Отчет по лабораторной работе №1
по курсу «Машинное обучение»

Студент группы 8О-308 Баженова Надежда, № по списку 1.

Контакты: nad110508@gmail.com
Работа выполнена: 30.03.19

Ахмед Самир Халид
Отчет сдан: 30.03.2019

1. Постановка задачи

Требуется сформировать/получить два набора данных соответствующие следующим критериям:

1) Один из датасетов должен представлять собой корпус документов. Язык, источник и тематика произвольна

2) Второй датасет должен содержать категориальные, количественные признаки. Для данного датасета определить предсказываемые признаки (для задачи регрессии и классфикации). Если такого признака нет, спроектировать

Данные датасеты будут в дальнейшем использованы в оставшихся лабораторных работах.

По каждому датасету построить распределения признаков (в случае корпуса документов – построить распределение слов) и объяснить имеющуюся картину. Вычислить статистические характеристики признаков. Обнаружить и решить возможные проблемы с данными. Если решить данную проблему невозможно, объяснить почему.

2. Требования

1) Датасеты должны быть уникальны

2) Исходный код должен быть написан в одном код стайле

3) Должен быть указан источник данных

3. Описание выполненной работы.

3.Описание проделанной работы

Текстовый датасет - набор данных для моделирования языка WikiText из более чем 100 миллионов токенов, извлеченных из набора проверенных хороших и популярных статей в Википедии. Набор данных доступен по лицензии Creative Commons Attribution-ShareAlike.

Ресурс: https://blog.einstein.ai/the-wikitext-long-term-dependency-language-modeling-dataset/

Проблемы данных: наличие заглавных букв (машина будет принимать одинаковые слова за разные, а значит, не сможет правильно собрать данные), наличие знаков препинания, наличие цифр (числа — это не слова, поэтому они будут лишь тратить память).

Решение: убрать большие буквы (есть проблема того, что имена собственные и нарицательные будут равны. Однако случаи, в которых действительно есть существенная разница между значениями, встречаются редко), убрать знаки, цифры.

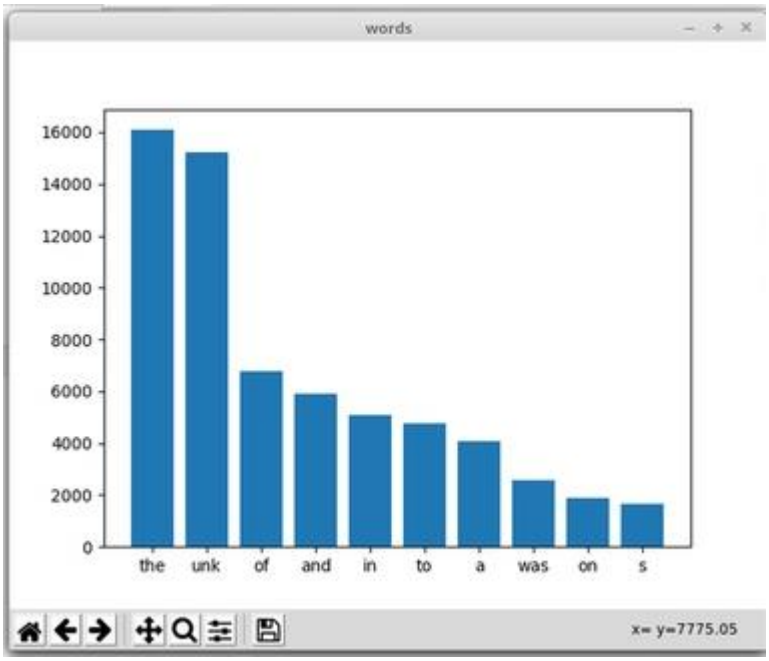
Для обработки датасета я использовала построчную обработку с дальнейшим преобразованием: я привела все буквы к нижнему регистру, отбросив все знаки препинания и цифры.

Далее я каждую строку разделила на слова и создала список всех слов, встреченных в файле.

С помощью конструктора построила словарь из всех встреченных слов и количеством вхождений каждого слова в файл.

Для просмотра результата я вывела словарь на экран с помощью форматного вывода.

