Шмаков Данила Борисович 181-351

Задание 1.1

Я использовал датасет IoT. Он содержит данные об отслеживании TCP, UDP и ICMP трафика, который включает в себя следующие 22 признака:

- Ts: время первого пакета (пример содержания: 1526756261.866500);
- Uid: уникальный идентификатор соединения (C9YvmJ3zxtuqxWxLW5);
- Id: ряд идентификаторов
    - Id_orig.h: ip создателя (192.168.2.5);
    - Id_orig.p: порт создателя (38792);
    - Id_resp.h: ip ответа (200.168.87.203);
    - Id_resp.p: порт ответа (59353).
- Proto: протокол (tcp);
- Service: идентификатор протокола приложения, отправляемого через соединение;
- Duration: Как долго длилась связь. При разрыве 3- или 4-стороннего соединения это не будет включать окончательный ACK. (2.998333);
- Orig_bytes: Количество байт полезной нагрузки, отправленных отправителем. Для TCP это берется из порядковых номеров и может быть неточным (например, из-за больших соединений);
- Resp_bytes: см orig_bytes, но только для ответа;
- Conn_state: возможные значения
    - S0: попытка подключения замечена, ответа нет;
    - S1: соединение установлено, но не разорвано;
    - SF: Нормальное установление и прекращение;
    - REJ: Попытка подключения отклонена;
    - S2: Установлено соединение и замечена попытка закрытия отправителем (но нет ответа от ответчика);
    - S3: соединение установлено и обнаружена попытка закрытия ответчиком (но нет ответа от отправителя);
    - RSTO: соединение установлено, отправитель прерван (отправлено RST);
    - RSTR: ответчик отправил RST;
    - RSTOS0: отправитель отправил SYN, за которым следует RST, нет SYN-ACK от ответчика;
    - RSTRH: Ответчик отправил SYN ACK, за которым следует RST, нет SYN от отправителя;

    - SH: Отправитель отправил SYN, за которым следует FIN, нет SYN ACK от ответчика (следовательно, соединение было «наполовину» открытым);
    - SHR: Ответчик отправил SYN ACK, за которым следует FIN, нет SYN от отправителя;
    - OTH: SYN не виден, только промежуточный трафик (один из примеров - «частичное соединение», которое не было позже закрыто).
- Local_orig: Если соединение создается локально, это значение будет T. Если оно было создано удаленно, оно будет F. В случае, если переменная Site :: local_nets не определена, это поле будет всегда оставаться пустым;
- Local_resp: : Если на соединение ответили локально, это значение будет T. Удаленно - F. В случае, если переменная Site :: local_nets не определена, это поле будет всегда оставаться пустым;
- Missed_bytes: Указывает количество байт, в пропусках содержимого, что свидетельствует о потере пакетов. Значение, отличное от нуля, обычно приводит к сбою анализа протокола, но некоторый анализ может быть завершен до потери пакета;
- History: Показывает историю подключений буквой или символом, которые значат:
    - s a SYN w/o the ACK bit set;
    - h a SYN+ACK (“handshake”);
    - a a pure ACK;
    - d packet with payload (“data”);
    - f packet with FIN bit set;
    - r packet with RST bit set;
    - c packet with a bad checksum (applies to UDP too);
    - g a content gap;
    - t packet with retransmitted payload;
    - w packet with a zero window advertisement;
    - i inconsistent packet (e.g. FIN+RST bits set);
    - q multi-flag packet (SYN+FIN or SYN+RST bits set).
- Orig_pkts: количество посланных пакетов;
- Resp_pkts: число отправленных ответных пакетов;
- Resp_ip_bytes: Количество байтов уровня IP, отправленных ответчиком;
- Tunnel_parents: значения uid для любых инкапсулирующих родительских соединений, используемых в течение времени существования этого внутреннего соединения;

- Label: Метка, в которой указано является трафик доброкачественным или вредоносным

Для следующих работ я разделю признаки на два класса и отмету часть из них, и так в ряд качественных признаков входят:

1. Uid – использоваться не будет
2. Proto
3. Service– использоваться не будет
4. Conn_state
5. Local_orig– использоваться не будет
6. Local_resp– использоваться не будет
7. History– использоваться не будет
8. Tunnel_parents– использоваться не будет

Количественных:

1. Ts– использоваться не будет
2. Все id– использоваться не будет
3. Duration
4. Orig/resp_bytes
5. Missed_bytes – использоваться не будет
6. Orig/resp_pkts
7. Resp_ip_bytes– использоваться не будет

Метка:

Label, в которой указано является трафик доброкачественным или вредоносных

Задание 1.2

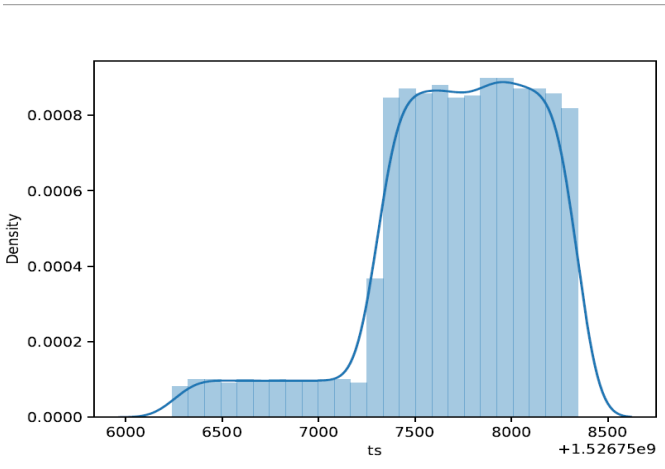
Так как в датасете в выбранных признаках встречается знак прочерка от него предварительно нужно избавиться. Заменим их на значением мат ожидания

```
data = pd.read_csv('IoT.csv', delimiter=',')
quantity = ['ts', 'duration', 'orig_bytes', 'resp_bytes', 'orig_pkts',
'resp_pkts']
data = pd.DataFrame(data, columns=['ts', 'duration', 'orig_bytes',
'resp_bytes', 'orig_pkts',
                                   'resp_pkts', 'proto', 'conn_state',
'label'])
```

```
for i in quantity:
    k = 0
    for item in data[i]:
        if item == '-':
            data[i][k] = np.nan
        else:
```

```
            data[i][k] = float(data[i][k])
        k += 1

imp = SimpleImputer(missing_values=np.nan, strategy='mean')
imp.fit(data[quantity])
data[quantity] = imp.transform(data[quantity])

print(data)
```

Для каждого рассматриваемого количественного признака:



ts

Минимумы = [1.52675689e+09 1.52675774e+09]

Матожидание: 1526757717.7921882

Интервалы: [(0, 1526756886.21), (1526756886.21, 1526757739.25), (1526757739.25, 1526758345.5189278)]

Кол-во интервалов: 3

Области однородности:

μ=763378443.1; доверительный интервал=(-1526756886.21, 3053513772.42)

μ=1526757312.73; доверительный интервал=(1526756033.17, 1526758592.29)

μ=1526758042.38; доверительный интервал=(1526757132.98, 1526758951.79)

duration

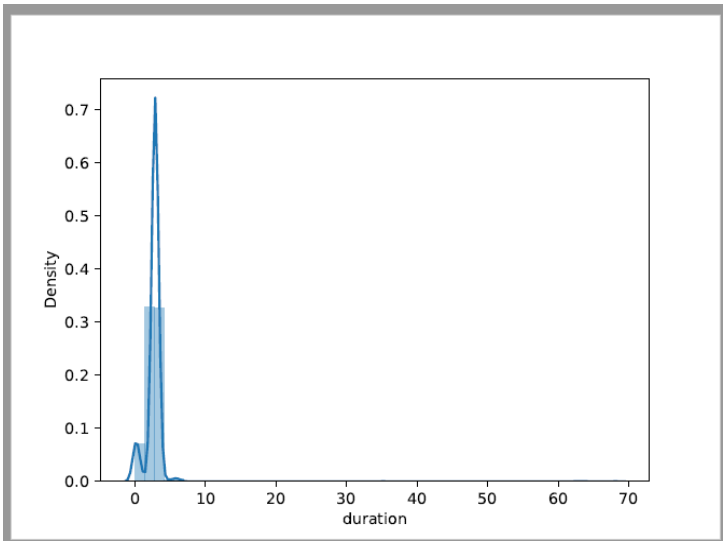
Минимумы = [1.44491459 5.00364234 8.56237009 13.90046171 26.00013606 36.67631931

