Методы восстановления пропусков в данных

Каюмов Эмиль

ММП ВМК МГУ

Спецсеминар
«Алгебра над алгоритмами и эвристический поиск закономерностей»

16 мая 2016

План

Зачем это нужно?

Большинство реальных данных имеют пропущенные значения.

- Ошибки при записи.
- Ошибки при измерении.
- Невозможность сбора.

Далеко не все алгоритмы умеют работать с неполными данными.

Содержание

Простые методы

- Удаление объектов с пропущенными значениями.
  - Можно удалять не объекты, а признаки.
  - Ничего не испортим, но что если данных и так мало?
- Замена специальным значением.
  - Для категориального признака можно интерпретировать как индикатор пропущенного значения.
  - Как понимать специальное значение в случае вещественного признака?
- Замена средним значением признака.
- Замена модой признака.

Замена с помощью сингулярного разложения

Сингулярное разложение: $X = U\Sigma V^*$

Используется по аналогии с приближением матрицы матрицей меньшего ранга, занулением диагональных элементов Σ за исключением k наибольших.

Algorithm 1 SVD Imputer

$X[missing] \leftarrow simple\ initialize(X)$
for $iteration = 1$ to $max\_iterations$ **do**
  $U, \Sigma, V \leftarrow SVD(X)$
  $\Sigma^{'} \leftarrow reduce(\Sigma, k)$
  $X_{approx} \leftarrow U\Sigma^{'}V^*$
  $X[missing] \leftarrow X_{approx}$
end for

Но надо выбрать ранг аппроксимирующей матрицы.

Замена с помощью метода k ближайших соседей

Algorithm 2 kNN Imputer

1: $X_{full} = X[rows\ without\ missing\ values]$
2: **for** $row\ with\ missing\ values$ **do**
3: $\quad X_{neighbors} \leftarrow find\ k\ neighbors(row, X_{full}, k)$
4: $\quad row[missing] \leftarrow mean(X_{neighbors})$
5: **end for**

Необходимо выбрать метрику и число соседей.

Замена с помощью случайного леса

Algorithm 3 RF Imputer

1: $X[missing] \leftarrow simple\ initialize(X)$
2: **for** $iteration = 1$ to $max\_iterations$ **do**
3: **for** $column\ with\ missing\ values$ **do**
4: $X_{train} \leftarrow X[without\ missing\ values]$
5: $X_{test} \leftarrow X[with\ missing\ values]$
6: $X[missing, column] \leftarrow predict\ RF(X_{train}, X_{test})$
7: **end for**
8: **end for**

Нет важных для настраивания параметров.

Замена с помощью линейной регрессии

Algorithm 4 LR Imputer

$X[missing] \leftarrow simple\ initialize(X)$
for $iteration = 1$ to $max\_iterations$ **do**
  for $column\ with\ missing\ values$ **do**
    $X_{train} \leftarrow X[without\ missing\ values]$
    $X_{test} \leftarrow X[with\ missing\ values]$
    $X[missing, column] \leftarrow predict\ LR(X_{train}, X_{test})$
  end for
end for

Можно заменить линейную регрессию на любой другой алгоритм предсказания.

Замена с помощью EM-алгоритма

Смесь нормальных распределений: $p(x) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\mu_i, \Sigma_i)$.
Коэффициенты регрессии: $\beta = cov(X, y)\Sigma^{-1}$.
По коэффициентам пересчитываем пропущенные значения, усредняем по смеси.

Algorithm 5 EM Imputer

$X[missing] \leftarrow simple\ initialize(X)$
for $iteration = 1$ to $max\_iterations$ **do**
  $\pi, \mu, \Sigma \leftarrow X$
  for $row\ with\ missing\ values$ **do**
    for $k = 1$ to K **do**
      $coef \leftarrow calculate(\mu, \Sigma)$
      $predict_i \leftarrow regression(coef, X[row, nonmissing]$
    end for
    $X[row, missing] \leftarrow \sum_{i=1}^{K} \pi_i predict_i$
  end for
end for

Замена с помощью метода k средних

Algorithm 6 K-means Imputer

1: $X[missing] \leftarrow simple\ initialize(X)$
2: **for** $iteration = 1$ to $max\_iterations$ **do**
3: $\quad centroids \leftarrow kmeans(X)$
4: $\quad X[missing] \leftarrow centroids$
5: **end for**

Как определить число кластеров?

