Практическое занятие 1

Факультатив «Введение в анализ данных и машинное обучение на Python»

30 ноября 2019 г.

1 Общие комментарии

1. На этом занятии Вам предстоит самостоятельно пройти первые шаги работы специалиста по машинному обучению: предварительный анализ набора данных, постановка возможных задач и решение одной из них. Задания, описанные здесь, помогут Вам сделать это.

2. Работа аналитика данных очень творческая, поэтому на приведённые ниже вопросы нет единственного правильного ответа. Если ответ кажется Вам интуитивно правильным, напишите его. Если Вам кажется, что на вопрос можно дать несколько ответов, опишите их все.

3. Для выполнения задания можно использовать:

   - Все тетрадки, презентации и другие материалы предыдущих занятий.
   - Любые ресурсы в Интернете, документацию, форумы и проч.
   - Любые печатные материалы и проч.

   Однако, не прибегайте, пожалуйста, к помощи Ваших коллег (более формально: задание выполняется индивидуально). Почему – см. пункты 7 и 8.

4. Если вы копируете код с минимальными изменениями, не забывайте указывать источник (в виде комментария Python в ячейке с кодом).

5. Для работы Вам необходимо создать пустую тетрадку в Jupyter Notebook. Все задания необходимо выполнять в этой тетрадке. Все смысловые комментарии и текстовые пояснения следует писать в ячейках `Markdown`. Если Вы не помните, как работать с ячейками `Markdown`, откройте тетрадку с первого занятия и посмотрите, как эти ячейки устроены там (напоминание: чтобы посмотреть содержимое ячейки, щёлкните по ней левой кнопкой мыши два раза).

6. Оформление важно. В первой ячейке тетрадки сделайте заголовок Вашей работы по типу того, как это устроено в тетрадках с предыдущих занятий. Во второй ячейке напишите свои фамилию и имя. При выполнении заданий указывайте номер задания (По типу «2.1»). У Вашего файла должна быть понятная структура. Не забудьте про оформление графиков.

7. Это задание *не на оценку*. Выполняя его, Вы сможете лучше понять пройденный материал, а также потренироваться для выполнения домашнего задания. Постарайтесь сделать как можно больше заданий, но не страшно, если Вы не успеете сделать все.

8. Тетрадку с выполненным заданием необходимо прислать на почту vsomelyusik@gmail.com

   Это нужно сделать для того, чтобы получить комментарии, которые помогут Вам в выполнении текущего и будущего домашних заданий.

2 Разминка

Так как мы не закончили выполнение дополнительных заданий из первого занятия, сейчас самое время сделать их!

1. Откройте тетрадку из «Темы 1: Введение в Python». Скопируйте задания и заготовки кода из «Части 7: Дополнительные задания» в Вашу тетрадку.
2. Выполните эти задания.

3 Анализ данных

1. Импортируйте необходимые библиотеки.
2. Импортируйте данные к занятию и представьте их в виде `dataframe`. Изучите описание данных по ссылке (если ссылка не работает, на GitHub, скачайте файл с заданием на компьютер и откройте его в программе для работы с PDF). Кратко опишите, как Вы поняли, что из себя представляют данные.
3. Предложите две задачи регрессии и две задачи классификации, которые можно бы было поставить для этого набора данных (задачи сформулируйте в виде вопросов, например, «Как цена на жильё зависит от среднего числа комнат в квартире и уровня загрязнения воздуха в районе?» – задача регрессии).
4. На этом занятии Вам необходимо решить задачу регрессии, в которой зависимую переменную необходимо объяснить не более чем тремя, но не менее чем двумя, независимыми переменными. Выберите одну из формулировок задачи из предыдущего пункта или предложите новую.
5. С учётом предыдущего пункта, выберите в данных переменную, которую Вы будете считать зависимой (то есть которую Вы будете предсказывать), а также выберите независимые переменные (две или три). Помните о том, какого типа должна быть зависимая переменная в задаче регрессии.
6. Создайте новый `dataframe`, в который будут входить только выбранные в предыдущем пункте переменные (и зависимая, и независимые).

