

## Цель работы

Определить простую задачу машинного обучения и решить ее.

## Краткая теория

Apache Spark MLlib используется для создания приложения машинного обучения. Приложение выполняет прогнозный анализ на открытом наборе данных. MLlib — это основная библиотека Spark, которая предоставляет множество служебных программ, полезных для задач машинного обучения, таких как:

1. Классификация;
2. Регрессия;
3. Кластеризация;
4. Моделирование сингулярного разложения и анализа по методу главных компонент;
5. Проверки гипотез и статистической выборки.

## Общие сведения о выбранном алгоритме машинного обучения

Linear regression (с англ. — «Линейная регрессия») — способ выбрать из семейства функций ту, которая минимизирует функцию потерь. Последняя характеризует насколько сильно пробная функция отклоняется от значений в заданных точках.

Используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной  $y$  от другой или нескольких других переменных (факторов, регрессоров, предикторов, независимых переменных)  $x$  с линейной функцией зависимости, имеет следующий вид:

$$f(x,b)=b_0+b_1 x_1+b_2 x_2+...+b_k x_k$$

где  $b_j$  — параметры (коэффициенты) регрессии,  $x_j$  — регрессоры (факторы модели),  $k$  — количество факторов модели.

Коэффициенты линейной регрессии показывают скорость изменения зависимой переменной по данному фактору, при фиксированных остальных факторах (в линейной модели эта скорость постоянна).

При выполнении классических предположений обычный метод наименьших квадратов позволяет получить достаточно качественные оценки параметров модели, а именно: они являются несмещёнными, состоятельными и наиболее эффективными оценками.

## Ход работы

В процессе работы мы рассмотрим набор данных, состоящий из данных о средней продолжительности жизни и смертности. Сперва мы подключим контекст Spark, а также укажем в качестве dataframe, описанный выше набор данных. Первые 20 строк набора показаны на рисунке 1.

Year	Race	Sex	Average	Death_Rate
2014	All Races	Both Sexes	78.9	724.6
2013	All Races	Both Sexes	78.8	731.9
2012	All Races	Both Sexes	78.8	732.8
2011	All Races	Both Sexes	78.7	741.3
2010	All Races	Both Sexes	78.7	747.0
2009	All Races	Both Sexes	78.5	749.6
2008	All Races	Both Sexes	78.2	774.9
2007	All Races	Both Sexes	78.1	775.3
2006	All Races	Both Sexes	77.8	791.8
2005	All Races	Both Sexes	77.6	815.0
2004	All Races	Both Sexes	77.5	813.7
2003	All Races	Both Sexes	77.6	843.5
2002	All Races	Both Sexes	77.0	855.9
2001	All Races	Both Sexes	77.0	858.8
2000	All Races	Both Sexes	76.8	869.0
1999	All Races	Both Sexes	76.7	875.6
1998	All Races	Both Sexes	76.7	870.1
1997	All Races	Both Sexes	76.5	877.7
1996	All Races	Both Sexes	76.1	893.7
1995	All Races	Both Sexes	75.8	909.5

Рисунок 1. Топ 20 строк набора данных

Поставим задачу предсказать среднюю продолжительность жизни. Дополнительно преобразовывать столбец метки нет необходимости, просто переименуем его в label.

В данной работе планируется использовать алгоритм машинного обучения – Линейная регрессия, для которого требуется сперва обозначить столбцы, которые будут использоваться в качестве функций.

В процессе работы у нас появляются трудности с объединением данных, поэтому мы используем VectorAssembler – это преобразователь, который объединяет заданный список столбцов в один векторный столбец. Это полезно для объединения необработанных функций и функций, созданных различными преобразователями функций, в один вектор функций.

Работать напрямую с данными, хоть и в случае моего небольшого набора данных, не составляет труда, но для удобства обращения воспользуемся StringIndexer, который кодирует строковый столбец меток в столбец индексов меток. Также введем столбец features, который будет агрегировать значения Год, Пол, Раса и Средняя смертность. Результаты представлены на рисунке 2.