Алгоритм ZET (1)

Основывается на линейной регрессии по выбранным компетентным столбцам и строкам.

$L_{iy} = \frac{\#nonmissing\ in\ i,y}{distance(i,y)}$ – компетентность строки i к строке y.
$L_{iy} = |cor(i,y)|distance(i,y)$ – компетентность столбца i к столбцу y.

Выбирается заданное число строк и столбцов с наибольшей компетентностью. Настраивается степень учета компетентности строки или столбца α как доставляющая минимальное отклонение предказаний известных значений строки и столбца с пропущенным значением: $\sum_i |a_{ik} - b_{ik}| \to min$.

Алгоритм ZET (2)

$$b_{ik} = \frac{\sum_{j=1}^{c-1} bl_{jk} L_{ij}^{\alpha}}{\sum_{j=1}^{c-1} L_{ij}^{\alpha}},$$

где bl_{ik} – прогноз для значений строки (столбца) k с помощью i строки (столбца) линейной регрессии вида $y = ax + b$.

После нахождения оптимальных α для строк и столбцов вычисляется по аналогичной формуле прогноз пропущенного значения по строкам и столбцам, прогнозы устредняются.

Необходимо задать количество компетентных строк и столбцов, пределы изменения α. Работает на порядок дольше любого другого метода.

Особенности реализации

1. Большинство методов может выдавать дробное число для признака, являющегося категориальным значением. Это можно обойти округлением до ближайшего известного значения в выборке.
2. Дополнительным вариантом является добавление нового бинарного признака-индикатора пропущенного значения.

Содержание

Датасеты

- Без пропущенных значений:
  1. KRKP (KingRook vs KingPawn chess game): 3196 объектов и 36 признаков, все из которых категориальные.
  2. Creditg (German Credit Data): 1000 объектов и 20 признаков, среди которых есть и категориальные, и количественные.
  3. Segment (Image Segmentation): 2310 объектов и 19 признаков, все из которых количественные.
- С пропущенными значениями:
  1. Horse (Horse Colic): 300 объектов и 22 признаки, среди которых есть и категориальные, и количественные. 30% пропущенных значений.
  2. Votes (Congressional Voting Records): 435 объектов и 16 признаков, среди которых все категориальные. 6% пропущенных значений.
  3. Cancer (Breast Cancer Wisconsin): 699 объектов и 9 признаков, все из которых количественные. Менее 1% пропущенных значений.

Алгоритмы

Алгоритмы разной природы:

- Случайный лес
- Логистическая регрессия
- Метод k ближайших соседей

Создание пропущенных значений

Создание пропущенных значений в случайном подмножестве признаков будет приводить к нестабильности в результатах (попадет ли важный признак в число удаляемых или нет). Поэтому пропуски создаются в фиксированном подмножестве важных по оценке случайного леса признаках.

Выбирается 25% наиболее важных признаков. Далее с заданной вероятностью создаются пропущенные значения в заданных признаках.

