоянием могут являться, как фактические состояния системы, так и искусственно агреггированные (например, диапазоны значений).

Следует отметить, что понятие предметной области в данном случае применимо не к области прогнозов модели (ее свойства нам априорно неизвестны), а лишь к обучающей выборке, имеющейся в нашем распоряжении. Ее статистические свойства, в частности, энтропию мы в состоянии исследовать и на этом основании построить «эффективную» в узком смысле этого слова модель.

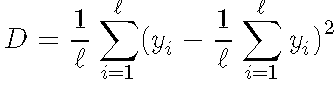
В основе популярных алгоритмов построения дерева решений, таких как ID3 и C4.5, лежит принцип жадной максимизации прироста информации – на каждом шаге выбирается тот признак, при разделении по которому прирост информации оказывается наибольшим. Дальше процедура повторяется рекурсивно, пока энтропия не окажется равной нулю или какой-то малой величине. В разных алгоритмах применяются разные эвристики для "ранней остановки" или "отсечения", чтобы избежать построения переобученного дерева.

Альтернативным критерием качества построения разбиений в дереве является максимизация **неопределенности Джини**:



В работе будет предложено сравнить качество обучения модели на базе обоих критериев.

Для задачи регрессии критерий качества может быть приведен к задаче минимизации дисперсии аппроксимирующей функции *y*:



**Метод ближайших соседей (k Nearest Neighbors, или kNN)**

Основой метода ближайших соседей является гипотеза компактности: если метрика расстояния между примерами введена правильно, то схожие (ближкие по метрике) примеры гораздо чаще лежат в одном классе, чем в разных.

Для классификации каждого из объектов тестовой выборки необходимо последовательно выполнить следующие операции:

* Вычислить расстояние до каждого из объектов обучающей выборки
* Отобрать k объектов обучающей выборки, расстояние до которых минимально
* Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей

Таким образом, единственной творческой задачей при использовании этого метода является определение правильной метрики, что зачастую является нетривиальной задачей.

**Выбор параметров модели и кросс-валидация**

Главная задача обучаемых алгоритмов – их способность обобщаться, то есть хорошо работать на новых данных. Поэтому, для оценки качества построенной модели часть обучающих данных используется для проверки качества. Чаще всего это делается одним из 2 способов:

* отложенная или тестовая выборка (held-out/hold-out set). При таком подходе мы оставляем какую-то долю обучающей выборки (как правило от 20% до 40%), обучаем модель на остальных данных (60-80% исходной выборки) и считаем некоторую метрику качества модели (например, долю правильных ответов в задаче классификации) на отложенной выборке.
* кросс-валидация (cross-validation, на русский еще переводят как скользящий или перекрестный контроль). В частности K-fold кросс-валидация, где модель обучается K раз на разных (K-1) подвыборках исходной выборки, а проверяется на одной подвыборке (каждый раз на разной). Получаются K оценок качества модели, которые обычно усредняются, выдавая среднюю оценку качества классификации/регресии на кросс-валидации.

Кросс-валидация дает лучшую по сравнению отложенной выборкой оценку качества модели на новых данных. Но кросс-валидация вычислительно дорогостоящая, если данных много.

**Краткая характеристика объекта**

Исследуются следующие наборы данных:

Искусственные классы «точки на плоскости», «невозврат кредитов», «непрерывная функция с шумом»

Датасеты по оттоку клиентов телеком-оператора (18 признаков), данные по рукописным цифрам MNIST (8x8), Kaggle "Titanic: Machine Learning from Disaster" (12 признаков)

**Литература**:

1. Открытый курс машинного обучения. Тема 3. Задача классификации, дерево решений и метод ближайших соседей - <http://nbviewer.jupyter.org/github/Yorko/mlcourse.ai/blob/master/jupyter_russian/topic03_decision_trees_knn/topic3_trees_knn.ipynb>
2. Открытый курс машинного обучения. Тема 3. Обучение с учителем. Методы классификации. Практика - <http://nbviewer.jupyter.org/github/Yorko/mlcourse.ai/blob/master/jupyter_russian/topic03_decision_trees_knn/lesson3_practice_trees_titanic.ipynb>

