МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

ОТЧЕТ

по лабораторной работе №1

по дисциплине

Предиктивная аналитика

РУКОВОДИТЕЛЬ:

_____ Санников А.Н.

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

_____ Напылов Е.И.

(подпись) (фамилия, и.,о.)

_М22-ИВТ-1_____

(шифр группы)

Работа защищена «___» _____

С оценкой _____

Нижний Новгород 2023

Содержание

1. Постановка задачи

В данной работе требуется решить задачу классификации с помощью метода опорных векторов.

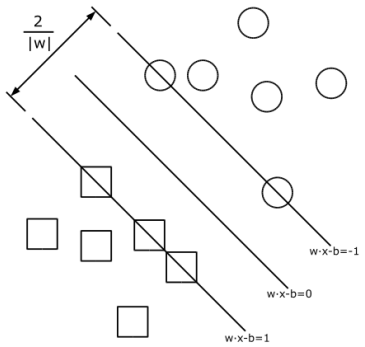
Был выбран датасет, содержащий данные о сетевых взаимодействиях. Требуется определить является ли взаимодействие опасным или безопасным по большому числу признаков трафика. Датасет содержит более 170000 объектов и 84 признака: IP, порты, размеры пакетов, скорости и т.п. Классы: Trojan - вирус и Benign - не вирус.

Список некоторых признаков из датасета:

```
 0   FlowID                      177482 non-null  object
 1   SourceIP                    177482 non-null  object
 2   SourcePort                  177482 non-null  int64
 3   DestinationIP               177482 non-null  object
 4   DestinationPort             177482 non-null  int64
 5   Protocol                    177482 non-null  int64
 6   Timestamp                   177482 non-null  object
 7   FlowDuration                177482 non-null  int64
 8   TotalFwdPackets             177482 non-null  int64
 9   TotalBackwardPackets        177482 non-null  int64
 10  TotalLengthofFwdPackets     177482 non-null  float64
 11  TotalLengthofBwdPackets     177482 non-null  float64
 12  FwdPacketLengthMax          177482 non-null  float64
 13  FwdPacketLengthMin          177482 non-null  float64
 14  FwdPacketLengthMean         177482 non-null  float64
 15  FwdPacketLengthStd          177482 non-null  float64
 16  BwdPacketLengthMax          177482 non-null  float64
 17  BwdPacketLengthMin          177482 non-null  float64
 18  BwdPacketLengthMean         177482 non-null  float64
 19  BwdPacketLengthStd          177482 non-null  float64
 20  FlowBytes/s                 177482 non-null  float64
 21  FlowPackets/s               177482 non-null  float64
 22  FlowIATMean                 177482 non-null  float64
 23  FlowIATStd                  177482 non-null  float64
 24  FlowIATMax                  177482 non-null  float64
 25  FlowIATMin                  177482 non-null  float64
 26  FwdIATTotal                 177482 non-null  float64
 27  FwdIATMean                  177482 non-null  float64
 28  FwdIATStd                   177482 non-null  float64
 29  FwdIATMax                   177482 non-null  float64
 30  FwdIATMin                   177482 non-null  float64
```

2. Метод опорных векторов

Метод опорных векторов (Support Vector Machine, SVM) - это алгоритм машинного обучения, который используется для классификации и регрессии. Он относится к группе методов, называемых линейными классификаторами, которые строят гиперплоскость для разделения данных разных классов. Суть метода заключается в том, чтобы найти гиперплоскость, которая максимально разделяет данные разных классов. Ширина полосы разделения максимизируется. Гиперплоскость - это n-мерная поверхность, где n - число признаков в нашем наборе данных. В случае двух классов, гиперплоскость является линией, которая разделяет два класса. SVM является мощным и гибким алгоритмом, который может быть применен в различных задачах машинного обучения. Однако, он также может быть чувствителен к выбору параметров и ядерной функции, и может иметь проблемы в случае несбалансированных классов.



В математической форме это выглядит так:

$$\begin{cases} \|\mathbf{w}\|^2 \to \min \\ c_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1, \quad 1 \leq i \leq n. \end{cases}$$

По теореме ККТ эта задача эквивалентна двойственной задаче поиска седловой точки функции Лагранжа:

$$\begin{cases} \mathbf{L}(\mathbf{w}, \mathbf{b}; \lambda) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \lambda_\mathbf{i}(c_i((\mathbf{w} \cdot \mathbf{x_i}) - b) - 1) \to \min_{w,b} \max_{\lambda} \\ \lambda_\mathbf{i} \geq 0, \quad 1 \leq i \leq n \end{cases}$$

$\lambda = (\lambda_\mathbf{1}, \ldots, \lambda_\mathbf{n})$ — вектор двойственных переменных.

Затем задача сводится к задаче квадратичного программирования:

$$\begin{cases} -\mathbf{L}(\lambda) = -\sum_{i=1}^{n} \lambda_\mathbf{i} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_\mathbf{i}\lambda_\mathbf{j}c_ic_j(\mathbf{x_i} \cdot \mathbf{x_j}) \to \min_{\lambda} \\ \lambda_\mathbf{i} \geq 0, \quad 1 \leq i \leq n \\ \sum_{i=1}^{n} \lambda_\mathbf{i}c_i = 0 \end{cases}$$

$$\mathbf{w} = \sum_{i=1}^{n} \lambda_\mathbf{i}c_i\mathbf{x_i} \qquad \mathbf{b} = \mathbf{w} \cdot \mathbf{x_i} - c_i, \quad \lambda_i > 0$$

$$a(x) = sign\left(\sum_{i=1}^{n} \lambda_\mathbf{i}c_i\mathbf{x_i} \cdot \mathbf{x} - b\right)$$

3. Данные и их обработка

Датасет содержит 85 различных признаков, такие как IP, порт, размер пакета, число пакетов, скорость соединения и т.д. Классами являются метки Trojan - вирус и Benign - не вирус. Датасет содержит более 170 000 записей сетевых взаимодействий.

