РАБОТА С ЛИНЕЙНОЙ РЕГРЕССИЕЙ

ПРЕДИКТИВНЫЙ АНАЛИЗ НА ОСНОВЕ ЦЕЛЕВОЙ ПЕРЕМЕННОЙ

Задание основано на применении пакета Python Scikit-learn для извлечения TF-IDF-признаков из текстов и предсказания целевой переменной на основе гребневой регрессии

Цели практики:

- овладеть основами применения линейной регрессии
- уметь применять линейную регрессию к текстовым данным

Начало работы

Для извлечения **TF-IDF**-признаков из текстов воспользуйтесь классом **sklearn.feature_extraction.text.TfidfVectorizer.**

Для предсказания целевой переменной применяется гребневая регрессия, которая реализована в классе **sklearn.linear_model.Ridg**e.

Обратите внимание, что признаки **LocationNormalized** и **ContractTime** представляют собой строки, поэтому работать с ними напрямую невозможно. Эти нечисловые признаки с неупорядоченными значениями обычно называются *категориальными или номинальными*. Один из распространенных методов их обработки заключается в кодировании категориального признака с **m** возможными значениями при помощи **m** бинарных признаков.

Каждый бинарный признак соответствует одному из возможных значений категориального признака и служит индикатором того, принимает ли данный объект данное значение. Этот метод часто называют **one-hot-кодированием**. Его необходимо применить для перекодирования признаков **LocationNormalized** и **ContractTime**.

Пример использования:

```
from sklearn.feature_extraction import DictVectorizer
enc = DictVectorizer()
X_train_categ = enc.fit_transform(
    data_train[['LocationNormalized', 'ContractTime']]
        .to_dict('records'))
X_test_categ = enc.transform(
    data_test[['LocationNormalized', 'ContractTime']]
        .to_dict('records'))
```

Также для решения Вам понадобится производить замену пропущенных значений на специальные строковые величины (например, **'nan'**). Для этого подходит следующий код:

```
data_train['LocationNormalized'].fillna('nan', inplace=True)
data_train['ContractTime'].fillna('nan', inplace=True)
```

Задание

1. Загрузите данные об описаниях вакансий и соответствующих годовых зарплатах из файла **salary-train.csv.**

2. Проведите предобработку:

- Приведите тексты к нижнему регистру.
- Замените все, кроме букв и цифр, на пробелы — это облегчит дальнейшее разделение текста на слова. Для такой замены в строке ***text*** подходит следующий вызов:

```
re.sub('[^a–zA–Z0–9]', ' ', text.lower())
```

- Примените **TfidfVectorizer** для преобразования текстов в векторы признаков. Оставьте только те слова, которые встречаются хотя бы в 5 объектах (параметр **min_df** у **TfidfVectorizer**).
- Замените пропуски в столбцах **LocationNormalized** и **ContractTime** на специальную строку **'nan'**. Код для этого был приведен выше.
- Примените **DictVectorizer** для получения **one-hot-кодирования** признаков **LocationNormalized** и **ContractTime.**
- Объедините все полученные признаки в одну матрицу "объекты-признаки". Обратите внимание, что матрицы для текстов и категориальных признаков являются разреженными. Для объединения их столбцов нужно воспользоваться функцией **scipy.sparse.hstack.**

3. Обучите гребневую регрессию с параметром **alpha=1**. Целевая переменная записана в столбце **SalaryNormalized**.

4. Постройте прогнозы для двух примеров из файла **salary-test-mini.csv**.

Значения полученных прогнозов являются ответом на задание. Укажите их через пробел.

Ответ на каждое задание — текстовый файл, содержащий ответ в первой строчке (#.txt). Обратите внимание, что отправляемые файлы не должны содержать перевод строки в конце.

Утонения по выполнению задания:

Если ответом является нецелое число, то целую и дробную часть необходимо разграничивать точкой, например, 0.42.

При необходимости округляйте дробную часть до двух знаков

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

Линейные методы отлично подходят для обработки данных с низкой плотностью, таких как текстовые данные. Это обусловлено высокой скоростью обучения и ограниченным числом параметров, что позволяет избежать переобучения.

В зависимости от используемого регуляризатора линейная регрессия может принимать различные формы.

В данном задании мы рассматриваем гребневую регрессию, где применяется квадратичный (L2) регуляризатор.