Понимание путаницы матрицы в Python

Эй, ребята! Сегодня мы посмотрим на одну из самых важных концепций науки о данных – путаницу матрицы и его реализации в Python.

Автор: **Pankaj Kumar <** https://pythobyte.com/author/5173065252783846647/>

09.04.2021 < https://pythobyte.com/confusion-matrix-62369a52/>

Автор оригинала: Pankaj Kumar < https://www.askpython.com/python/examples/confusion-matrix> .

Понимание путаницы матрицы в Python

Эй, ребята! Сегодня мы посмотрим на одну из самых важных концепций науки о данных – **Путаница матрицы и его реализация в Python** Отказ

Наука данных является огромным доменом, в котором мы изучаем данные, очистите данные и выполняем прогнозы, используя различные алгоритмы машинного обучения. После того, как использовал любой алгоритм, для нас очень важно проверить точность и влияние алгоритма вклад вклад желаемого выхода.

Одной из таких ошибок метрики для оценки производительности модели является путаница матрицы.

Что такое путаница матрицы?

Время от времени мы сталкиваемся с ситуациями, когда нам нужно применить определенные алгоритмы ML для прогнозирования результатов для `Задача классификации` I.E. Бизнес-проблемы, в которых исход/переменная целевой/ответной перемены является

категорические данные Отказ Пример: обнаружить, является ли электронная почта SPAM или Not-Spam.

Итак, в приведенном выше сценарии нам нужна специальная метрика ошибок для оценки точности и точности модели для лучшего приспособления.

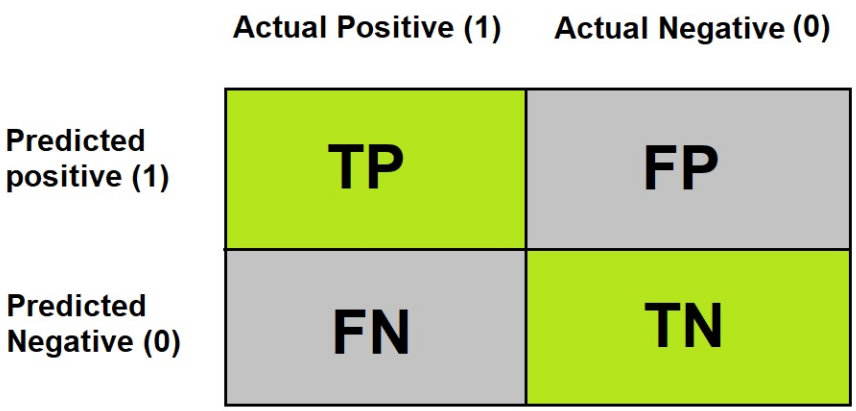
Путаница Matrix – это **Ошибка метрики** , это используется для оценки производительности алгоритмов обучения машины классификации. Он предоставляет нам подробную информацию о уровне точности, точной скорости и процентах ошибок модели.

Используя путаницу Matrix, мы можем различить фактический правильный и прогнозируемый результат категорической переменной реагирования.

Итак, поняв необходимость путаницы Matrix, давайте теперь сосредоточимся на различных компонентах, через которые мы можем судить и предсказать правильный алгоритм посадки для любой модели.

Компоненты путаницы матрицы

Посмотрите на нижестоящую структуру путаницы Matrix!



Он представляет собой краткое изложение прогнозов, выполненных классификационными моделями.

- **Истинный отрицательный (тн)** : Значения, которые на самом деле негативны, а также предсказаны как отрицательные.
- **Ложно отрицательный (FN)** : Значения, которые на самом деле являются позитивными, но предсказаны как отрицательные.
- **Ложный положительный (FP)** : Значения, которые на самом деле негативны, но предсказаны как позитивные.
- **Истинный положительный (TP)** : Значения, которые на самом деле являются позитивными и предсказаны как положительны.

Итак, теперь давайте посмотрим на другую информацию о том, что путаница Matrix доставляет о модели

1. Точность – Он определяется как значение, которое показывает процент успешного прогнозирования от приведенного ввода.

Точность + TN/(TP + TN + FP + FN)

2. Точность балла – Это значение, которое определяет набор значений, которые правильно предсказаны как истинные, а также происходит, чтобы быть верным в фактическом наборе.

По точному точности мы хотим понимать, что положительные ценности действительно предсказывают как позитивные.

Точность/(TP + FP)

3. Напомним счет – Это значение, которое представляет собой набор значений, которые на самом деле верны, а также правильно предсказаны так же, как правильно,

Напомним, мы хотим понимать, что конкретный класс образцов правильно предсказан.

Напомним/(TP + FN)

4. Оценка F1

Оценка F1 помогает нам оценить точность и эффективность модели, когда данные неразбавлены. Это на самом деле гармоническое среднее значение точности и результатов отзыва.

F1 * (отзыв * точность)/(отзыв + точность)

Давайте теперь реализуем концепцию путаницы матрицы через пример, как показано в предстоящем разделе.

Реализация путаницы матрицы в Python

В этом примере мы передали список прогнозируемых значений и фактические значения для создания матрицы путаницы. Нам нужно импортировать библиотеку Sklearn, чтобы использовать функцию Matrix Confusion.

```
from sklearn import metrics

pred = ["T", "F", "T", "T", "F"] #predicted set of values

actual = ["F", "F", "F", "T", "T"] #actual set of values
CM = metrics.confusion_matrix(pred, actual, labels=["T", "F"]

print(CM)
report = metrics.classification_report(pred, actual, labels=[
print(report)
```

`Классификация_Matrix () Функция` Представляет набор значений, которые были правильно и неправильно идентифицированы. Далее `Классификация_Report () Функция` Представляет значение метрик для каждой категории входов прошло я. «T» и «F».

Выход:

```
[[1 2]
 [1 1]]
              precision    recall  f1-score   support

           T       0.50      0.33      0.40         3
           F       0.33      0.50      0.40         2

    accuracy                           0.40         5
   macro avg       0.42      0.42      0.40         5
weighted avg       0.43      0.40      0.40         5
```

Заключение

По этому, мы подошли к концу этой темы. Не стесняйтесь комментировать ниже, если вы столкнетесь с любым вопросом.

До этого, счастливое обучение!

Читайте Ещё По Теме:

- **NumPy ogrid и его использование в Python < https://pythobyte.com/numpy-ogrid-00060/>**
- **Редкая матрица в Python – упрощенная < https://pythobyte.com/sparse-matrix-9b193d51/>**