Year	Race	Sex	label	Death_Rate	indexSex	indexRace	features
2014.0	All Races	Both Sexes	78.9	724.6	2.0	1.0	[2014.0,724.5999755859375,2.0,1.0]
2013.0	All Races	Both Sexes	78.8	731.9	2.0	1.0	[2013.0,731.9000244140625,2.0,1.0]
2012.0	All Races	Both Sexes	78.8	732.8	2.0	1.0	[2012.0,732.7999877929688,2.0,1.0]
2011.0	All Races	Both Sexes	78.7	741.3	2.0	1.0	[2011.0,741.2999877929688,2.0,1.0]
2010.0	All Races	Both Sexes	78.7	747.0	2.0	1.0	[2010.0,747.0,2.0,1.0]
2009.0	All Races	Both Sexes	78.5	749.6	2.0	1.0	[2009.0,749.5999755859375,2.0,1.0]
2008.0	All Races	Both Sexes	78.2	774.9	2.0	1.0	[2008.0,774.9000244140625,2.0,1.0]
2007.0	All Races	Both Sexes	78.1	775.3	2.0	1.0	[2007.0,775.2999877929688,2.0,1.0]
2006.0	All Races	Both Sexes	77.8	791.8	2.0	1.0	[2006.0,791.7999877929688,2.0,1.0]
2005.0	All Races	Both Sexes	77.6	815.0	2.0	1.0	[2005.0,815.0,2.0,1.0]
2004.0	All Races	Both Sexes	77.5	813.7	2.0	1.0	[2004.0,813.7000122070312,2.0,1.0]
2003.0	All Races	Both Sexes	77.6	843.5	2.0	1.0	[2003.0,843.5,2.0,1.0]
2002.0	All Races	Both Sexes	77.0	855.9	2.0	1.0	[2002.0,855.9000244140625,2.0,1.0]
2001.0	All Races	Both Sexes	77.0	858.8	2.0	1.0	[2001.0,858.7999877929688,2.0,1.0]
2000.0	All Races	Both Sexes	76.8	869.0	2.0	1.0	[2000.0,869.0,2.0,1.0]
1999.0	All Races	Both Sexes	76.7	875.6	2.0	1.0	[1999.0,875.5999755859375,2.0,1.0]
1998.0	All Races	Both Sexes	76.7	870.1	2.0	1.0	[1998.0,870.0999755859375,2.0,1.0]
1997.0	All Races	Both Sexes	76.5	877.7	2.0	1.0	[1997.0,877.7000122070312,2.0,1.0]
1996.0	All Races	Both Sexes	76.1	893.7	2.0	1.0	[1996.0,893.7000122070312,2.0,1.0]
1995.0	All Races	Both Sexes	75.8	909.5	2.0	1.0	[1995.0,909.5,2.0,1.0]

Рисунок 2. Результаты агрегирования данных

Определим данные, которые мы ищем, дополнительно переименовав значения со стандартами Spark MLib. Ввиду наличия большого объема данных, нам представляется возможным разбить их на более мелкие части. Таким образом, мы подготовим данные для случайного леса.

Определим изначальное количество данных в наборе, а также число в обучающую выборку и в тестовую. Результаты продемонстрируем на рисунке 3.

```
dataframe count: 1035
training count: 717
test count: 318
```

Рисунок 3. Число данных по выборкам

Зададим необходимые параметры для обучения и поддержания точности на уровне, указанном в цели работы. Часть полученных результатов отобразим на рисунке 4.

Year	Death_Rate	indexSex	indexRace	label	prediction
1900.0	2410.4	1.0	1.0	48.3	50.662331200541956
1900.0	2630.8	0.0	1.0	46.3	47.389927607934915
1900.0	3423.3	2.0	0.0	33.0	33.46996837318214
1901.0	3167.2	1.0	0.0	35.3	37.86096527144801
1901.0	2334.7	1.0	2.0	51.0	52.55668804112707
1902.0	2430.1	0.0	2.0	50.2	51.30470878248581
1903.0	2250.6	1.0	1.0	52.0	53.26878563160394
1903.0	2513.5	0.0	1.0	49.1	49.3140920179041
1903.0	2231.5	1.0	2.0	52.5	54.24081238474067
1903.0	2494.2	0.0	2.0	49.5	50.28932878216955
1905.0	3654.7	0.0	0.0	29.6	30.35525708934601
1905.0	2404.1	2.0	2.0	49.1	51.23135475515645
1905.0	2544.7	0.0	2.0	47.6	49.50595794124584
1906.0	2244.6	1.0	1.0	50.8	53.406141832277996
1906.0	3341.0	0.0	0.0	31.8	35.405065735336414
1907.0	2494.4	2.0	1.0	47.6	49.14364458043795
1907.0	2660.3	0.0	1.0	45.6	47.01207749391636
1907.0	3408.1	0.0	0.0	31.1	34.34151989037946
1908.0	2963.6	1.0	0.0	36.0	41.2252943948384
1909.0	2111.5	1.0	1.0	53.8	55.58395983225553

Рисунок 4. Результаты предсказаний

Таким образом, наша модель может по входным признакам (Год, Пол, Раса, Средняя смертность) предсказать среднюю продолжительность жизни. Столбец Prediction содержит ответ модели. В целом, мы видим, что ответы модели приблизительно равны верным результатам.

