УДК 004.896

ОСНОВНЫЕ ОПТИМИЗИРУЮЩИЕ МЕТОДОЛОГИИ FEATURES EXTRACTION ДЛЯ ПОВЫШЕНИЯ ТОЧНОСТИ МОДЕЛЕЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Кудаева И. В., студент группы ИТм-201, I курс
Ивина О.А., к.т.н., доцент
Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

Data Visualization – один из важнейших этапов в реализации прикладного программного решения на основе искусственного интеллекта. Извлечение признаков (features extraction) позволяет повысить точность моделей искусственного интеллекта ещё на этапе подготовки данных, существенно облегчая последующий процесс реализации и настройки программного алгоритма.

В современном мире всё более часто проявляется тенденция к обработке больших данных. В связи с этим, становится все более распространенным решение задач, которым свойственно оперирование огромными наборами исходных данных (такие данные также называют датасетами). Соизмеримо данным приумножается и число признаков, присутствующих в исходных датасетах [1].

Если число характеристик (или число столбцов) становится больше, чем количество наблюдений, хранящихся в наборе данных (число строк), то такая ситуация, скорее всего, может привести к переобучению модели искусственного интеллекта. Чтобы избежать проблем такого типа, необходимо применять методы регуляризации или уменьшения размерности (извлечение признаков). Например, в машинном обучении размерности набора данных отождествляются с количеством переменных, используемых для его представления.

Использование регуляризации, безусловно, может помочь снизить риск переобучения, но такой подход не всегда представляется возможным или удобным (может возникать очень сложная алгоритмическая сложность решения задачи). В таких ситуациях на помощь приходит использование методов извлечения признаков (от английского – features extraction, сокращенно: FE), которые обладают целым рядом существенных преимуществ:

- ✓ Позволяют улучшить итоговую точность модели;
- ✓ Заметно снижают риск перегрузки;
- ✓ Темпы обучения становятся существенно выше;
- ✓ Облегчается процесс визуализации данных;
- ✓ Увеличивается величина интерпретируемости модели (то есть критерий, который позволяет логически объяснить смысл модели).

Методология извлечения признаков (FE) снижает порядок объектов в наборе данных путем создания новых объектов из существующих (и

последующего исключения исходных объектов, которые несущественно влияют на итоговую точность модели). Этот новый сокращенный набор признаков затем может обобщать бóльшую часть информации, содержащейся в исходном наборе функций. Таким образом, обобщенная версия исходных функций может быть создана из комбинации исходного набора.

Другой, широко используемый метод уменьшения количества объектов в наборе данных – выборка отдельных объектов. Разница между выбором объектов и их извлечением заключается в том, что выборка объектов направлена на то, чтобы ранжировать приоритет существующих объектов в наборе данных и отбрасывать менее важные (новые объекты не создаются).Определим основные типы Feature Extraction:

- ✓ Принципиальный анализ компонентов (от английского: principal component analysis – сокращенно PCA)

PCA является одним из наиболее часто используемых методов линейного уменьшения размерности. При использовании метода PCA, в качестве входных данных принимаются исходные данные. После этого осуществляется поиск такой комбинации входных функций, которая могла бы наилучшим образом обобщить исходное распределение данных. Основная цель – уменьшение исходных размеров датасета. Принципиальный анализ компонентов может осуществлять такую операцию путем максимизации отклонений и минимизации ошибки восстановления, рассматривая парные расстояния. В PCA исходные данные проецируются на набор ортогональных осей, и каждая из осей ранжируется в порядке важности признаков [2].

Анализ компонентов – это неконтролируемый алгоритм обучения, поэтому его основной смысл состоит в поиске минимального количества вариаций признаков. Соответственно, следует помнить о том, что при таком подходе нужно самостоятельно разрешить проблему контроля меток размеченных данных (например, заданного класса признака), чтобы избежать неправильной классификации данных.

В этом примере рассмотрено использование методологии PCA, которая позволяет уменьшить исходный набор данных с трёх до двух измерений, а затем получить фрейм данных с новым набором признаков. Практическая реализация метода представлена на языке программирования Python [3], при помощи бесплатной, свободно распространяемой программной библиотеки машинного обучения sklearn. Листинг примера представлен на рисунке 1.

