**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра ИС**

**ОТЧЕТ**

**по лабораторной работе №3**

**по дисциплине «Машинное обучение»**

**Тема: Исследование алгоритмов классификации**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 2372 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Мельникова М. А. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Татчина Я.А. |

Санкт-Петербург

2024

**Цель работы.**

Выбор набора данных, который будет использоваться в дальнейшем при исследовании алгоритмов кластеризации и классификации.

**Задание на работу.**

1. Необходимо оценить и сравнить результаты классификации, используя следующие алгоритмы классификации:

* kNN
* дерево решений

2. Сравните полученные результаты с помощью различных метрик оценки качества:

* Accuracy
* Presicion, Recall, F-measure
* ROC

3. Объяснить полученные результаты. Отчет должен включать описания выполнения каждой подзадачи.

**Выполнение работы.**

Исследование продолжим на наборе данных о студенческом образе жизни.

Атрибуты датасета: идентификатор студента, количество учебных часов в день, время дополнительной активности в день, время сна в день, время социальной активности в день, часы физической активности в день, учебный балл, уровень стресса.

Целевым классом для классификации выберем Stress\_Level, атрибут, показывающий уровень стресса студентов, выражающийся в трёх значениях "Low", "Moderate", "High". Оценим сбалансированность, получив количественное значение:

High 1029

Moderate 674

Low 297



Рис1. Диаграмма сбалансированности значений атрибута ‘Stress\_Level’

Видно, что значения встречаются неравномерно. Поэтому приведём их к сбалансированному количеству.

Применим модель k-ближайших соседей (kNN).

kNN – это алгоритм, который не строит явной модели на этапе обучения. Вместо этого он запоминает все обучающие данные. Когда поступает новый объект для классификации, kNN находит k ближайших соседей, вычисляет расстояния от нового объекта до всех объектов в обучающем наборе данных, определяет класс нового объекта, основываясь на классах его k ближайших соседей.

knn = KNeighborsClassifier(n\_neighbors=10)

knn.fit(X\_train\_scaled, y\_train)

y\_pred\_knn = knn.predict(X\_test\_scaled)

KNeighborsClassifier(n\_neighbors=10) создает экземпляр модели kNN с параметром n\_neighbors10. Это означает, что для классификации каждого нового объекта будут учитываться 10 ближайших к нему соседей.

knn.fit(X\_train\_scaled, y\_train) – обучение модели kNN.

knn.predict(X\_test\_scaled): Прогнозирование классов для тестовых объектов X\_test\_scaled. Алгоритм для каждого объекта находит 10 ближайших соседей из обучающих данных и на основе их классов выдает прогноз.

Применим модель дерева решений.

Дерево решений строит модель в виде дерева, где каждый внутренний узел представляет собой проверку значения одного из признаков, каждое ребро – результат этой проверки, а каждый листовой узел – предсказанное значение. Классификация нового объекта происходит путем прохода по дереву, начиная с корня, до тех пор, пока не будет достигнут лист.

tree = DecisionTreeClassifier(random\_state=42)

tree.fit(X\_train, y\_train)

y\_pred\_tree = tree.predict(X\_test)

DecisionTreeClassifier(random\_state=42) создает экземпляр модели дерева решений. Параметр random\_state обеспечивает воспроизводимость результатов при случайных выборах на этапе обучения дерева.

tree.fit(X\_train, y\_train) – обучение модели дерева решений. Строится дерево на основе данных, алгоритм выбирает признаки и значения разбиения таким образом, чтобы максимизировать разделение классов в обучающем наборе данных.

tree.predict(X\_test) – прогнозирование классов. Алгоритм для каждого тестового объекта проходит по дереву, начиная с корня, до достижения листового узла, который и является предсказанным классом.

Сравним полученные от разных моделей результаты с помощью следующих метрик оценки качества: Accuracy, Presicion, Recall, F-measure, ROC.