 50.91123031 65.85788686]

Матожидание: 2.7124759669777454

Интервалы: [(0, 1.44), (1.44, 5.0), (5.0, 8.56), (8.56, 13.9), (13.9, 26.0), (26.0, 36.68), (36.68, 50.91), (50.91, 65.86), (65.86, 68.014546)]

Кол-во интервалов: 9



Области однородности:

μ=0.72; доверительный интервал=(-1.44, 2.88)

μ=3.22; доверительный интервал=(-2.12, 8.56)

μ=6.78; доверительный интервал=(1.44, 12.12)

μ=11.23; доверительный интервал=(3.22, 19.24)

μ=19.95; доверительный интервал=(1.8, 38.1)

μ=31.34; доверительный интервал=(15.32, 47.36)

μ=43.8; доверительный интервал=(22.45, 65.14)

μ=58.38; доверительный интервал=(35.96, 80.81)

μ=66.94; доверительный интервал=(63.71, 70.17)

orig_bytes

Минимумы = [98.38867243 147.23582775 237.95197333 311.22270631 391.47160432

513.58949261]

Матожидание: 21.217516152189518

Интервалы: [(0, 98.39), (98.39, 147.24), (147.24, 237.95), (237.95, 311.22), (311.22, 391.47), (391.47, 513.59), (513.59, 605.0)]

Кол-во интервалов: 7

Области однородности:

μ=49.2; доверительный интервал=(-98.39, 196.78)

μ=122.82; доверительный интервал=(49.54, 196.09)

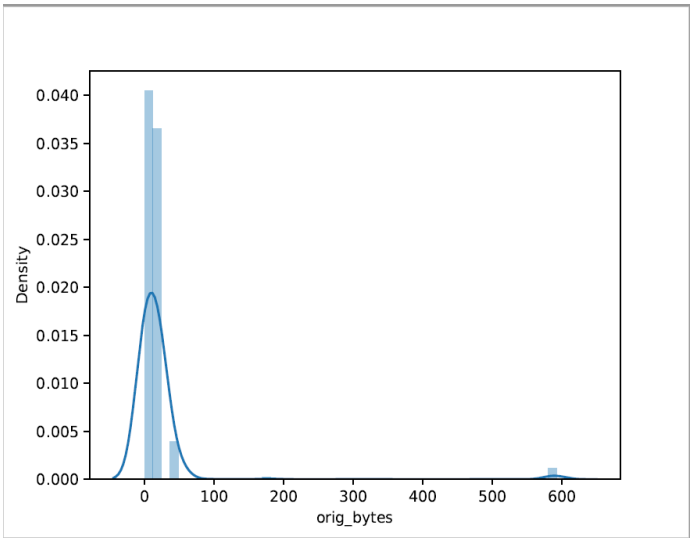
μ=192.6; доверительный интервал=(56.53, 328.66)

μ=274.59; доверительный интервал=(164.68, 384.49)

μ=351.34; доверительный интервал=(230.97, 471.72)

μ=452.53; доверительный интервал=(269.35, 635.71)

μ=559.3; доверительный интервал=(422.18, 696.41)