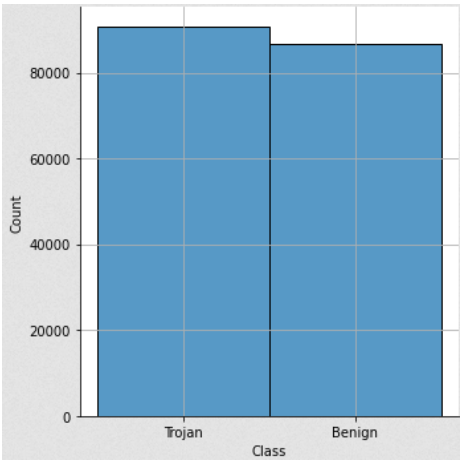
```
data.head()
```

	FlowID	SourceIP	SourcePort	DestinationIP	DestinationPort	Protocol	Timestamp	FlowDuration	TotalFwdPacke
73217	10.42.0.42-121.14.255.84-49975-80-6	10.42.0.42	49975	121.14.255.84	80	6	17/07/201701:18:33	10743584	
72089	172.217.6.226-10.42.0.42-443-49169-17	10.42.0.42	49169	172.217.6.226	443	17	17/07/201710:25:25	254217	
96676	10.42.0.1-10.42.0.42-53-37749-17	10.42.0.42	37749	10.42.0.1	53	17	30/06/201707:16:12	1023244	
42891	10.42.0.1-10.42.0.42-53-41352-17	10.42.0.42	41352	10.42.0.1	53	17	13/07/201703:48:44	286483	
169326	10.42.0.151-107.22.241.77-44353-443-6	10.42.0.151	44353	107.22.241.77	443	6	05/07/201710:47:35	65633087	

5 rows × 85 columns

Диаграммы рассеивания наиболее интересных признаков:

Классы сбалансированы:



Текстовые данные (['FlowID', 'SourceIP', 'DestinationIP', 'Timestamp', 'Class']) были закодированы в числа с помощью sklearn.preprocessing.LabelEncoder.

Метки классов закодированы 0 и 1.

```
data.at[data['Class'] == 'Trojan', 'Class'] = 1
data.at[data['Class'] == 'Benign', 'Class'] = 0
```

В результате после всей обработки датасет выглядит так:

	FlowID	SourceIP	SourcePort	DestinationIP	DestinationPort	Protocol	Timestamp	FlowDuration	TotalFwdPackets	TotalBackwardPackets	...	min_
73217	-0.254680	-0.301850	0.615199	-0.551040	-0.374989	-0.505653	1.285073	-0.038628	-0.092527	-0.055528	...	
72089	0.846469	-0.301850	0.571479	0.048341	-0.351038	1.916987	1.531425	-0.494133	-0.025086	-0.027333	...	
96676	-1.665593	-0.301850	-0.047966	-0.931863	-0.376771	1.916987	1.765842	-0.460738	-0.193688	-0.083723	...	
42891	-1.619663	-0.301850	0.147468	-0.931863	-0.376771	1.916987	0.755896	-0.492732	-0.193688	-0.083723	...	
169326	-1.224047	-0.307994	0.310249	-0.696746	-0.351038	-0.505653	-0.346225	2.344973	0.177238	0.000862	...	

5 rows × 85 columns

4. Обучение SVM и результаты

Данные были поделены на обучающую и тестовую выборки в соотношении 70 на 30.

```
X = data.drop('Class', axis = 1).to_numpy()
y = np.array(data['Class'], dtype='int')
```

```
X.shape, y.shape
```

```
((177482, 84), (177482,))
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)
```

```
y_train.shape, y_test.shape
```

```
((124237,), (53245,))
```

Модель достаточно долго обучалась, возможно, это проблема sklearn-a.

```
%%time
model = svm.SVC(verbose=True)
```

```
Wall time: 0 ns
```

```
%%time
model.fit(X_train, y_train)
```

```
[LibSVM]Wall time: 10min 42s

SVC(verbose=True)
```

Результаты на обучающей выборке:

	precision	recall	f1-score	support
0	0.97	0.93	0.95	60725
1	0.93	0.98	0.95	63512
accuracy			0.95	124237
macro avg	0.95	0.95	0.95	124237
weighted avg	0.95	0.95	0.95	124237

Результаты на тестовой выборке:

	precision	recall	f1-score	support
0	0.97	0.92	0.95	26074
1	0.93	0.97	0.95	27171
accuracy			0.95	53245
macro avg	0.95	0.95	0.95	53245
weighted avg	0.95	0.95	0.95	53245

Точность достигла значения 0.95, что является очень хорошим результатом. При этом на обучающей и тестовой выборке точность идентична, что является идеальным результатом - отсутствует недообучение и переобучение.

5. Выводы

В результате работы была решена задача бинарной классификации сетевого трафика на безопасный и вирусный по большому число признаков. Для этого была проведена предобработка данных - кодирование текстовых признаков и последующая нормализация всего датасета. Для классификации был использован метод опорных векторов, который показал отличные результаты - точность 0.95. При этом удалось получить идеальное поведение на тестовой и обучающей выборке, при которой точность оказалась идентичной, что говорит о полном отсутствии переобучения и недообучения.