Для анализа полученного словаря я нашла 10 самый часто встречающихся слов. Построила график распределения их частоты повторения. Самыми часто употребляемыми словами оказались предлоги. Действительно, в текстах предлоги употребляются чаще всего.



Код программы:

```
import pandas as pd
import pylab as plt
import numpy as np
import string

with open('wikibig.text', 'r') as f:
    data_text = f.readlines() #считываем строки из файл
    data_text_word = [] #объявление списка
    for line in data_text: # обработка каждой из строк
        line = line.lower() # приведение к нижнему регистру
        line = "".join(c for c in line if not c.isdigit() and c not in string.punctuation) #
удаление цифр и знаков припинания
        line = line.split() #преобразование строки к списку слов
```

```
        data_text_word.extend(line) # добавление элементов в список всех слов
    # конструирование словаря слово:число вхождений в текст
    dict_word_counts = {i: data_text_word.count(i) for i in list(set(data_text_word))}
    for i in dict_word_counts:
        print("%s" % (i.ljust(19)), dict_word_counts[i]) #ljust(n) - левоориентированный
вывод n знаков
  data = pd.DataFrame()
  data['values'] = dict_word_counts.values()
  data['key'] = dict_word_counts.keys()
  data = data.sort_values(by = 'values',ascending=False)
  top_words = data[:10]

  plt.figure('words')
  plt.bar(top_words['key'], top_words['values'])
  plt.show()
```

Второй датасет - данные о лесных пожарах в природных парках Montesinho, находящимся в Tr´as-os-Montes северном регионе Португалии.

1. X - пространственная координата оси X на карте парка Montesinho: от 1 до 9
2. Y - пространственная координата оси Y на карте парка Montesinho: от 2 до 9
3. месяц - месяц года: от «января» до «декабря»
4. день - день недели: с понедельника по воскресенье
5. FFMC - индекс FFMC из системы FWI: с 18,7 до 96,20
6. DMC - индекс DMC из системы FWI: от 1,1 до 291,3
7. DC - индекс DC от системы FWI: от 7,9 до 860,6
8. ISI - индекс ISI из системы FWI: от 0,0 до 56,10.
9. temp - температура в градусах Цельсия: от 2,2 до 33,30.
10. RH - относительная влажность в%: от 15,0 до 100
11. ветер - скорость ветра в км / ч: от 0,40 до 9,40
12. дождь - наружный дождь в мм / м2: от 0,0 до 6,4
13. площадь - сожженная площадь леса (в га): от 0,00 до 1090,84

Категориальные признаки: month, day, X, Y

Количественные: FFMC, DMC, DC, ISI, temp, RH, wind, rain, area

Расшифровка признаков

Индексы 5-8 являются компонентами канадская система оценки пожарной опасности.

The Fine Fuel Moisture Code (FFMC) представляет собой числовую оценку содержания влаги в поверхностном мусоре и других отвержденных тонких видах топлива. Это показывает относительную легкость воспламенения и воспламеняемость мелкого топлива. Содержание влаги в тонком топливе очень чувствительно к погоде. Даже день дождя или хорошей ветреной погоды существенно повлияет на рейтинг FFMC. Система использует временную задержку в две трети дня для точного измерения содержания влаги в тонких видах топлива. Рейтинг FFMC по шкале от 0 до 99.

LOW	MODERATE	HIGH	VERY HIGH	EXTREME
0.0-80.9	81.0-87.9	88.0-90.4	90.5-92.4	92.5+

Duff Moisture Code (DMC) - это числовая оценка среднего содержания влаги в слабо уплотненных органических слоях умеренной глубины. Код указывает глубину, на которой огонь будет гореть в умеренных слоистых слоях и древесных материалах среднего размера. Система применяет временную задержку в 12 дней для расчета DMC. Рейтинг DMC более 30 является сухим, а значение выше 40 указывает на то, что интенсивное горение будет происходить в слабом и среднем топливе. Операции сгорания не должны проводиться, когда рейтинг DMC выше 40.

