Лабораторная работа №5: «Выявление аномалий»

Набор данных **ex8data1.mat** представляет собой файл формата *.mat (т.е. сохраненного из Matlab). Набор содержит две переменные X_1 и X_2 - задержка в мс и пропускная способность в мб/с серверов. Среди серверов необходимо выделить те, характеристики которых аномальные. Набор разделен на обучающую выборку (X), которая не содержит меток классов, а также валидационную (Xval, yval), на которой необходимо оценить качество алгоритма выявления аномалий. В метках классов 0 обозначает отсутствие аномалии, а 1, соответственно, ее наличие.

Набор данных **ex8data2.mat** представляет собой файл формата *.mat (т.е. сохраненного из Matlab). Набор содержит 11-мерную переменную X - координаты точек, среди которых необходимо выделить аномальные. Набор разделен на обучающую выборку (X), которая не содержит меток классов, а также валидационную (Xval, yval), на которой необходимо оценить качество алгоритма выявления аномалий.

Задание.

1. Загрузите данные **ex8data1.mat** из файла.
2. Постройте график загруженных данных в виде диаграммы рассеяния.
3. Представьте данные в виде двух независимых нормально распределенных случайных величин.
4. Оцените параметры распределений случайных величин.
5. Постройте график плотности распределения получившейся случайной величины в виде изолиний, совместив его с графиком из пункта 2.
6. Подберите значение порога для обнаружения аномалий на основе валидационной выборки. В качестве метрики используйте F1-меру.
7. Выделите аномальные наблюдения на графике из пункта 5 с учетом выбранного порогового значения.
8. Загрузите данные **ex8data2.mat** из файла.
9. Представьте данные в виде 11-мерной нормально распределенной случайной величины.
10. Оцените параметры распределения случайной величины.
11. Подберите значение порога для обнаружения аномалий на основе валидационной выборки. В качестве метрики используйте F1-меру.
12. Выделите аномальные наблюдения в обучающей выборке. Сколько их было обнаружено? Какой был подобран порог?
13. Ответы на вопросы представьте в виде отчета.

Реализация:

In[1]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pnd
```

1.1 Загрузите набор данных ex8data1.mat из файла.

In[2]:

```
mat = loadmat('data/ex8data1.mat')
X = mat['X']
X_val = mat['Xval']
y_val = mat['yval']
y_val = y_val.reshape(y_val.shape[0])

X.shape
(307, 2)


X_val.shape
(307, 2)
```

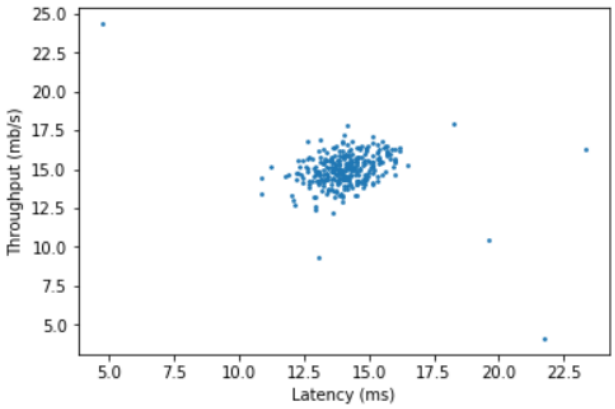
1.2 Постройте график загруженных данных в виде диаграммы рассеяния

In[3]:

```
plt.scatter(X[:,0],X[:,1], s=3)
plt.xlabel("Latency (ms)")
plt.ylabel("Throughput (mb/s)")
```

Out[3]:

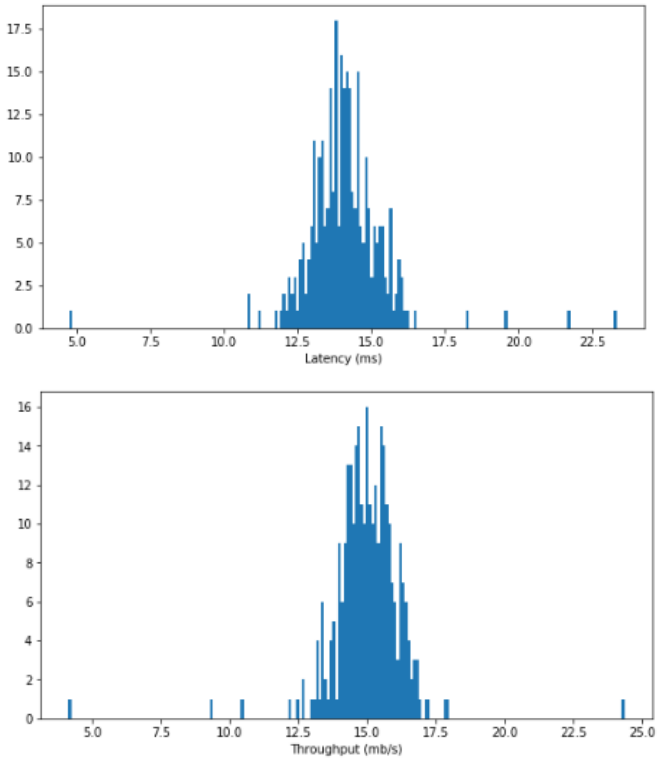
```
Text(0, 0.5, 'Throughput (mb/s)')
```



1.3 Представьте данные в виде двух независимых нормально распределенных случайных величин

In[4]:

```
x1, x2 = X[:, 0], X[:, 1]

fig, axs = plt.subplots(1, 2, figsize=(20, 5))
axs[0].hist(x1, bins=200)
axs[0].set_xlabel("Latency (ms)")

axs[1].hist(x2, bins=200)
axs[1].set_xlabel("Throughput (mb/s)")

plt.show()
```



1.4 Оцените параметры распределений случайных величин

Оба признака являются нормально распределенными случайными величинами.

Gaussian Distribution $p(x;\ \mu,\ \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$

$\mu_i = \frac{1}{m}\sum_{j=1}^{m} x^{(j)}$

$\sigma_i^2 = \frac{1}{m}\sum_{j=1}^{m} (x^{(j)} - \mu_j)^2$

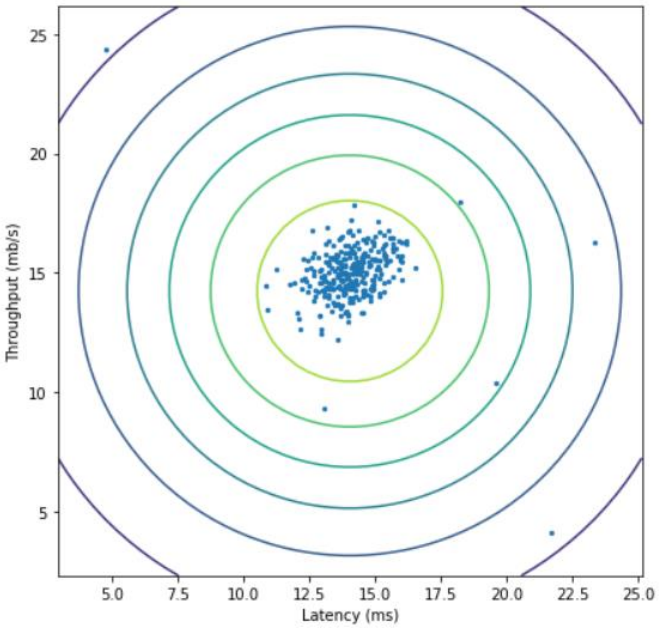
In[5]:

```
def estimate_gaussian(X):
    return X.mean(axis=0), X.std(axis=0)

mu, sigma = estimate_gaussian(X)
```

1.5 Постройте график плотности распределения получившейся случайной величины в виде изолиний, совместив его с графиком из пункта 2

In[6]:

```
import scipy.stats as stats

def p(X):
    axis = int(len(X.shape) > 1)
    mu, sigma = estimate_gaussian(X)
    return stats.norm.pdf(X, mu, sigma).prod(axis=axis)

x, y = X[:, 0], X[:, 1]

h = 1.8
u = np.linspace(x.min() - h, x.max() + h, 50)
v = np.linspace(y.min() - h, y.max() + h, 50)
u_grid, v_grid = np.meshgrid(u, v)
Xnew = np.column_stack((u_grid.flatten(), v_grid.flatten()))
z = p(Xnew).reshape((len(u), len(v)))

fig, ax = plt.subplots(figsize=(7, 7))
ax.contour(u, v, z)
ax.scatter(x, y, s=6)

plt.xlabel("Latency (ms)")
plt.ylabel("Throughput (mb/s)")
plt.show()
```



1.6. Подберите значение порога для обнаружения аномалий на основе валидационной выборки. В качестве метрики используйте F1-меру

In[7]:

```
def predict_anomalies(X, mu, sigma, eps):
    axis = int(len(X.shape) > 1)
    p = stats.norm.pdf(X, mu, sigma).prod(axis=axis)
    res = p < eps
    return res.astype(int) if axis else int(res)

def calc_eps(y_val, p_y_val, X_val, mu, sigma):
    best_eps = 0
    best_F1 = 0

    stepsize = (max(p_y_val) - min(p_y_val))/1000
    eps_range = np.arange(p_y_val.min(), p_y_val.max(), stepsize)
    for eps in eps_range:
        predictions = predict_anomalies(X_val, mu, sigma, eps)
        tp = np.sum(predictions[y_val==1]==1)
        fp = np.sum(predictions[y_val==0]==1)
        fn = np.sum(predictions[y_val==1]==0)

        # compute precision, recall and F1
        prec = tp/(tp+fp)
        rec = tp/(tp+fn)

        F1 = (2*prec*rec)/(prec+rec)

        if F1 > best_F1:
            best_F1 = F1
            best_eps = eps

    return best_eps, best_F1

p_y_val = p(X_val)
mu, sigma = estimate_gaussian(X_val)
eps, f1_score = calc_eps(y_val, p_y_val, X_val, mu, sigma)

eps
```

Out[7]:

```
0.0001572946256973518
```