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)
PCA_df = pd.DataFrame(data = X_pca, columns = ['PC1', 'PC2'])
PCA_df = pd.concat([PCA_df, df['class']], axis = 1)
PCA_df['class'] = LabelEncoder().fit_transform(PCA_df['class'])
PCA_df.head()
```

Рисунок 1 – Программная реализация алгоритма PCA

Используя метод принципиального анализа компонентов, можно исследовать, какая часть исходной дисперсии данных была сохранена (это необходимо, чтобы сохранять адекватность представления данных). Для этого используется стандартная функция ***explained_variance_ratio_***.

Например, повторный запуск классификатора случайного леса (Random Forest Classifier) с использованием набора из 2 функций, построенных PCA (вместо всего набора данных), позволил получить точность классификации 98% по сравнению с использованием всего набора данных из 3 признаков (точность 88%). Результаты приведены на рисунках 2 – 3.

	precision	recall	f1-score
0	0.87	0.89	0.88
1	0.89	0.86	0.88
accuracy			0.88
macro avg	0.88	0.88	0.88
weighted avg	0.88	0.88	0.88

Рисунок 2 – Применение классификации на стандартном наборе.

	precision	recall	f1-score
0	0.97	0.99	0.98
1	0.99	0.96	0.98
accuracy			0.98
macro avg	0.98	0.98	0.98
weighted avg	0.98	0.98	0.98

Рисунок 3 – Применение набора с извлечёнными признаками методом PCA

- ✓ Независимый компонентный анализ (от английского Independent Component Analysis – ICA)

ICA – это метод линейного уменьшения размерности, в качестве входных данных которого принимается общность независимых компонентов. Методология стремится правильно идентифицировать каждый из признаков по отношению к итоговой точности (удаляя все малозначительные). Две входные характеристики можно считать независимыми, если их линейная и нелинейная зависимость равны нулю. Независимый анализ обычно используется в медицинских приложениях, таких как анализ ЭЭГ и МРТ, для различения полезных сигналов от бесполезных [4].

В качестве простого примера приложения ICA рассмотрим аудиозапись, в которой разговаривают два разных человека. Используя ICA, например, можно идентифицировать два разных независимых компонента (два разных голоса).

Используя ICA, снова уменьшим набор исходных данных до трех функций (непосредственный набор состоит из пяти признаков), проведем проверку повышения точности с помощью классификатора случайного леса и отобразим результаты (рисунки 4 – 5)

Classification Report:

	precision	recall	f1-score	support
0	0.87	1.00	0.93	75
1	1.00	0.56	0.72	25
accuracy			0.89	100
macro avg	0.94	0.78	0.82	100
weighted avg	0.90	0.89	0.88	100

Metrics:

	Random Forest B
Accuracy	0.890000
Precision	1.000000
Recall	0.560000
F1-score	0.717949
F beta-score	0.560044
ROC AUC	0.964267

Рисунок 4 – Получение сравнительных результатов от исходного набора данных

```
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.99      0.94        75
           1       0.94      0.68      0.79        25

    accuracy                           0.91       100
   macro avg       0.92      0.83      0.87       100
weighted avg       0.91      0.91      0.90       100


Metrics:
              Random Forest A
Accuracy             0.910000
Precision            0.944444
Recall               0.680000
F1-score             0.790698
F beta-score         0.680034
ROC AUC              0.983467
```

Рисунок 5 – Получение сравнительных результатов от внедрения ICA к исходному набору данных

Не всегда результаты заметны существенным образом: в данном случае точность по обобщающей метрике точности f1-score повысилась с 0,717949 до 0,790698.

Однако, во многих задачах благоприятно любое увеличение точности, поэтому повышение даже сотых долей такими простыми средствами – значительный результат в разработку модели.

- ✓ Линейный Дискриминантный Анализ (от английского Linear Discriminant Analysis – LDA)

LDA – это не только методика уменьшения размерности датасета, но также и непосредственно один из классификаторов в машинном обучении. LDA направлена на максимизацию расстояния между средними значениями каждого класса и минимизации распределения данных внутри самого класса. Именно поэтому методика дискриминантного анализа используется в классах и между классами как своеобразные мера регулирования расстояния. Такой подход позволяет определить максимальное расстояние между средними для каждого класса при проецировании данных в пространство меньшего размера, благодаря чему достигаются лучшие результаты в итоговой классификации. При использовании LDA предполагается, что входные данные соответствуют распределению Гаусса, поэтому применение метода к данным, имеющим негауссово распределение может привести к плохим результатам общей классификации [5].

В рассматриваемом примере данные имеют нормальное (Гауссово) распределение, и методология LDA позволяет концентрировать набор данных до размерности двух признаков. Изучая точность итоговых результатов, заметим, что правильный выбор метода (данные имеют необходимое распределение) во многом позволяет повысить итоговую точность даже до 100% (на тестовых данных). Результат представлен на рисунке 6.

```
1.2756952610000099
[[1274    0]
 [   0 1164]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1274
           1       1.00      1.00      1.00      1164

    accuracy                           1.00      2438
   macro avg       1.00      1.00      1.00      2438
weighted avg       1.00      1.00      1.00      2438
```

Рисунок 6 – Точность алгоритма с применением LDA-методологии

Описанная концепция также не стала исключением для повышения точности алгоритма. Детальное исследование предметной области и визуализация данных во многом позволяют облегчить понимание главного смысла получаемых для решения данных, способствуют более быстрому определению закономерностей в информации. Features extraction существенным образом предопределяет результат производительности алгоритма машинного обучения, на данном этапе точность во многом зависит от особенностей математического и статистического распределения используемых в задаче данных. Верно определённые закономерности помогают определить результат и получить максимально возможную эффективность модели искусственного интеллекта, создавая качественную основу перед непосредственным этапом разработки, обучения и тестирования модели искусственного интеллекта.

Список литературы:

1. David Julian. Designing Machine Learning Systems with Python – Packt Publishing. 2016. – 209 с
2. Ивина О. А. Программирование массивов с помощью numpy //Донецкие чтения 2020: образование, наука, инновации, культура и вызовы современности. – 2020. – С. 220-222.
3. Samir Madhavan. Mastering Python for Data Science – Packt Publishing. 2018. – 276 с
4. Mark J. Johnson. A Concise Introduction to Programming in Python Second Edition. – CRC Press. 2018. – 197 с
5. David Kopec. Classic Computer Science Problems in Python. – Manning Shelter Island. 2019. – 201 с