LOW	MODERATE	HIGH	VERY HIGH	EXTREME
0.0-12.9	13.0-27.9	28.0-41.9	42.0-62.9	63.0+

The Drought Code (DC) - это числовая оценка содержания влаги в глубоких, компактных, органических слоях. Это полезный показатель сезонной засухи, который показывает вероятность возникновения пожара с глубокими слоями и большими бревнами. Для высыхания этих видов топлива и воздействия на DC необходим длительный период сухой погоды (система использует 52 дня). Номинальное значение постоянного тока 200 - высокое, а 300 и более - экстремальное, что указывает на то, что при пожаре будут использоваться глубокие подповерхностные и тяжелые виды топлива. Выгорание не должно быть разрешено, когда номинальное значение постоянного тока выше 300.

LOW	MODERATE	HIGH	VERY HIGH	EXTREME
0.0-79.9	80.0-209.9	210.0-273.9	274.0-359.9	360.0+

The Initial Spread Index (ISI) указывает, что скорострельность будет распространяться на ранних стадиях. Он рассчитывается на основе рейтинга FFMC и коэффициента ветра. Открытая шкала ISI начинается с нуля, а рейтинг 10 указывает на высокий уровень распространения вскоре после зажигания. Рейтинг 19 или более указывает на чрезвычайно высокую скорость распространения.

LOW	MODERATE	HIGH	VERY HIGH	EXTREME
0.0-3.9	4.0-7.9	8.0-10.9	11.0-18.9	19.0+

Ресурс: http://www3.dsi.uminho.pt/pcortez/forestfires/

Проблемы данных: наличие категориальных признаков не дает сделать полную статистическую оценку.

Решение: закодируем категории цифрами.

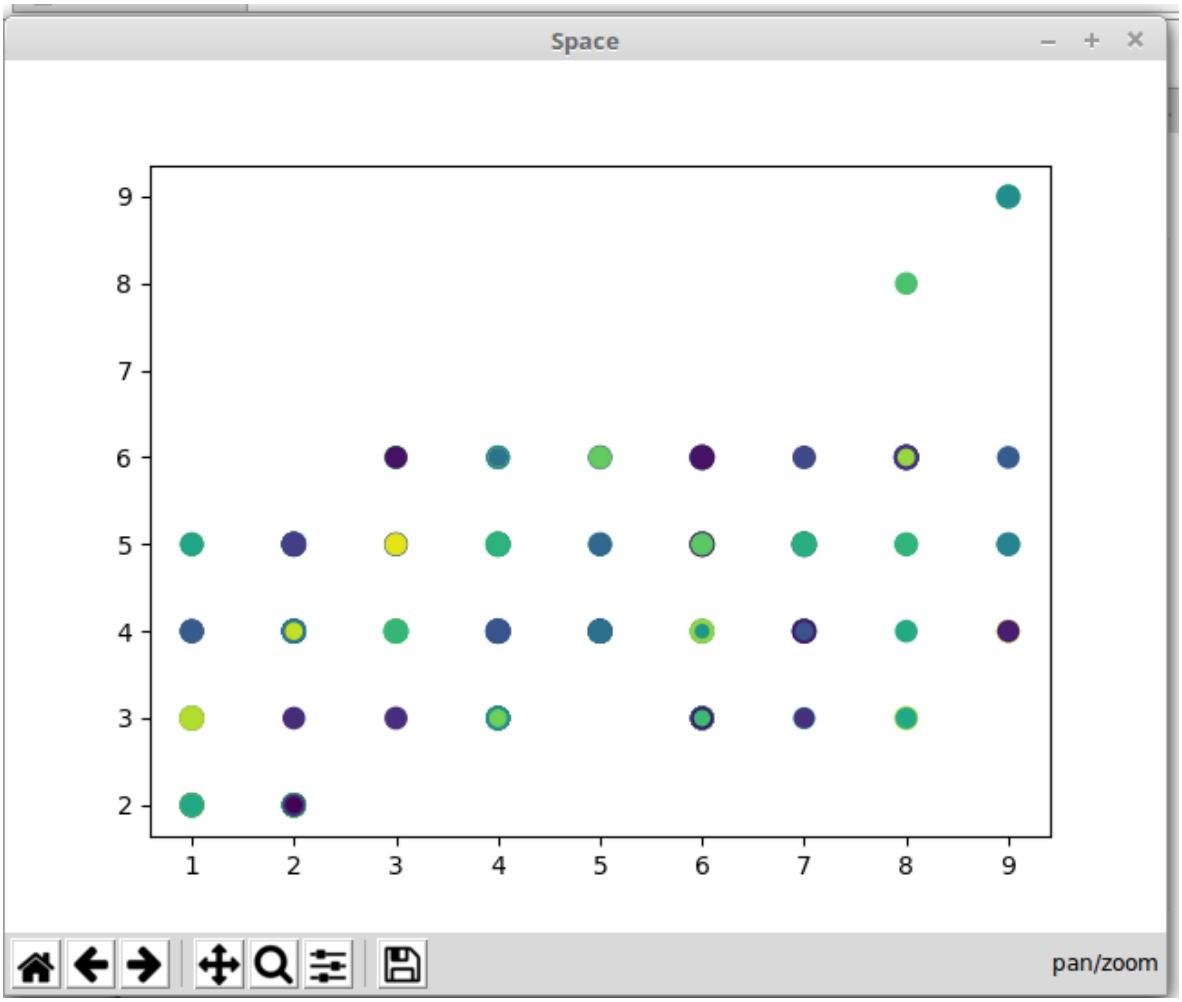
Анализ данных:

С помощью библиотеки pandas я считала данные, хранящиеся в файле forestfire.csv. Проанализировав, полученные данные, я решила переобозначить некоторые категориальные признаки, заменив их на цифры. Я пронумеровала дни недели и месяцы, создав 2 словаря, хранящих замены.

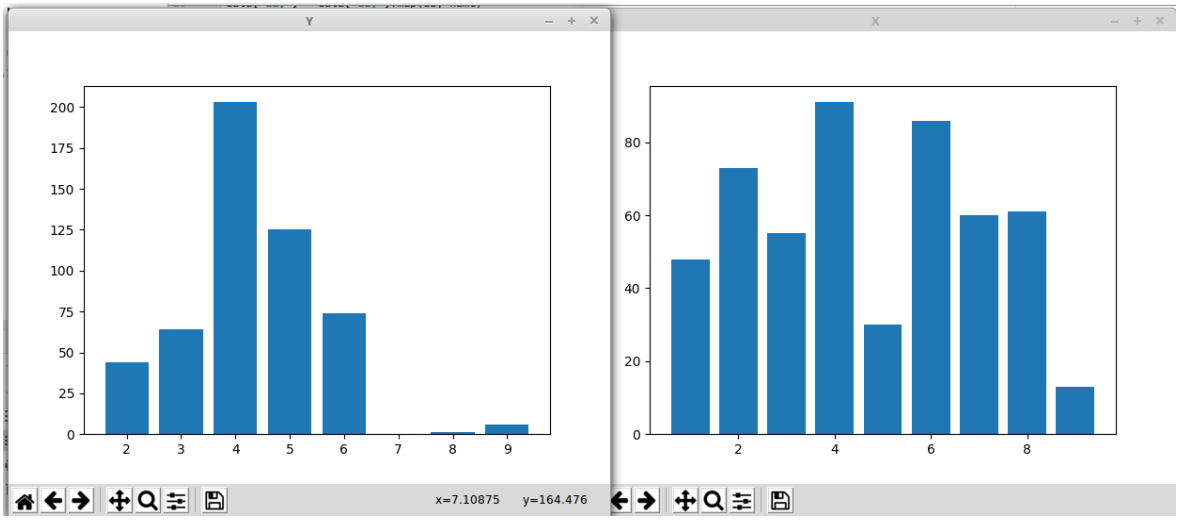
Это замена будет удобна при подсчете статистических признаков.

