Студент: Тимур Хабибуллин
Дата: 7 ноября 2021 г.

Будем анализировать код с помощью PyLint, Bandit, autopep 8.

1. **PyLint**
   Данный инструмент был выбран потому, что он ищет логические и стилистические ошибки, что очень удобно, потому что не надо использовать несколько линтеров- отдельно для логических и отдельно стилистических.

   Файл a-statistical-analysis-ml-workflow-of-titanic.py очень большой (так получилось), поэтому PyLint сделал очень много замечаний.

   Общая оценка кода -5,91/10 (она отрицательная!)
   trailing whitespace - 456 раз - в файле очень много пустых закоментированных строк
   line-too-long - 229 раз - очень длинные строки
   invalid-name - 45 раз - нужны названия в стиле snake_case
   unnecessary-semicolon - 21 раз - лишние точка с запятой
   superfluous-parens - 6 раз - почему-то после if elif условие находится в скобках
   reimported - 14 раз - одни и те же библиотеки перевызываются несколько раз.
   В общем, очень много недочетов.

   Настройки: при искользовании я искользовал ключик -ry для того, чтобы получить полный отчет от PyLint.

2. **Bandit**
   Этот инструмент позволяет проанализировать безопасность кода и найти уязвимые места. При запуске использовал ключ -r для вывода полного отчета.
   Bandit сказал, что безопасность кода в порядке.

3. **autopep 8**
   Данный автоформаттер позволяет привести код к формату pep8. В результате его работы (autopep8 –aggressive) в коде удалились ненужные пробелы (например, было print (...), а стало print(...)), но при этом добавились новые - между знаками сложения, умножения и тд; собрались вместе все импорты библиотек, а так же добавились переносы при перечислении аргументов функций. Таким образом, код стал более читаемым.
   Еще можно заметить, что после этого инструмента, при анализе кода с помощью PyLint, общая оценка кода стала 3.99/10 (число уже положительное!). В качестве настроек можно изменять "уровень агрессии тогда автоформаттер будет находить больше недочетов и вностить изменений.