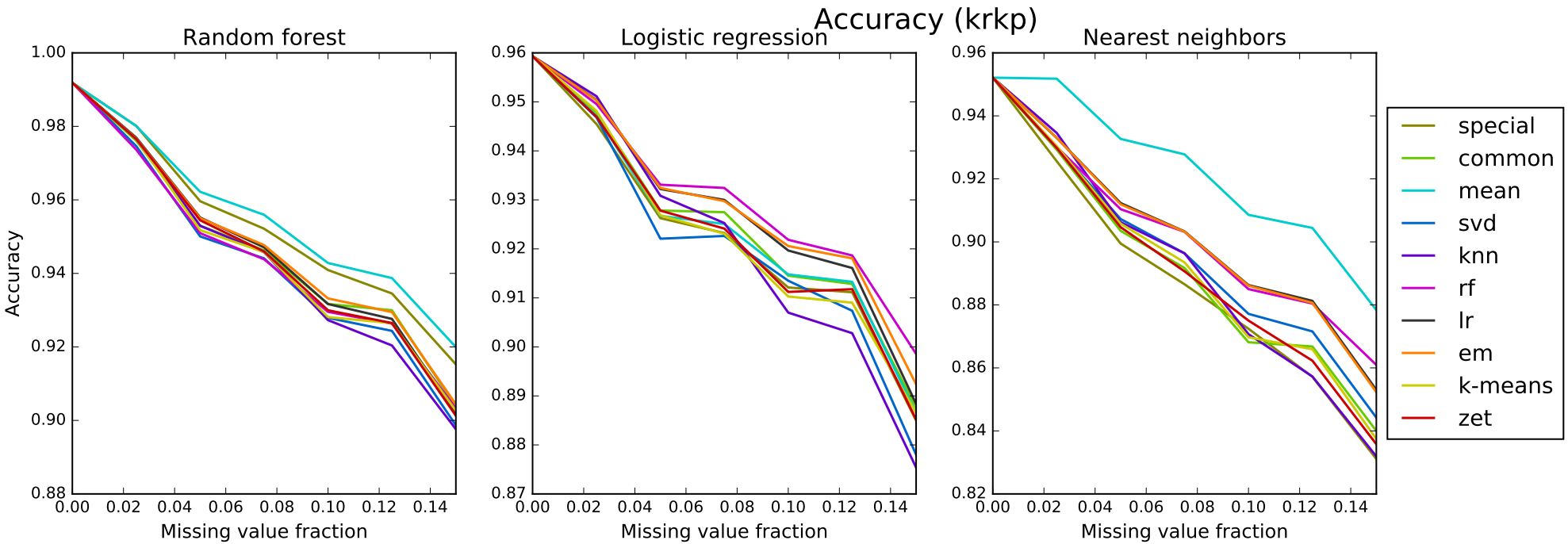
Параметры методов

1. Замена специальным значением: для применения случайного леса пропуск заменялся -1, для логистической регрессии и метода ближайшего соседа — 0.
2. Сингулярной разложение: ранг аппроксимирующей матрице в два раза меньше количества признаков, максимальное число итераций равно 10.
3. Метод k ближайших соседей: k = 5, метрика пространства L2.
4. Случайный лес: 10 деревьев, максимальное число итераций — 3.
5. Линейная регрессия: максимальное число итераций — 3.
6. EM-алгоритм: 1 смесь нормального распределения с полной матрицей ковариации.
7. Метод k средних: 8 кластеров, максимальное число итераций — 3.
8. ZET: число компетентных строк — 6, число компетентных столбцов — 4.