Распределения признаков:

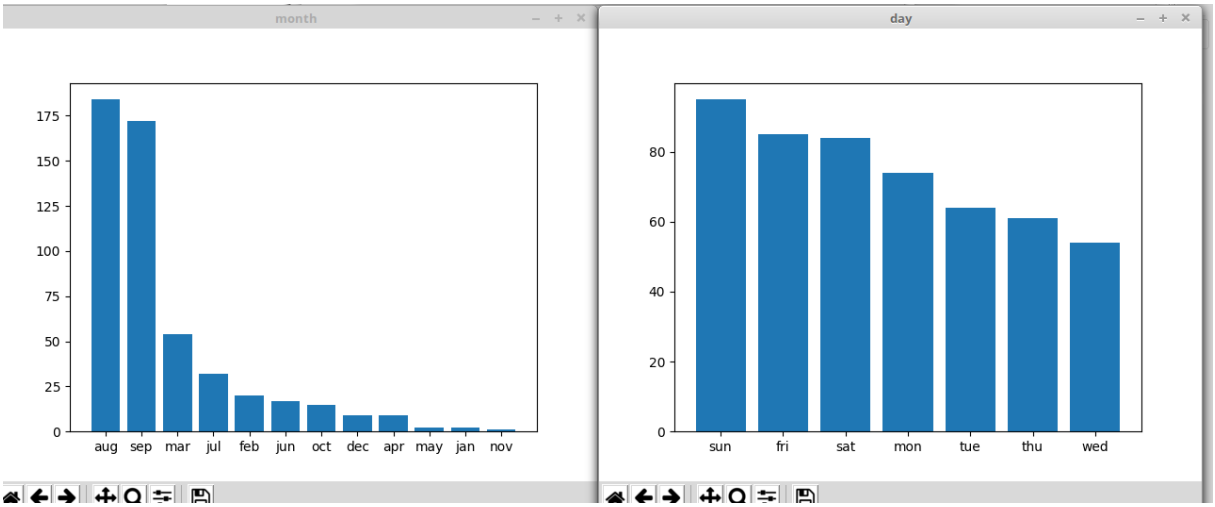
1. Распределение пожара по парку (ось абсцисс — координаты X, ось ординат — координаты Y) Видно, что есть квадраты, в которых пожары возникали неоднократно. Это можно использовать, чтобы усилить меры безопасности и слежения на таких участках.



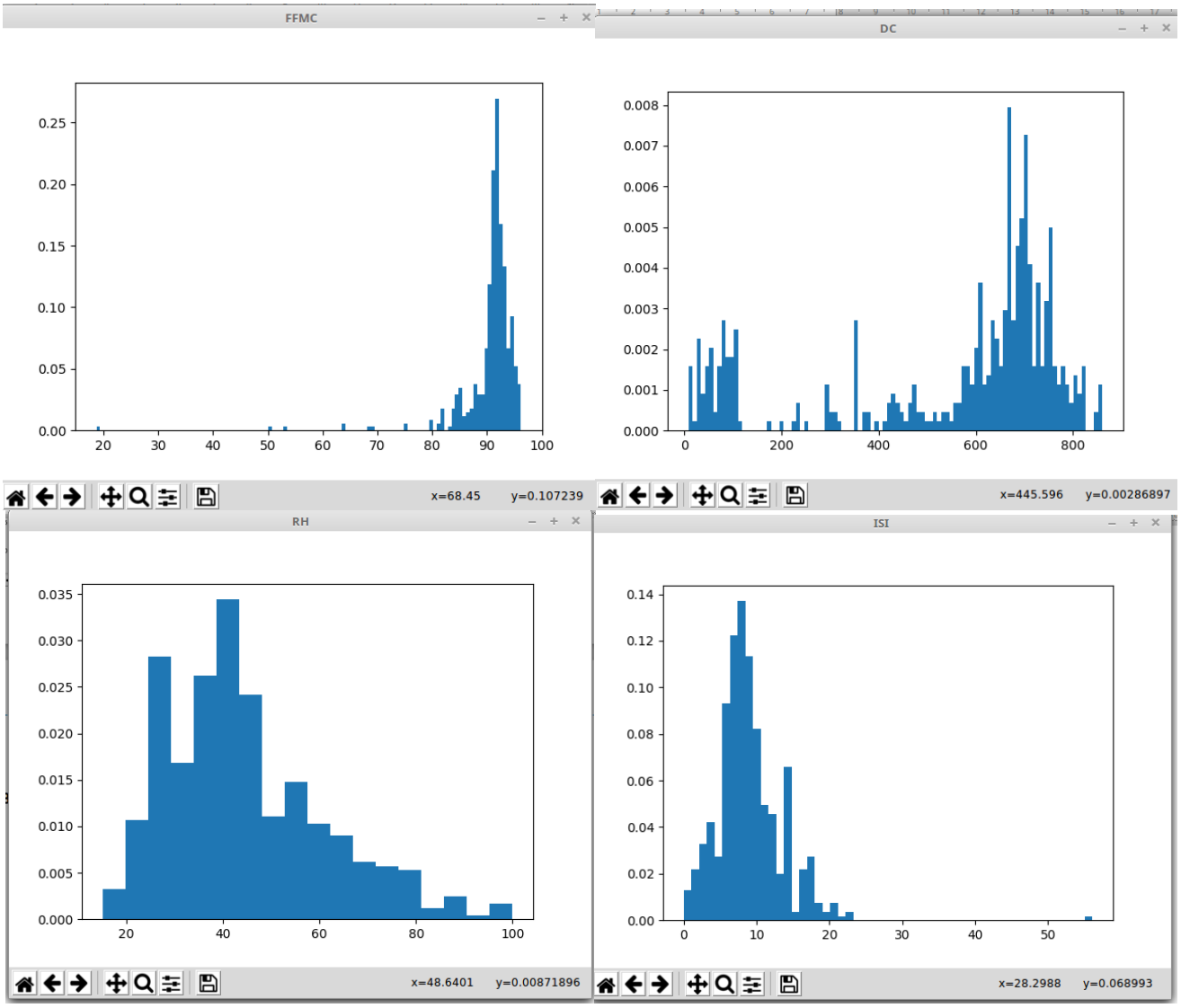
Распределение по осям X и Y разное. Однако оно подтверждает картину, полученную на графике выше: по координате X распределение близко к равномерному, а по Y больше похоже на нормальное.



