МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»
(Самарский университет)

Институт информатики, математики и электроники
Факультет информатики
Кафедра технической кибернетики

Отчет по лабораторной работе № 2

Дисциплина: “Databases in Enterprise Systems”

(«Корпоративные базы данных»)

Выполнила: Дубман Л. Б.

Группа: 6133-010402D

Самара 2024

СОДЕРЖАНИЕ

Задание на лабораторную работу №2.

1. Сформулировать задачу машинного обучения для БД из 1 лабораторной, которую собираетесь решать.

2. Разделить данные из БД на тренировочные и тестовые.

3. Определить вектор признаков для данных вашей предметной области. Самая сложная часть работы, от которой зависит дальнейшая эффективность машинного обучения, советую ориентироваться на существующие принципы выбора признаков и предварительной обработки данных в выбранной предметной области.

4. Выбрать тип классификатора: от самых простых kNN и линейного до нейронных сетей и RandomForest.

5. Оценить оптимальные значения гиперпараметров. Построить соответствующие зависимости качества распознавания от гиперпараметров.

6. Обучить классификатор на тренировочной выборке и оценить его эффективность на тестовой выборке.

Выполнение лабораторной работы

Выполнение лабораторной работы было разделено на 5 основных этапов:

Этап 1 - Подготовительный этап, на котором производим подключение библиотек, Google Drive, а также загрузку данных датасета из файла. В качестве источника данных был использован dataset данных, с сайта kaggle.com по ссылке https://www.kaggle.com/datasets/pritsheta/diabetes-dataset.

Этап 2 - Производится подготовка данных: выделение вектора признаков, разделение данных на тестовую и тренировочную выборки, а также нормализация полученных наборов.

Этап 3 - Производится выбор классификатора, при помощи которого и будем производить классификацию данных. В качестве классификаторов я использовала следующие:

- kNN - (k Nearest Neighbor или k Ближайших Соседей) - простейший алгоритм классификации, используемый в машинном обучения;
- DecisionTreeClassifier - алгоритм классификации, построенный на основе дерева решений;
- RandomForestClassifier - алгоритм классификации, построенный на основе ансамбля деревьев решений.

Этап 4 - Производится подбор оптимальных значений гиперпараметров, для выполнения данной операции использовалась функция GridSearchCV.

Этап 5 - Производится выбор классификатора с учетом гиперпараметров.

Этапы 3-5 выполнения лабораторной работы

Рассмотрим более подробно этапы 3-5, поскольку основная часть работы выполнена на данных этапах.

Этап 3

На данном этапе мы производим инициализацию и обучение классификатора с параметрами по умолчанию, после чего производим запуск классификатора. Например kNN:

```
classifierKNN = KNeighborsClassifier()
classifierKNN.fit(X_train_scaler, y_train)
classifierPredictionKNN = classifierKNN.predict(X_test_scaler)
```

Важным шагом на данном этапе является проверка точности работы классификатора, собственно по данному параметру мы и будем оценивать работу классификатора.

```
print("Accuracy KNeighborsClassifier:", accuracy_score(y_test, classifierPredictionKNN) * 100)
```

Этап 4

На данном этапе мы производим подбор оптимальных гиперпараметров при помощи GridSearchCV, а также инициализируем классификатор с оптимальными гиперпараметрами.

```
DecisionTreeParams = {
    "max_depth": range(1, 33),
    "min_samples_split": np.linspace(0.01, 0.1, 10, endpoint=True),
    "min_samples_leaf": np.linspace(0.01, 0.1, 10, endpoint=True),
}
DecisionTreeGSCV = GridSearchCV(classifierDTC, DecisionTreeParams)
```

Этап 5

На финальном этапе работы с классификаторами, запустим их с учетом подобранных оптимальных гиперпараметров, а так же посмотрим точность предсказания. Например:

```
RandomForestGSCV.fit(X_train_scaler, y_train)
print(RandomForestGSCV.best_estimator_)
RandomForestGSCV_Predict = RandomForestGSCV.predict(X_test_scaler)
print("Accuracy RandomForestClassifier with optimal hyperparameters:", accuracy_score(y_test, RandomForestGSCV_Predict) * 100)
```

Результаты выполнения лабораторной работы

Рассмотрим полученные результаты в ходе выполнения лабораторной работы. Для удобства изучения, они были сведены в таблицу.

	kNN	Decision Tree Classifier	Random Forest Classifier
Without hyperparameter tuning	71.75%	72.49%	75.09%
With hyperparameter tuning	75.09%	72.86%	75.84%

Нетрудно заметить, что полученные результаты совсем неоднозначны. Самым точным классификатором является RandomForestClassifier, несмотря на незначительный прирост 0.74% точности, произошедшую в процессе настройки гиперпараметров. Наибольший прирост точности показал классификатор kNN, который повысил точность на 3.35%, после настройки гиперпараметров.

Итоги выполнения лабораторной работы

Подводя итоги, можно отдать первое место RandomForestClassifier, второе место занимает kNN, на третьем месте DecisionTreeClassifier. Данная оценка носит чисто субъективный характер и зависит только от мнения автора. Так же хочется сказать немного про полученные результаты. Точность предсказаний классификатора зависит от многих факторов, таких как, процентное соотношение данных при разбиении выборки данных на тестовый и тренировочный пакеты, нормализация данных, подбор гиперпараметров, причем подбор гиперпараметров не всегда может гарантировать увеличение точности предсказаний классификатора. Код лабораторной работы представлен в репозитории GitHub, по ссылке: https://github.com/WonMin13/EnterpriseDataBase/blob/main/Lab%20Work%20%232/README.md