**Порядок выполнения работы**

1. Подготовить среду исполнения Pythonс необходимыми библиотеками.
2. Загрузить Jupiter Notebook из раздела «Литература»
3. Выполнить загрузку данных для экспериментов. Выполнить просмотр head данных.
4. Выполнить действия, описанные в Jupiter Notebook.
5. Исследовать зависимость качества обучения от параметров моделей.
6. Дописать вывод в Notebook.
7. Полученный результат и является отчетом. Отчет должен включать в себя описание выполненных Вами нетривиальных действий, возникших проблем; и результатов, которые иллюстрируют Ваши успехи и проблемы.

Отчеты сдаются исключительно в электронном виде на [sapr91@mail.ru](mailto:sapr91@mail.ru). Тема письма – фамилия и номер лабораторной работы.

**Вопросы для самоконтроля**

1. Выполнить сравнения качества моделей на одинаковых данных. Сделать выводы о том, в каких ситуациях Вы предпочли бы ту или иную модель.
2. Выполнить оценку влияния параметров моделей в Scikit-learn. Определить для себя приоритеты и оптимальные пути регулировки параметров.
3. Каким образом можно установить переобученность модели? Как ее избежать?

Как на предложенных наборах данных доказать или опровергнуть тезис, что оптимальным разбиением на обучающую и тестовую выборки является 80/20?

**Линейные модели классификации и регрессии**

**Тема:**

Линейные модели классификации и регрессии

**Цель:**

формирование практических навыков применения линейных моделей для интеллектуальных систем

**Задачи**

* разработать решение задачи линейной регрессии,
* разработать решение задачи логистической регрессии с линейным классификатором.
* Выполнить оценки влияния регуляризации на качество моделей

**Введение**

В настоящей работы исследуются линейные модели для задач классификации и регрессии.

**Классификация** – отнесение объекта к одной из категорий на основании его признаков.

**Регрессия** – прогнозирование количественного признака объекта на основании прочих его признаков.

Для задачи регрессии линейная модель означает линейную зависимость целевого признака от значений признаков объекта. Для задачи классификации линейная модель предполагает разделимость пространства признаков гиперплоскостями.

**Линейная регрессия**

Строгая линейная зависимость выражается формулой *y = wx + w0,* где *y* – прогнозируемая переменная, *x* - вектор признаков, а *w* – вектор весов признаков (далее без ограничения общности можно увеличить размерность вектора признаков на 1, добавив единичное значение, которое поглотит свободный член зависимости *w0*, приведя ее к виду *y = wx*).

Учитывая стохастический характер процесса, модель наблюдений задается следующим образом:



где *y* – вектор наблюдений, *X* – расширенная матрица наблюдений, ε - случайная переменная, соответствующая случайной, непрогнозируемой ошибке модели, распределение которой по наблюдению удовлетворяет следующим ограничениям:

математическое ожидание равно 0, дисперсия конечна, корреляция равна 0.

Согласно теореме Маркова-Гаусса наилучшей среди линейных и несмещенных оценок для таким образом определенной модели будет оценка методом наименьших квадратов, определяемая как



Минимизация этой функции по вектору весов является задачей линейной регрессии. Эта задача может быть решена аналитически, решением является значение , но ввиду большой размерности матрицы наблюдений (как правило), целесообразней применение численных методов. В библиотеке SciPy для этого используется класс LinearRegression. Следует также обратить внимание, что в случае сингулярности матрицы наблюдений (а это может быть вызвано сильной корреляцией признаков) аналитическое решение найти невозможно, а численные решения будут обладать неустойчивой сходимостью. Для компенсации этого явления применяются алгоритмы *регуляризации,* которые изучаются в натоящей работе.