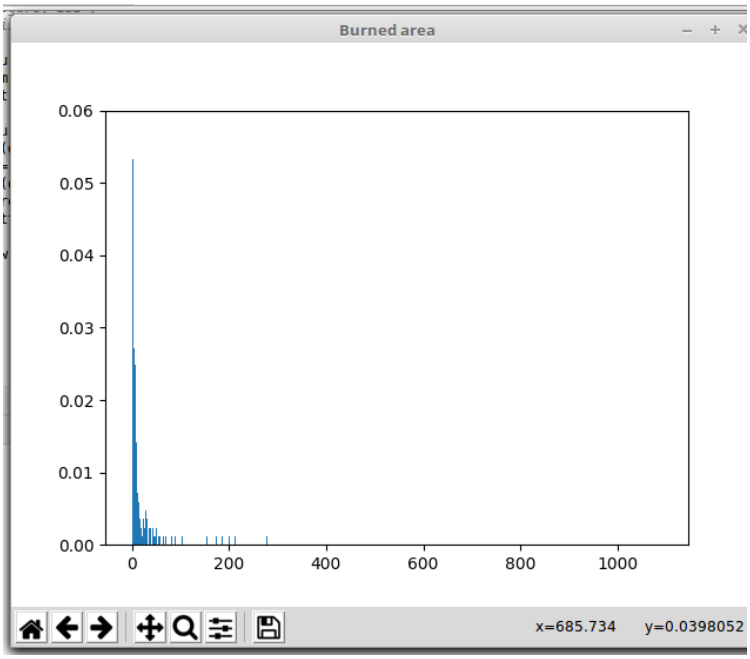
2. По распределению видно, что наибольшая вероятность пожара по месяцам – в августе, по дням недели – в воскресенье.