Результаты работы классификатора kNN:

Accuracy: 0.935

Precision: 0.9276993397470017

Recall: 0.9273894650286524

F1-score: 0.9264257481648785

ROC: 0.49208061495104083

Результаты работы классификатора Дерево решений:

Accuracy: 1.0

Precision: 1.0

Recall: 1.0

F1-score: 1.0

ROC: 1.0

 Модель дерева решений показала идеальные результаты по всем метрикам: Accuracy, Precision, Recall, F1-score – значение 1.0. То есть, после обучения, на данном дата сете она классифицирует все объекты абсолютно верно. Модель kNN также очень хорошие результаты при значении n\_neighbors=10. Accuracy показывает долю правильно классифицированных объектов из общего числа объектов в тестовой выборке. Модель kNN правильно классифицировала 93.5% всех объектов. Precision показывает долю правильно предсказанных положительных результатов среди всех объектов, которые модель предсказала как положительные. В данном случае kNN модель предсказывает в 92.8% случаев. Это означает, что модель редко выдает ложноположительные срабатывания. Recall показывает долю истинно положительных результатов среди всех объектов, которые на самом деле являются положительными. В данном случае модель обнаруживает 92.7% всех объектов, относящихся к положительному классу. Значит модель редко пропускает объекты положительного класса. F1-метрика — это гармоническое среднее точности и полноты. В данном случае 0.926 означает, что баланс между точностью и полнотой в этой модели достаточно хороший. ROC показывает, насколько хорошо модель разделяет классы. В данном случае ROC AUC составляет 0.492, что является очень низким значением.

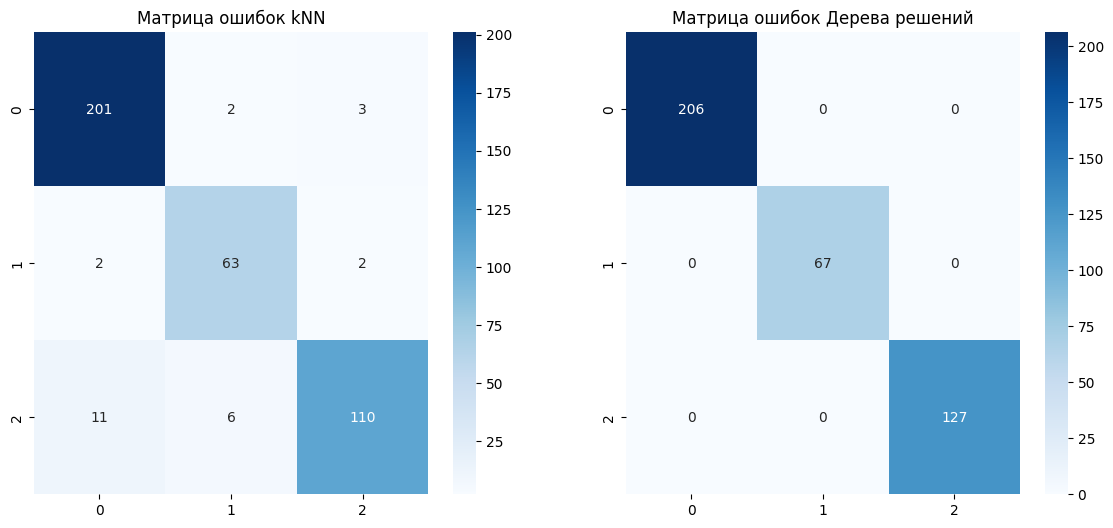
Для оценки количества верно классифицированных значений построим матрицу ошибок. Большие значения на главной диагонали свидетельствуют о том, что модели успешно справляются с прогнозом значений.

Рис2. Матрица ошибок

**Выводы.**

В ходе работы были изучены алгоритмы классификации KNN и дерево решений, полученные результаты были сравнены с помощью метрик оценки качества: Accuracy, Presicion, Recall, F-measure, ROC. Было выявлено, что на исследуемом датасете о студенческом образе жизни модель дерева решений показывает более высокую производительность, чем модель kNN.