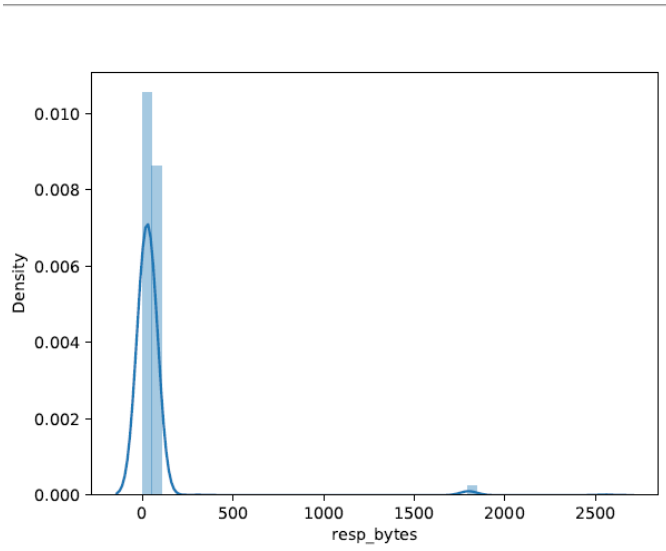


resp_bytes

Минимумы = [245.27084517 960.60129678 1647.31853032 2190.96967354]

Матожидание: 53.64680545585067

Интервалы: [(0, 245.27), (245.27, 960.6), (960.6, 1647.32), (1647.32, 2190.97), (2190.97, 2565.0)]



Кол-во интервалов: 5

Области однородности:

μ=122.64; доверительный интервал=(-245.27, 490.54)

μ=602.94; доверительный интервал=(-470.06, 1675.93)

μ=1303.96; доверительный интервал=(273.88, 2334.04)

μ=1919.14; доверительный интервал=(1103.67, 2734.62)

μ=2377.98; доверительный интервал=(1816.94, 2939.03)

orig_pkts

Минимумы = [2.00933588 4.56801033 5.48182263 6.66977863 10.41640907 14.16303951

16.5389515]

Матожидание: 2.095714857829395

Интервалы: [(0, 2.01), (4.57, 5.48), (5.48, 6.67), (6.67, 10.42), (10.42, 14.16), (14.16, 16.54), (16.54, 17.0)]

Кол-во интервалов: 7

Области однородности

μ=1.0; доверительный интервал= (-2.01, 4.02)

μ=5.03; доверительный интервал= (3.66, 6.39)

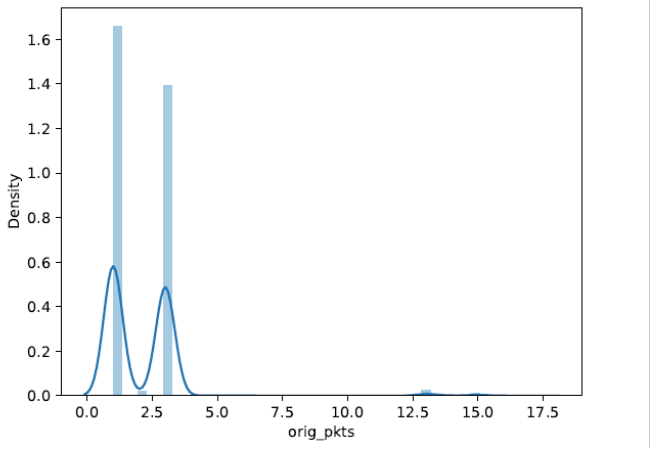
μ=6.08; доверительный интервал= (4.29, 7.86)

μ=8.54; доверительный интервал= (2.92, 14.17)

μ=12.29; доверительный интервал= (6.68, 17.9)

μ=15.35; доверительный интервал= (11.78, 18.92)

μ=16.77; доверительный интервал = (16.08, 17.46)

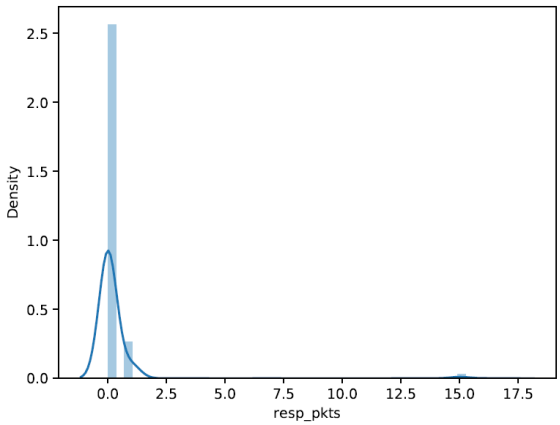


resp_pkts

Минимумы = [2.62988682 3.21204681 4.95852676 8.54851333 10.97417993 12.91471322]