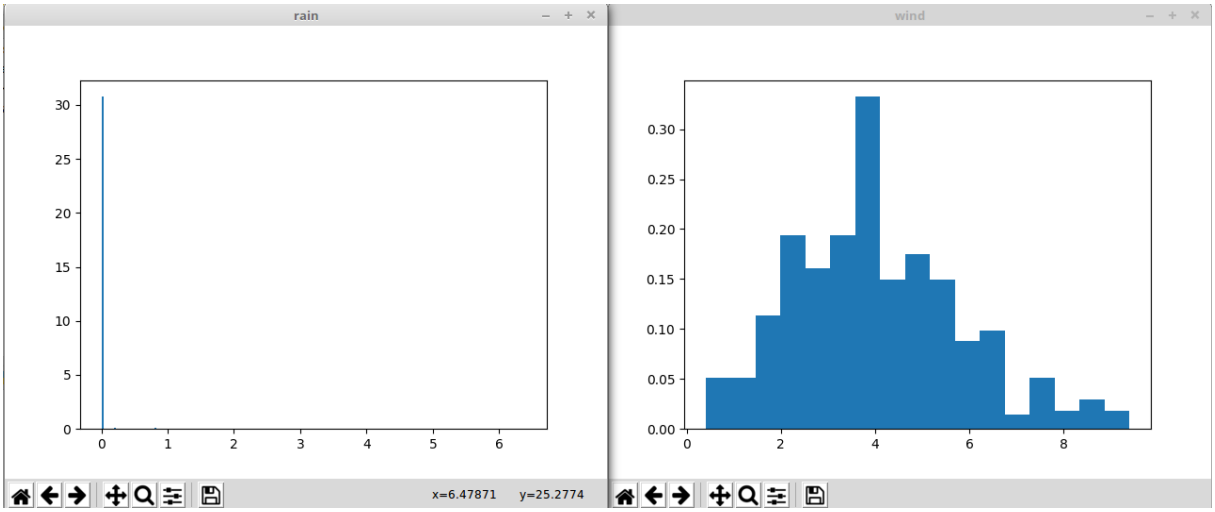
3.Проанализировав графики распределение индексов, можно сделать вывод, что распределение FFMC похоже на экспоненциальное распределение; RH, ISI похожи на нормальное распределение; DC – распределение не похоже на стандартные, четко выделяются 2 локальных максимума, т.к. это числовая оценка содержания влаги в глубоких, компактных, органических слоях, можно сделать вывод, что пожары с глубокими слоями происходят на сухих, либо сильно напитанных влагой участках глубокого слоя. При умеренной влажности возникновение пожара маловероятно.



4. Распределение площади сожженного леса отдаленно похоже на геометрическое распределение. По данным в большинстве случаев лес не страдает от пожаров, поэтому в выборке большое содержание нулевых значений. Из этого следует сильная неравномерность распределения.



5. Картина на первом графике вполне ожидаема. Т.к. пожары возникают в основном при отсутствии дождя, вероятность выпадения нулевого значения наиболее вероятна. На втором графике распределение похоже на нормальное. Объяснить равенство максимального значения скорости среднему значению можно объяснить тем, что такие аномалии в погоде как сильные ветры случаются редко.



Статистические оценки:

count() - количество;

mean() - среднее значение;

std() - стандартное отклонение;

min() - минимальный элемент;

max() - максимальный элемент;

50% - медиана;

0.25 и 0.75 квартили.

Static signs

x

count	517.000000
mean	4.669246
std	2.313778
min	1.000000
25%	3.000000
50%	4.000000
75%	7.000000
max	9.000000

Name: X, dtype: float64

y

count	517.000000
mean	4.299807
std	1.229900
min	2.000000
25%	4.000000
50%	4.000000
75%	5.000000
max	9.000000

Name: Y, dtype: float64

month

count 517.000000
mean 6.475822
std 2.275990
min 0.000000
25% 6.000000
50% 7.000000
75% 8.000000
max 11.000000
Name: month, dtype: float64

day
count 517.000000
mean 3.259188
std 2.072929
min 0.000000
25% 1.000000
50% 4.000000
75% 5.000000
max 6.000000
Name: day, dtype: float64

FFMC
count 517.000000
mean 90.644681
std 5.520111
min 18.700000
25% 90.200000
50% 91.600000
75% 92.900000
max 96.200000
Name: FFMC, dtype: float64

DC

count 517.000000

mean 547.940039

std 248.066192

min 7.900000

25% 437.700000

50% 664.200000

75% 713.900000

max 860.600000

Name: DC, dtype: float64

wind

count 517.000000

mean 4.017602

std 1.791653

min 0.400000

25% 2.700000

50% 4.000000

75% 4.900000

max 9.400000

Name: wind, dtype: float64

Задача классификации: классифицировать пожары на две группы: 1- происходящие с янв.-авг., 2 - с сен – дек. Данная классификация может пригодиться, например, при составлении сметы на требуемое оборудование и технику на следующий год.

