Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра информатики

Отчет по лабораторной работе:

Лабораторная работа №8 “ Выявление аномалий”

Выполнил: Карп Александр Игоревич

магистрант кафедры информатики

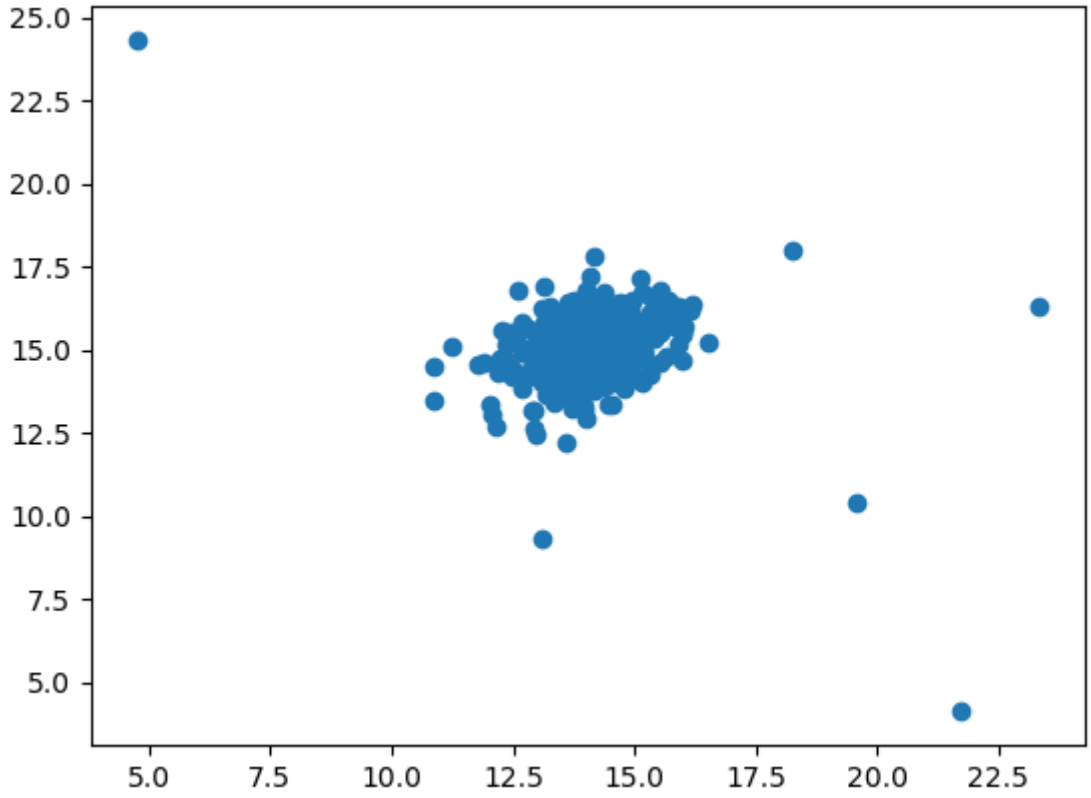
группа №858641

Минск 2019

1. Загрузите данные ex8data1.mat из файла.

```
#1
data = sio.loadmat('ex8data1.mat')
X = data.get('X')
Xval = data.get('Xval')
yval = data.get('yval')
```

2. Постройте график загруженных данных в виде диаграммы рассеяния.



На графике видны выбросы: они находятся в отдалении от основной массы элементов.

3. Представьте данные в виде двух независимых нормально распределенных случайных величин.