Матожидание: 0.3263916700040048

Интервалы: [(0, 2.63), (2.63, 3.21), (3.21, 4.96), (4.96, 8.55), (8.55, 10.97), (10.97, 12.91), (12.91, 17.0)]



Кол-во интервалов: 7

Области однородности:

μ=1.32; доверительный интервал=(-2.63, 5.26)

μ=2.92; доверительный интервал=(2.05, 3.79)

μ=4.08; доверительный интервал=(1.46, 6.71)

μ=6.76; доверительный интервал= 1.37, 12.14)

μ=9.76; доверительный интервал=(6.13, 13.39)

μ=11.94; доверительный интервал=(9.03, 14.85)

μ=14.96; доверительный интервал=(8.82, 21.09)

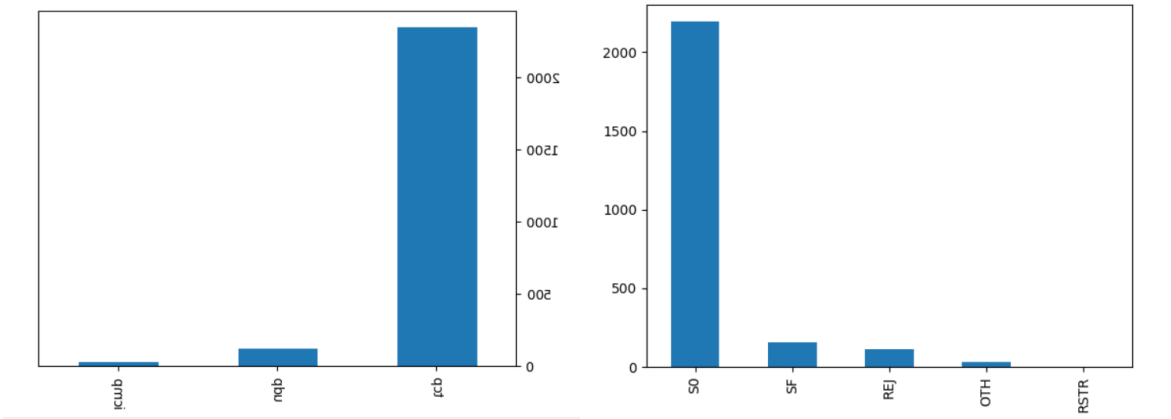
Программный код:

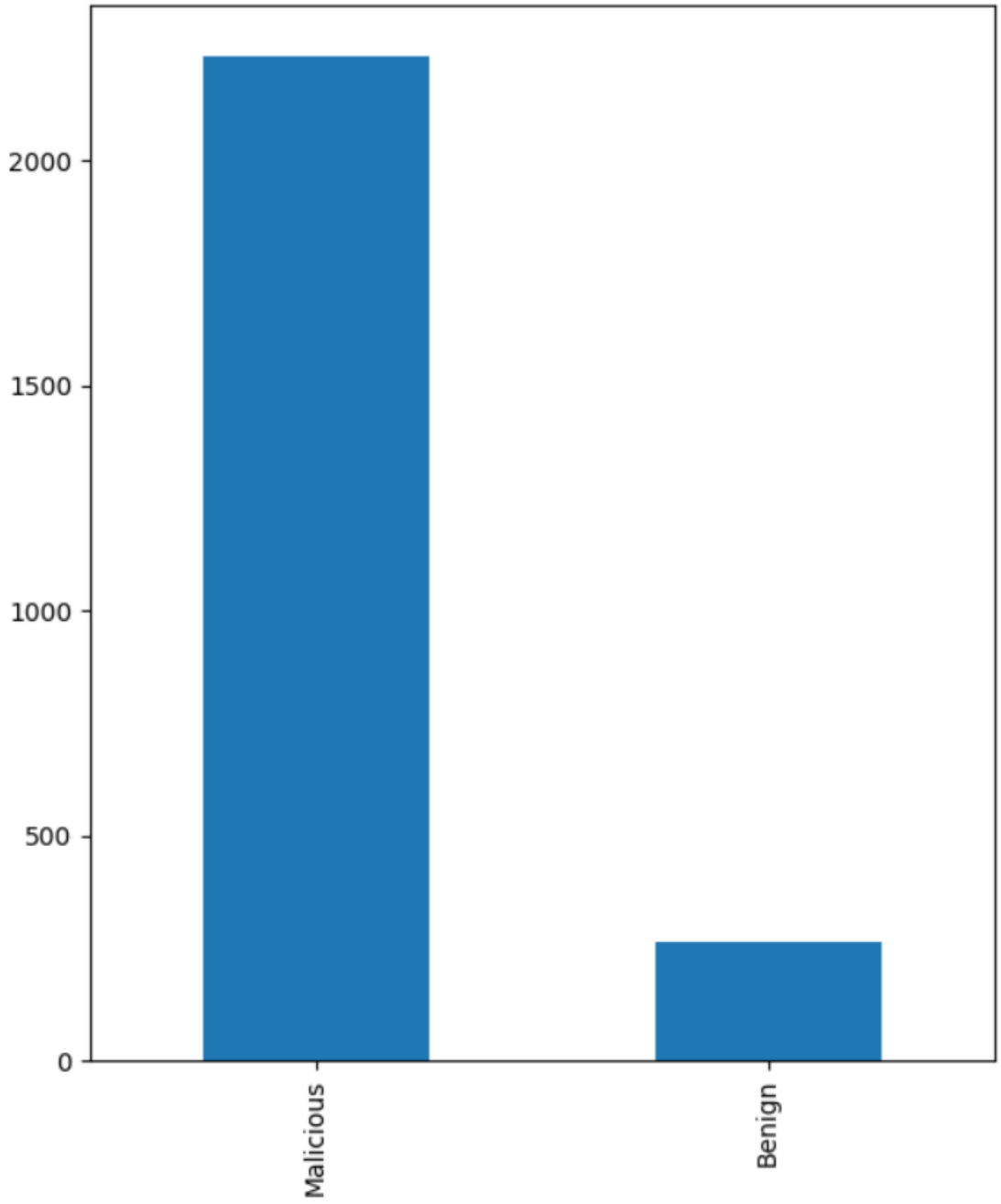
```
for item in quantity:
    plt.figure()
    sns.distplot(data[item])
    plt.savefig(f'{item}.pdf')
    #plt.show()

    x = data[item]
```

```
    mindata = sns.distplot(x).get_lines()[0].get_data()
    plt.clf()
    minIndex = argrelextrema(mindata[1], np.less)
    minimums = mindata[0][minIndex]
    print(f'{item}')
    print(f'Минимумы = {minimums}')
    print(f'Матожидание: {np.mean(x)}')
    intervals = [()]
    intervals[0] = (0, round(minimums[0], 2))
    for i in range(1, len(minimums)-1):
        intervals.append((round(minimums[i], 2), round(minimums[i+1], 2)))
    intervals.append((round(minimums[len(minimums)-1], 2), max(x)))
    print(f'Интервалы: {intervals}')
    print(f'Кол-во интервалов: {len(intervals)}')

    meanList = []
    for i in intervals:
        mean = np.mean(i)
        std = np.std(i)
        meanList.append((round(mean, 2), (round(mean-3*std, 2),
round((mean+3*std), 2))))
    print("Области однородности:")
    for ml in meanList:
        print(f"μ={ml[0]}; доверительный интервал={ml[1]}")
```

Качественные признаки:

```
 quality = ['proto', 'conn_state', 'label']

for item in quality:
    x = data[item]
    x.value_counts().plot.bar()
    plt.show()
```





Как можно заметить по этой гистограмме захватывается в основном вредоносный трафик