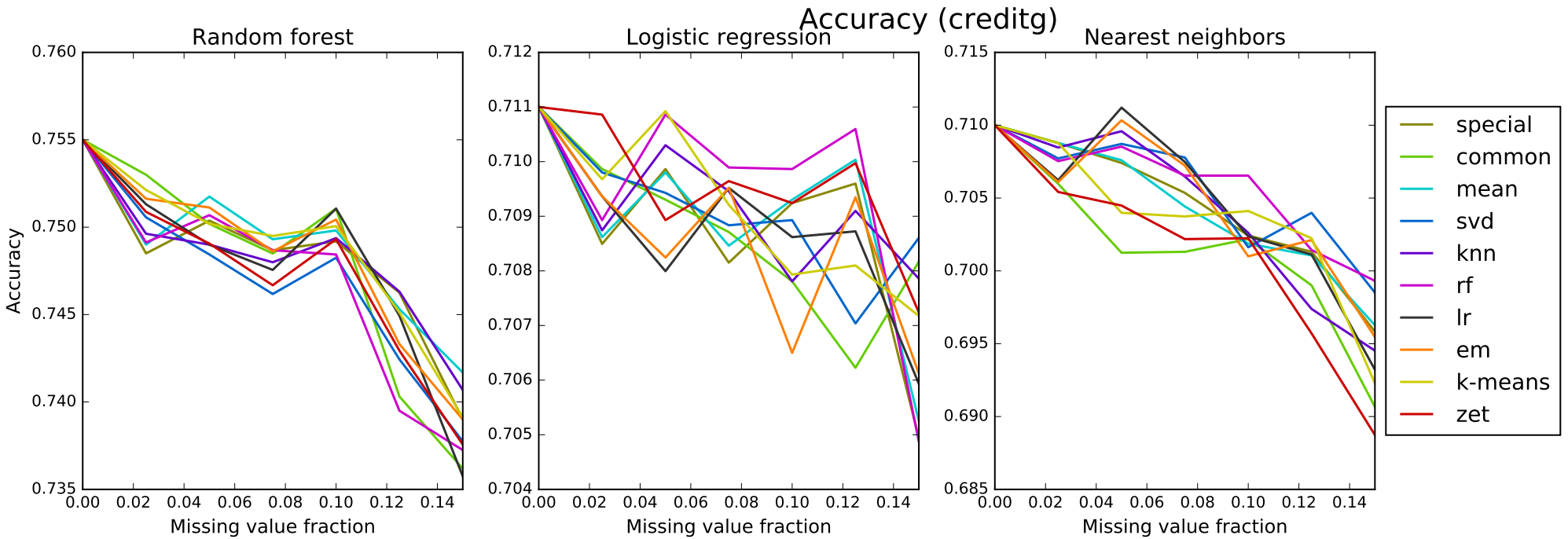
Условия

- 10-folds stratified cv.
- Усреднее результатов по 10 запускам.
- Измеряется точность классификации и среднеквадратичное отклонение (для датасетов без пропущенных значений).
- Для всех методов за исключением замены средним и модой производится округление до ближайшего известного значения.

