

Лабораторная работа 8

Анализ SQL injection атак моделями машинного обучения

Дан датасет SQL injection атак

<https://www.kaggle.com/datasets/gambleryu/biggest-sql-injection-dataset>

Разработать и протестировать модели машинного обучения для классификации атак SQL Injection на основе датасета Biggest SQL Injection Dataset. Провести сравнительный анализ различных алгоритмов.

Этапы выполнения

1. Подготовка данных

1. Загрузка данных

- Скачать датасет с Kaggle и загрузить в среду разработки (Google Colab, Jupyter Notebook, PyCharm).
- Использовать pandas и numpy для работы с данными.

2. Анализ данных

- Определить целевую переменную (атака SQL Injection / безопасный запрос).
- Проверить баланс классов (value_counts()).
- Изучить текстовые признаки (например, URL, HTTP-запросы, параметры).
- Проверить наличие пропущенных значений (df.isnull().sum()).

3. Предобработка данных

- Заполнить или удалить пропущенные значения.
- Очистить текстовые данные (удаление специальных символов, нормализация).
- Преобразовать текст в числовой формат (использовать TF-IDF, CountVectorizer, Word2Vec, FastText).
- Разделить данные на обучающую и тестовую выборки (train_test_split).

2. Обучение моделей машинного обучения

Обучить и протестировать следующие модели:

1. Наивный Байесовский классификатор (Naïve Bayes)

- MultinomialNB или GaussianNB из sklearn.naive_bayes.
- Хорошо работает с текстовыми данными (TF-IDF).

2. Логистическая регрессия (Logistic Regression)

- LogisticRegression из sklearn.linear_model.
- Проверить влияние параметров (penalty, C).

3. Метод опорных векторов (Support Vector Machine, SVM)

- SVC из sklearn.svm с разными ядрами (linear, rbf).
- Оптимизация C и gamma.

4. Метод k-ближайших соседей (k-Nearest Neighbors, k-NN)

- KNeighborsClassifier из sklearn.neighbors.
- Оптимизация k с кросс-валидацией.

5. Дерево решений (Decision Tree)

- DecisionTreeClassifier из sklearn.tree.
- Настроить max_depth, criterion.

6. **Случайный лес (Random Forest)**
 - RandomForestClassifier из sklearn.ensemble.
 - Оптимизация n_estimators, max_depth.
7. **Градиентный бустинг (XGBoost)**
 - XGBClassifier из xgboost.
 - Оптимизация learning_rate, n_estimators, max_depth.
8. **CatBoost**
 - CatBoostClassifier из catboost.
 - Оптимизация iterations, depth, learning_rate.
9. **AdaBoost**
 - AdaBoostClassifier из sklearn.ensemble.
 - Подбор n_estimators, learning_rate.

3. Оценка моделей

1. **Метрики качества**
 - accuracy
 - precision
 - recall
 - F1-score
 - ROC-AUC
2. **Кросс-валидация**
 - KFold или StratifiedKFold для проверки устойчивости моделей.
3. **Матрица ошибок (Confusion Matrix)**
 - confusion_matrix для анализа ошибок классификации.

4. Визуализация результатов

1. **Графики:**
 - Сравнение accuracy моделей.
 - ROC-кривые для нескольких моделей.
 - Матрицы ошибок лучших моделей.

5. Анализ и выводы

1. Сравнить модели:
 - Какая модель показала наилучший результат?
 - Время обучения моделей.
 - Какие модели лучше справляются с данной задачей?
 - Влияют ли признаки на качество классификации?
2. Сделать вывод о применимости машинного обучения для выявления атак SQL Injection.