Посмотрим отдельно на отклонения ответа модели и правильного.

residuals
-1.3690278133513587
-2.0868873474738905
0.957710094394919
-2.0041304028785945
-2.891008322505357
-1.7378749243439557
-0.3035279392932182
-1.0376417154892579
-0.2900428124351677
-1.1466298955152325
-0.1859563200367873
-0.9402336794012953
-0.8476171104725267
-0.6752775645914824
-1.2646464507675574
-0.5535576631939207
-2.6362979449338866
-4.1042830961777526
-1.5063798910631192
-1.2135878764290666

Рисунок 5. Различия ответов модели



В таблице мы видим, что в среднем, ответ модели отличается от истинного на 1-4 года. Для оценки качества модели, используем стандартные метрики качества MSE, RMSE, R2.

```
RMSE: 1.8088440204545106
MSE: 3.271916690334038
r2: 0.9772914689526722
```

Рисунок 6. Метрики качества

Самая легко интерпретируемая метрика MSE (Mean squared error). В целом можно интерпретировать как дисперсию отклонений ответов и предсказаний модели. Таким образом, можно утверждать, что в подавляющем большинстве случаев (~65%) ответы модели отличаются от истинных меньше чем на 3 года.

Попробуем улучшить качество модели нормализовав входные данные.

features	normFeatures
[2014.0, 724.5999755859375, 2.0, 1.0]	[0.9409523991456148, 0.3385372817519572, 9.344115185160028E-4, 4.672057592580014E-4]
[2013.0, 731.9000244140625, 2.0, 1.0]	[0.9398079011561438, 0.34170165216130677, 9.337386002544896E-4, 4.668693001272448E-4]
[2012.0, 732.7999877929688, 2.0, 1.0]	[0.9396182592338743, 0.3422227877219849, 9.340141741887418E-4, 4.670070870943709E-4]
[2011.0, 741.2999877929688, 2.0, 1.0]	[0.9382812557490945, 0.34587164765448786, 9.331489365978065E-4, 4.6657446829890327E-4]
[2010.0, 747.0, 2.0, 1.0]	[0.9373595106815106, 0.34836196740253156, 9.326960305288663E-4, 4.6634801526443316E-4]
[2009.0, 749.5999755859375, 2.0, 1.0]	[0.9369061907482655, 0.3495793219070252, 9.327090002471534E-4, 4.663545001235767E-4]
[2008.0, 774.9000244140625, 2.0, 1.0]	[0.9329409670763558, 0.36002787757188603, 9.29224070793183E-4, 4.646120353965915E-4]
[2007.0, 775.2999877929688, 2.0, 1.0]	[0.9328182276736046, 0.36034577006896085, 9.295647510449473E-4, 4.6478237552247364E-4]
[2006.0, 791.7999877929688, 2.0, 1.0]	[0.9301613909293404, 0.36714944066966226, 9.273792531698309E-4, 4.6368962658491545E-4]
[2005.0, 815.0, 2.0, 1.0]	[0.9263905419402576, 0.37656273899317205, 9.240803410875387E-4, 4.6204017054376936E-4]
[2004.0, 813.7000122070312, 2.0, 1.0]	[0.9265345828643372, 0.3762081843248242, 9.246852124394582E-4, 4.623426062197291E-4]
[2003.0, 843.5, 2.0, 1.0]	[0.921613233372831, 0.38810821884672136, 9.202328840467608E-4, 4.601164420233804E-4]
[2002.0, 855.9000244140625, 2.0, 1.0]	[0.9194933968659728, 0.3931041063067709, 9.185748220439288E-4, 4.592874110219644E-4]
[2001.0, 858.7999877929688, 2.0, 1.0]	[0.9189401410100773, 0.39439569309441447, 9.184809005597974E-4, 4.592404502798987E-4]
[2000.0, 869.0, 2.0, 1.0]	[0.9171643267080055, 0.39850789995462843, 9.171643267080055E-4, 4.5858216335400276E-4]
[1999.0, 875.5999755859375, 2.0, 1.0]	[0.9159821968252876, 0.40121760339038276, 9.164404170338044E-4, 4.582202085169022E-4]
[1998.0, 870.0999755859375, 2.0, 1.0]	[0.9168337785406577, 0.39926779195424883, 9.177515300707285E-4, 4.5887576503536424E-4]
[1997.0, 877.7000122070312, 2.0, 1.0]	[0.9154800836855296, 0.40236198328797357, 9.16855366735633E-4, 4.584276833678165E-4]
[1996.0, 893.7000122070312, 2.0, 1.0]	[0.9126894417403085, 0.40865258778784685, 9.14518478697704E-4, 4.57259239348852E-4]
[1995.0, 909.5, 2.0, 1.0]	[0.9099046611851377, 0.41481618513678337, 9.121851239951255E-4, 4.5609256199756275E-4]

Рисунок 7. Исходные признаки и нормализованные

```
RMSE: 2.1717343626508003
MSE: 4.716430141918279
r2: 0.9672659146161299
```

Рисунок 7. Метрики качества при нормализованных данных

После нормализации данных, качество несколько снизилось. Скорее всего, это произошло из-за наличия категориальных признаков Пола и Расы. В нормализованных данных, категориальные признаки имеют очень маленькие веса, хотя из предыдущей работы было видно, что признак Расы имеет достаточно большое влияние на среднюю продолжительность жизни.