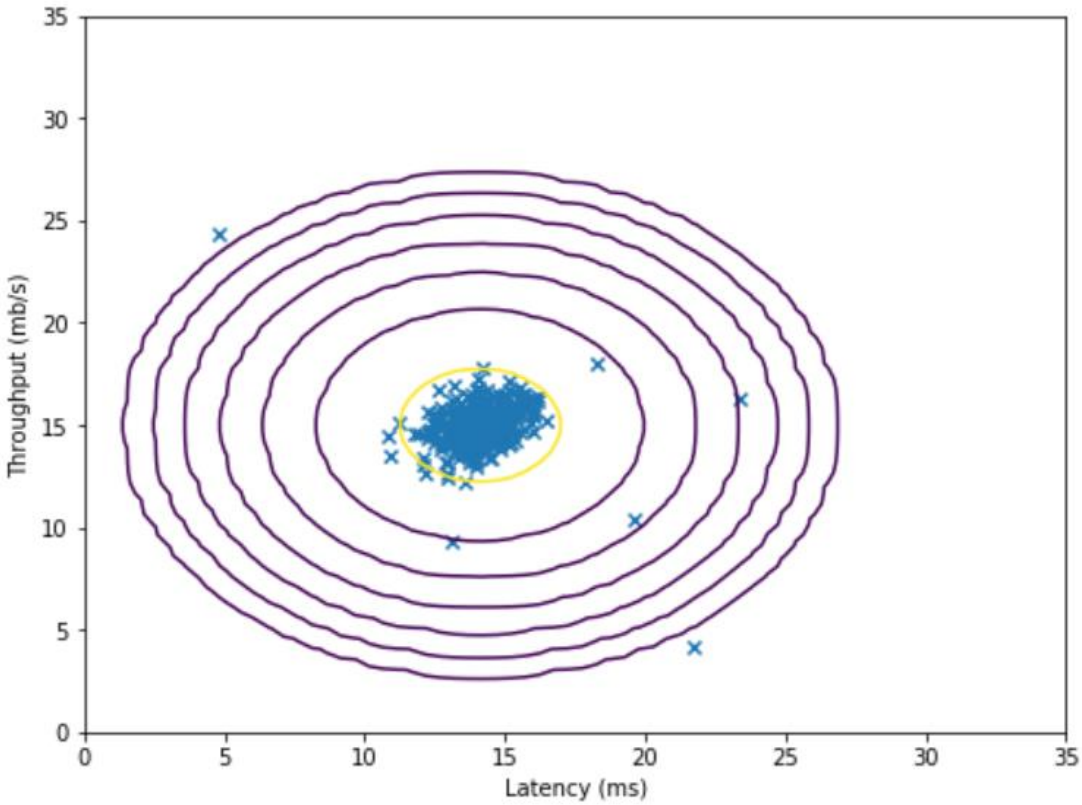
```
def estimateGaussian(X):
    m = X.shape[0]
    sum_ = np.sum(X, axis=0)
    mu = 1 / m * sum_
    var = 1 / m * np.sum((X - mu) ** 2, axis=0)
    return mu, var
```

4. Оцените параметры распределений случайных величин.

```
def multivariateGaussian(X, mu, sigma2):
    k = len(mu)

    sigma2 = np.diag(sigma2)
```

```
    X = X - mu.T
    p = 1 / ((2 * np.pi) ** (k / 2) * (np.linalg.det(sigma2) ** 0.5)) *
np.exp(
        -0.5 * np.sum(X @ np.linalg.pinv(sigma2) * X, axis=1))
    return p
```