Задача регрессии: определить в каких квадратах произойдет пожар, по имеющимся данным. Это упростит задачу отслеживания ситуации и позволит сконцентрировать людей, технику для борьбы с пожаром в нужном месте.

Код программы:

```
import pandas as pd
import pylab as plt
import numpy as np
data = pd.read_csv('forestfires.csv')
#создание словарей
month_name = {month: i for i, month in enumerate(['jan', 'feb', 'mar',
'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec'])}
day_name = {day: i for i, day in enumerate(['mon', 'tue', 'wed', 'thu',
'fri', 'sat', 'sun'])}
```

```
data['month'] = data['month'].map(month_name)
data['day'] = data['day'].map(day_name)
# статистические признаки
print('Static signs \n')
print('x\n', data['X'].describe())
print('\ny\n', data['Y'].describe())
print('\nmonth\n', data['month'].describe())
print('\nday\n', data['day'].describe())
print('\nFFMC\n', data['FFMC'].describe())
print('\nDC\n', data['DC'].describe())
print('\nwind\n', data['wind'].describe())
#построение графиков
plt.figure('X')
values = data['X'].value_counts()
plt.figure('X')
plt.bar(values.index, values)
plt.figure('Y')
plt.figure('Y')
values = data['Y'].value_counts()
plt.figure('Y')
plt.bar(values.index, values)
values = data['month'].value_counts()
plt.figure('month')
plt.bar(values.index, values)
values = data['day'].value_counts()
plt.figure('day')
plt.bar(values.index, values)
plt.figure('wind')
plt.hist(data['wind'],bins='auto', density=True)
plt.figure('rain')
plt.hist(data['rain'],bins=200, density=True)
plt.figure('DC')
plt.hist(data['DC'],bins=100, density=True)
plt.figure('FFMC')
plt.hist(data['FFMC'],bins='auto', density=True)
plt.figure('RH')
plt.hist(data['RH'],bins='auto', density=True)
plt.figure('ISI')
plt.hist(data['ISI'],bins='auto', density=True)
plt.figure('Burned area')
plt.ylim([0, 0.06])
plt.hist(data['area'],bins='auto', density=True)
plt.figure('Space')
N = len(data['X'])
colors = np.random.rand(N)# point color
area = (data['FFMC'])**5/10e7  # point radius
print(area)
plt.scatter(data['X'], data['Y'], s=area, c=colors, alpha=1)
plt.show()
```

4. Вывод

Делая данную лабораторную работу, я освоила базовые понятия, алгоритмы нового для меня языка Python, научилась писать базовые программы, познакомилась с библиотеками numpy.py, pandas.py, plot.py, которые пригодились мне в обработки данных.

Задача выбора, обработки данных является начальной и важной задачей в машинном обучении. Именно качество отобранной информации, эффективное хранение данных будут играть одну из определяющих ролей в дальнейшем обучении нашей машины и в результатах, которые она будет показывать на реальных данных.

В этой лабораторной работе я научилась строить графики. Передо мной стояла задача анализа графика, отражающего распределение признаков. Это очень сложная задача, требующая не только навыки программирования, но и знания теории вероятности. Я смогла сделать лишь предположительную оценку, основанную на полученных мною визуализациях и статистических признаках. Несомненно, улучшение качества оценки и более глубокий анализ будут хорошей доработкой моей работы.

Формулирование задач классификации и регрессии определяет успех и востребованность нашей машины в будущем. Поэтому важно хорошо продумать эту часть работы.

По завершении лабораторной работы, я имею хороший набор данных, проанализированный и готовый для дальнейшего обучения машины.