   В дальнейших заданиях речь идёт о данных, которые Вы сформировали сами в пункте 6.
7. Каково число наблюдений в данных?
8. Если считаете нужным, измените названия столбцов Вашего набора данных. Определите типы каждой независимой переменной.
9. Есть ли в данных пропущенные значения? Если да, то сколько их? Обработайте пропущенные значения так, как считаете нужным. Сколько наблюдений в данных теперь?
10. Постройте гистограммы всех переменных. Проинтерпретируйте результаты.
11. Постройте корреляционную матрицу переменных. Что показывает корреляция? Какие независимые переменные наиболее связаны с зависимой?
12. Постройте диаграммы рассеяния зависимой переменной (по оси Y) против каждой независимой переменной (по оси X). Проинтерпретируйте результат. Согласуются ли выводы с выводами предыдущего пункта?

13. Какой вид зависимости наблюдается для каждой независимой переменной (линейная / нелинейная, возрастающая / убывающая)? Приведите возможное объяснение наблюдаемым зависимостям.

4 Построение простой модели

1. Мы будем строить модель вида:
$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i},$$
где Y_i – зависимая переменная, X_1 – объясняющая переменная, имеющая наибольшую корреляцию с зависимой переменной, $\hat{\beta}_0$, $\hat{\beta}_1$ – некоторые коэффициенты.

   Как называется данный вид регрессии?

2. С математической точки зрения, что означает выписанное уравнение модели (то есть какой математический объект представляет собой это уравнение)?

3. Подробности того, как с теоретической точки зрения построить данную линию, мы узнаем на следующем занятии. Пока же мы доверимся библиотеке `sklearn`, которую также подробно разберём на следующих занятиях. Выполните следующую команду:

   `from sklearn.linear_model import LinearRegression`

   – и поясните, что она делает.

4. Выполните следующую команду:

   `model = LinearRegression()`

   где `model` – это название переменной, а `LinearRegression()` – класс, в котором содержится реализация используемой модели. Вспомните второе занятие и напишите, чем с точки зрения Python является переменная `model`.

5. Обучим модель! Обучение в данном случае заключается в подборе коэффициентов $\hat{\beta}_0$ и $\hat{\beta}_1$. Для обучения модели используется команда:

   `model.fit(X, Y)`

   где вместо `X` передаётся матрица объясняющих переменных, вместо `Y` – столбец зависимых переменных. Обычно переменные можно передавать обычным индексированием столбцов из `dataframe`, однако в случае одной объясняющей переменной могут возникнуть проблемы с размерностью. Тогда объясняющую переменную нужно представить в виде массива `numpy` правильной размерности. Пока не вдаваясь в подробности, для переменной `Humidity` это можно сделать так:

   `np.array(data['Humidity']).reshape(-1, 1)`

   Тогда полный код обучения модели будет выглядеть следующим образом:

   `model.fit(np.array(data['Humidity']).reshape(-1, 1), data['Temperature (C)'])`

   где `Temperature (C)` – зависимая переменная.

   Повторите эту процедуру для Вашей модели.

6. Для получения обученных коэффициентов воспользуйтесь командами: `model.coef_` – для коэффициента наклона $\hat{\beta}_1$, `model.intercept_` – для коэффициента сдвига $\hat{\beta}_0$. Проинтерпретируйте знаки коэффициентов. Соотносятся ли они с выводами об этой объясняющей переменной, полученными из предыдущего анализа?

7. Постройте на одном графике:

a) Диаграмму рассеяния $\hat{Y}_i$ и X_{1i}.

b) Уравнение обученной модели (задайте его вручную, используя полученные коэффициенты).

Как Вы считаете, насколько хорошо модель соответствует данным? Велика ли её обобщающая способность?

5 Построение более сложной модели

1. В этой части мы будем строить модель вида:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \hat{\beta}_3 X_{3i},$$

где Y_i – зависимая переменная, X_i – объясняющие переменные (понятно, что это общий вид модели, и их количество равно тому, сколько Вы выбрали объясняющих переменной для Вашей модели), $\hat{\beta}$ – некоторые коэффициенты.

2. Выполните команду: `model2 = LinearRegression()`. По аналогии с пунктом 4 из предыдущего задания, поясните, что делает данная команда.

3. Адаптируйте следующую команду для Вашей модели:

```
model2.fit(data.loc[:, ['Humidity', 'Visibility (km)']], data['Temperature (C)'])
```

По аналогии с пунктом 5 из предыдущего задания, поясните, что делает данная команда.

4. По аналогии с пунктом 6 из предыдущего задания, получите коэффициенты обученной модели (коэффициенты в списке идут в соответствии с порядком независимых переменных, заданном при обучении модели).

5. Повторите пункт 7 из предыдущего задания для каждой независимой переменной в модели из этого задания.