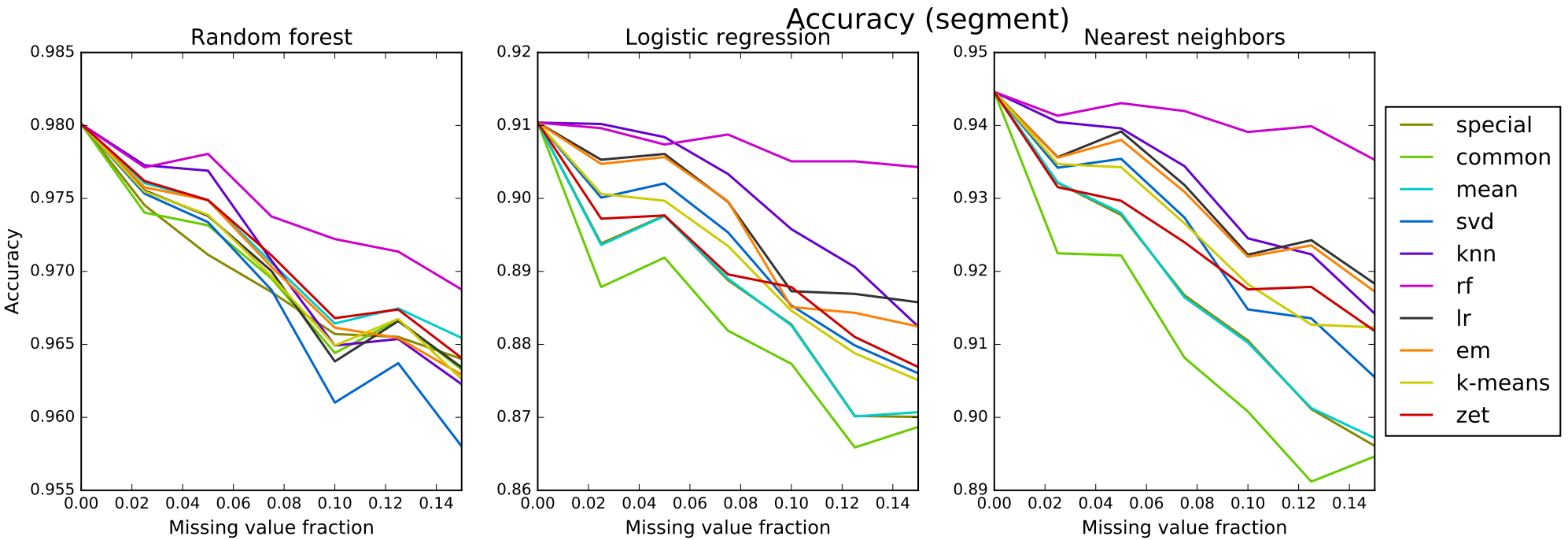
KRKP


Accuracy (krkp)
Random forest
Logistic regression
Nearest neighbors
Accuracy
Missing value fraction
Missing value fraction
Missing value fraction
special
common
mean
svd
knn
rf
lr
em
k-means
zet

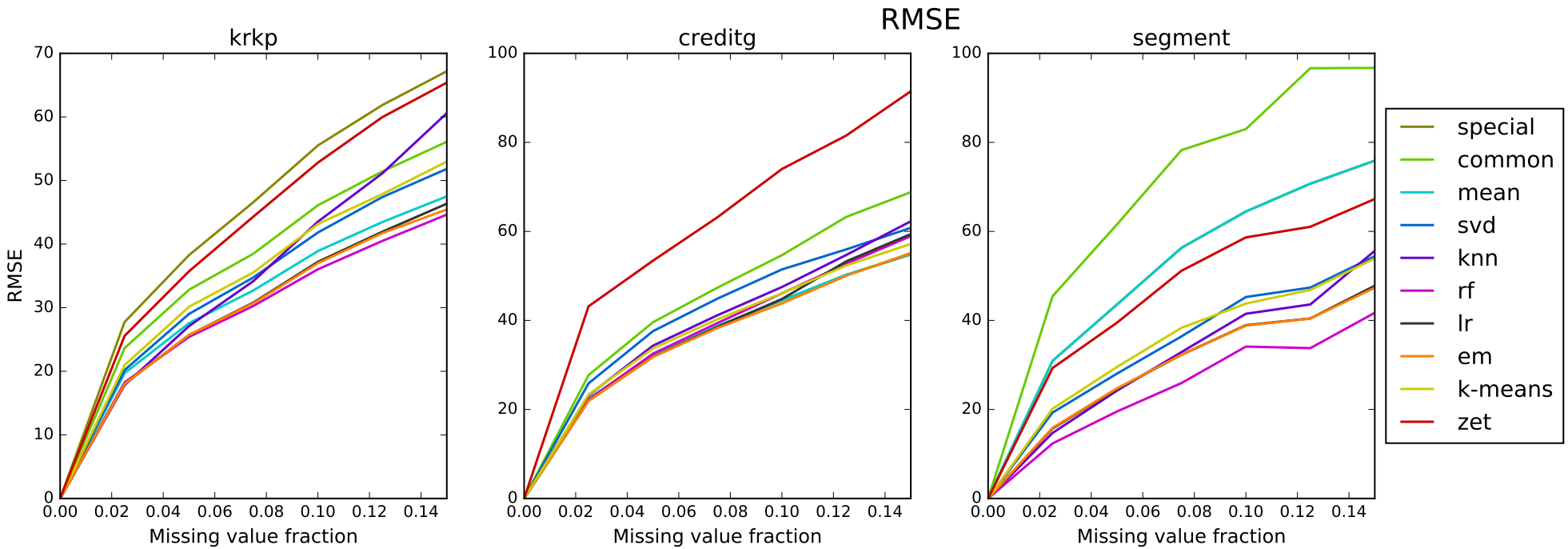
Creditg


Accuracy (creditg)
Random forest
Logistic regression
Nearest neighbors
Accuracy
Missing value fraction
Missing value fraction
Missing value fraction
special
common
mean
svd
knn
rf
lr
em
k-means
zet