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.

```
def selectThreshold(yval, pval):
    """
    Find the best threshold (epsilon) to use for selecting outliers
    """
    best_epi = 0
    best_F1 = 0

    stepsize = (max(pval) - min(pval)) / 1000
    epi_range = np.arange(pval.min(), pval.max(), stepsize)
    for epi in epi_range:
        predictions = (pval < epi)[:, np.newaxis]
        tp = np.sum(predictions[yval == 1] == 1)
        fp = np.sum(predictions[yval == 0] == 1)
        fn = np.sum(predictions[yval == 1] == 0)

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

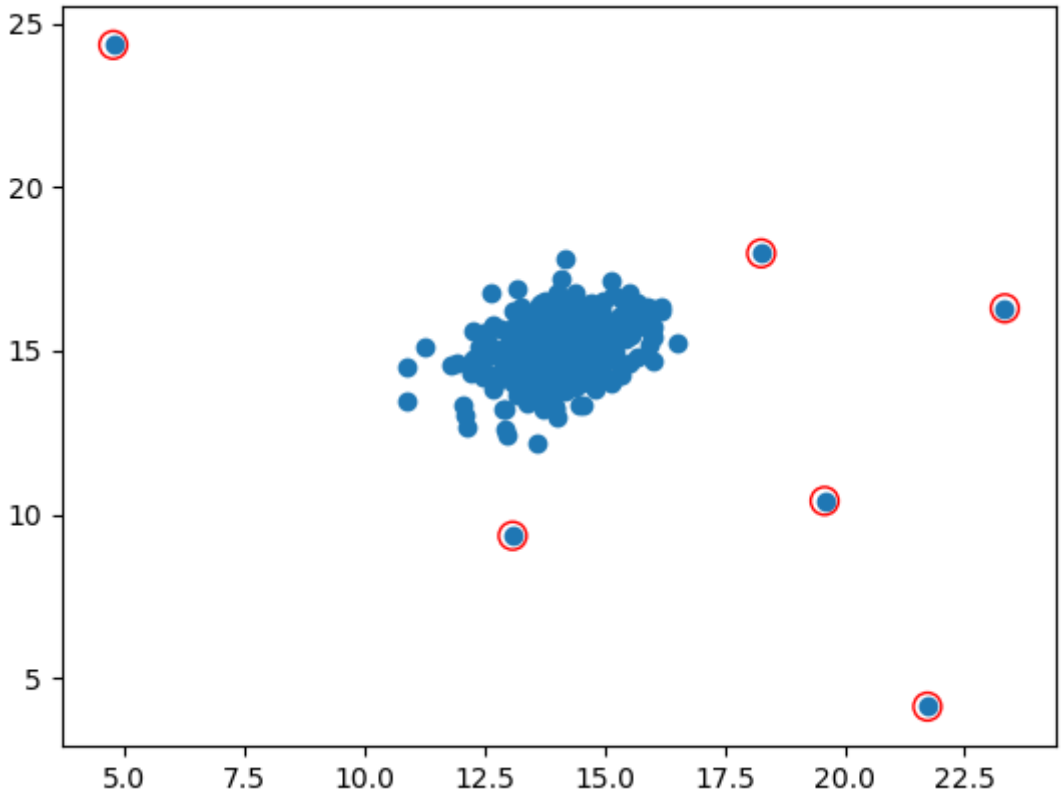
```
        F1 = (2 * prec * rec) / (prec + rec)

        if F1 > best_F1:
            best_F1 = F1
            best_epi = epi

    return best_epi, best_F1
```

$$Best\ epsilon\ found\ using\ cross-validation: 8.990852779269495e-05$$

$$Best\ F1\ on\ Cross\ Validation\ Set: 0.8750000000000001$$

7. Выделите аномальные наблюдения на графике из пункта 5 с учетом выбранного порогового значения.

```
#7
# Circling of anomalies
outliers = np.nonzero(p<epsilon)[0]
plt.scatter(X[outliers,0],X[outliers,1],marker
="o",facecolor="none",edgecolor="r",s=100)
plt.scatter(X[:,0], X[:, 1])
plt.show()
```



8. Загрузите данные **ex8data2.mat** из файла.

```
#8
data = sio.loadmat('ex8data2.mat')
X = data.get('X')
Xval = data.get('Xval')
yval = data.get('yval')
```

9. Представьте данные в виде 11-мерной нормально распределенной случайной величины.
10. Оцените параметры распределения случайной величины.

```
mu2, sigma2_2 = estimateGaussian(X)
#10
# Training set
p = multivariateGaussian(X, mu2, sigma2_2)
```

$$\mu = [\ 4.93\ \ -9.63\ \ 13.81\ \ -10.46\ \ -7.95\ 10.19\ \ -6.01\ \ 7.96\ \ -6.25\ \ \ 2.32\ 8.47]$$

$$\sigma^2 = [60.97\ 53.20\ 58.51\ 84.20\ 65.26\ 89.57\ 55.63\ 87.16\ 29.629\ 70.785\ \ 50.50]$$

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

```
# Find the best threshold
epsilon2, F1_2 = selectThreshold(yval, pval)
print("Best epsilon found using cross-validation:",epsilon2)
print("Best F1 on Cross Validation Set:",F1_2)
```

Best epsilon found using cross-validation: 1.3772288907613575e-18
Best F1 on Cross Validation Set: 0.6153846153846154

12. Выделите аномальные наблюдения в обучающей выборке. Сколько их было обнаружено? Какой был подобран порог?

```
print("# Outliers found:",np.sum(p<epsilon2))
```

Было найдено 117 аномалий при пороге 1.3772288907613575e-18

Выводы

В рамках данной работы был изучен метод нахождения аномалий в данных с использованием нормального распределения.