Machine Learning in Economics

Assignment 1

Цель задания: разработать функцию автозаполнения места выдачи паспорта по вводу кода подразделения и дате выдачи паспорта.

Оценка экономической выгоды от использования автозаполнения:

Каждый год в системе компании регистрируется 100 тыс. новых пользователей. Оператор в среднем тратит 30 секунд на ввод данных паспорта одного клиента: код подразделения, дата выдачи, место выдачи. Зарплата оператора - 0.05 евро/минута. В год оператор тратит 50 тыс. минут в год на ввод данных, это работа оценивается в 2500 евро в год.

Анализ данных показывает, что в среднем за эти 30 секунд оператор набирает 80 символов, из них 66 символа необходимы для ввода данных о месте выдачи паспорта.*

Введение функции автозаполнения места выдачи паспорта, позволит уменьшить траты времени на ввод данных: оператор вместо 30 секунд (80 символов) будет тратить 5.25 секунд (12 символов) на одного клиента. В этом случае данная работа оператора будет оцениваться в 437.5 евро в год.

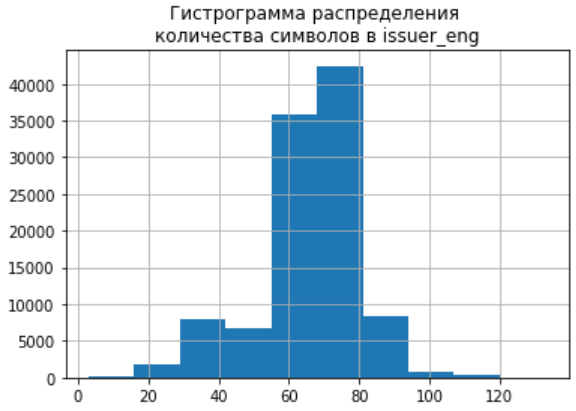
Введение функции автозаполнения позволит оптимизировать время сотрудника и уменьшить денежные издержки данной работы оператора в 5.7 раз.

* Оценка среднего количества символов, которые требуется для ввода информации одного клиента - 80 символов:

dept_code - 6 символов

issuer_date - 8 символов

issuer_eng - 66 символов (среднее)



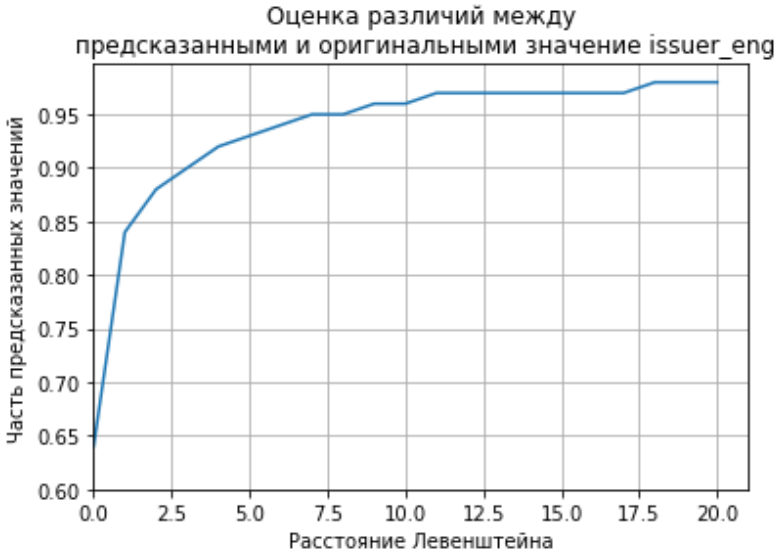
Код представлен на GitHub по ссылке:
https://github.com/assavkina/ML-in-Econometrics

Реализация функции автозаполнения

Данные были поделены на тренировочную и тестовую выборку. Значение issuer_eng из тестовой выборки по значениям dept_code и issuer_date предсказывалось по данным тренировочной выборки (train). В train отбирались все строки с данными значением dept_code. Из этих строк выбиралась единственная, в которое значение даты выдачи паспорта была ближайшей к данному issuer_date. Из выбранной строки значение issuer_eng становилось нашим автозаполнением.

Точность предсказанных значений мы определяем с помощью расстояния Левенштейна: сравниваются две строчки и, чем меньше это расстояние, тем больше похожи строчки. В случае сравнения двух одинаковых строчек расстояние Левенштейна равно 0.

График показывает, что доля предсказанных значений, которые полностью похожи на оригинальные значения, около 65 %. Количество предсказанных значений, для которых расстояние Левенштейна равно или меньше 7, быстро увеличивается до 95 %. При больших значениях расстояния Левенштейна доля предсказанных значений растет медленно, это может говорить нам, что только около 5 % всех предсказанных значений значительно не совпадают с оригинальными значениями issuer_eng.



Кластеризация значений issuer_eng

В тестовой выборке мы имеем 3367 уникальных значений issuer_eng. Многие из них отличаются только опечаткой, сокращением слова или наличием лишнего пробела. Вместо того, чтобы вручную редактировать эти ошибки, мы используем метод кластеризации для уникальных значений, который позволит устранить эти ошибки: схожие уникальные значения issuer_eng (элементы одного класса) будут заменяться на единственное значение issuer_eng (центр каждого класса).

В тренировочной выборке для каждого кода подразделения (dept_code) были получены уникальные значения места выдачи паспорта (issuer_eng). Для элементов из данного списка уникальных значений была составлена симметричная матрица попарных расстояний Левенштейна между ними. Эта матрица подавалась на вход методу кластеризации Affinity Propagation. В результате обученный метод распределял все уникальные значения issuer_eng для одного dept_code на классы и определял их центры. В процессе постобработки результатов метода Affinity Propagation был сформирован новый столбец Examplar в тренировочной выборке, его элементы связывают каждое уникальное описание issuer_eng с центром его кластера.

В результате такого способа чистки данных у нас осталось 803 уникальных значений issuer_eng в тренировочной выборке.

Оценка работы функции автозаполнения на очищенных данных

При повторном использовании функции автозаполнения мы использовали такой же алгоритм как и в первый раз, но вместо issuer_eng предсказанным значением теперь являлся элемент из столбца examplar.

График, описывающий точность наших предсказаний, показывает, что стало хуже. Если в первый раз доля точных предсказания была около 65%, то в данной случае она около 10%. Возможно это связано с тем, что мы очистили issuer_eng только в тренировочной выборки, и оставшиеся issuer_eng значительно отличается от того, что есть в тренировочной выборке. Хотя это маловероятно, потому что разделение данных на test и train было произвольно. Вторым объяснением может быть недостаточно хорошая кластеризация данных. Для подробного изучения этого вопроса нужно протестировать кластеризацию методом AffinitiPropagation с разными параметрами и выбрать оптимальный вариант.

Вопросы по заданию:

1. Мы проводили кластеризацию только по данным из тренировочной выборки (70% всех данных). В процессе кластеризации для нескольких dept_code метод Affinity Propagation не нашел центры. Когда мы провели кластеризацию по всем данным (100 % данных), для всех dept_code уникальные значения issuer_eng разбились на классы. Можно сделать вывод, что существуют ограничений на данные, которые подаются на вход методу. Какие?
2. В теории казалось, что после очистки данных автозаполнение должно работать лучше. Почему это не произошло в нашем случае? Это связано с тем, что мы почистили только тренировочную выборку?

Будем рады услышать обратную связь по решению задания.