**Классификация методом логистической регрессии**

Основная идея линейного классификатора заключается в том, что признаковое пространство может быть разделено гиперплоскостью на два (или более для многозначных классификаторов) подпространства, в каждом из которых прогнозируется одно из значений целевого класса.

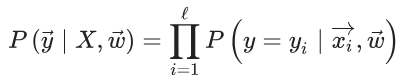
Если это можно сделать без ошибок, то обучающая выборка называется линейно разделимой. Разумеется, линейная разделимость возможна не во всех ситуациях (пример – XOR-проблема).

Предположим, что выборка линейно разделима (или почти разделима). Возникают следующие проблемы: оценка качества разбиения и выбор метода оптимизации этой оценки.

Для преобразования ступенчатой функции классификации 1/0 к непрерывному виду используется функция логистического сигмоида , которая преобразует линейную свертку вектора признаков *x* весами гиперплоскости *w* в диапазон [0; 1], что может быть интерпретировано как вероятность отнесения вектора наблюдений к одному из классов (положительному). Таким образом, задача нахождения оптимального разбиения сводится к задаче максимального правдоподобия или максимизации суммарных вероятностей безошибочной классификации наблюдений.



Вероятность правильной классификации отдельного наблюдения выражается формулой , тогда задача оптимизации сводится к нахождению максимума функции



что эквивалентно минимизации



называемой логистической функцией потерь.

Решение оптимизационной задачи осуществляется методом градиентного спуска, что возможно благодаря непрерывности и дифференцируемости целевой функции.

Так же, как и для линейной регрессии для улучшения сходимости и предотвращения локальных экстремумов применимы алгоритмы регуляризации. В данной работе предлагается исследовать реализации линейного классификатора методами SciPy, а также рассмотреть границы применимости модели – случаи, когда описательных средств линейных классификаторов недостаточно.

**Литература**:

1. Открытый курс машинного обучения. Тема 4. Линейные модели классификации и регрессии - <http://nbviewer.jupyter.org/github/Yorko/mlcourse.ai/tree/master/jupyter_russian/topic04_linear_models/>

**Порядок выполнения работы**

1. Подготовить среду исполнения Python с необходимыми библиотеками.
2. Для изучение метода линейной регрессии загрузить или синтезировать подходящий dataset (включается в отчет). Выполнить обучение модели и, по возможности, визуализацию результатов. Сделать оценки качества модели (обратите внимание, отрицательный результат также является результатом – можно сделать вывод, что линейная модель не подходит к выбранному набору данных).

Возможной (и полезной) альтернативой является выполнение заданий из Jupiter Notebook из раздела «Литература», касающихся линейной регрессии с регуляризацией: *lesson6\_practice\_linreg\_optimization.ipynb* и *lesson6\_lasso\_ridge.ipynb*.

1. Для изучения классификации методом логистической регрессии загрузить соответствующие Jupiter Notebook (*part2-5, alice)* из раздела «Литература»
2. Выполнить загрузку данных для экспериментов.
3. Выполнить действия, описанные в Jupiter Notebook.
4. Исследовать зависимость качества обучения от параметров моделей.
5. Дописать вывод в Notebook.
6. Полученный результат и является отчетом. Отчет должен включать в себя описание выполненных Вами нетривиальных действий, возникших проблем; и результатов, которые иллюстрируют Ваши успехи и проблемы.

Отчеты сдаются исключительно в электронном виде на [sapr91@mail.ru](mailto:sapr91@mail.ru). Тема письма – фамилия и номер лабораторной работы.

**Вопросы для самоконтроля**

1. Выполнить сравнения качества моделей на одинаковых данных. Сделать выводы о том, в каких ситуациях Вы предпочли бы ту или иную модель.
2. Выполнить оценку влияния параметров моделей в Scikit-learn. Определить для себя приоритеты и оптимальные пути регулировки параметров.
3. Каким образом можно установить переобученность модели? Как ее избежать?
4. Как на предложенных наборах данных доказать или опровергнуть тезис, что оптимальным разбиением на обучающую и тестовую выборки является 80/20?