Segment


Accuracy (segment)
Random forest
Logistic regression
Nearest neighbors
Accuracy
Missing value fraction
Missing value fraction
Missing value fraction
special
common
mean
svd
knn
rf
lr
em
k-means
zet

RMSE

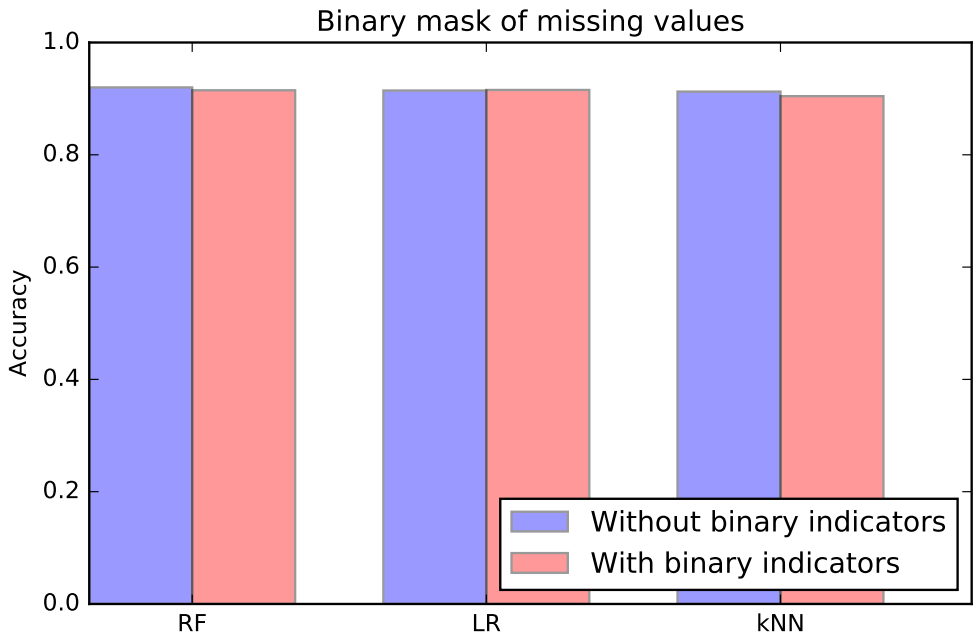

RMSE
krkp
creditg
segment
RMSE
Missing value fraction
Missing value fraction
Missing value fraction
special
common
mean
svd
knn
rf
lr
em
k-means
zet

Натуральные пропуски

Datasets	Horse			Votes			Cancer		
Methods	RF	LR	kNN	RF	LR	kNN	RF	LR	kNN
Ignore	-	-	-	0.9517	0.9527	0.9225	0.9591	**0.9694**	0.9694
Special	**0.8599**	0.8004	0.8400	0.9586	0.9584	0.9217	0.9557	0.9686	**0.9719**
Common	**0.8532**	0.8135	0.8270	**0.9633**	**0.9608**	0.9264	0.9542	0.9686	0.9685
Mean	0.8433	0.8004	0.8400	**0.9632**	0.9562	0.9401	0.9585	0.9686	**0.9714**
SVD	0.8201	0.8097	**0.8601**	0.9495	0.9540	0.9309	**0.9628**	0.9686	0.9700
kNN	0.8434	0.8166	0.8101	0.9517	0.9587	0.9240	**0.9628**	0.9686	0.9700
RF	0.8203	0.8065	0.8133	0.9490	0.9539	0.9240	0.9600	0.9686	0.9700
LR	0.8339	**0.8196**	0.8266	0.9565	0.9517	0.9332	**0.9628**	0.9686	0.9700
EM	0.8366	**0.8197**	0.8266	0.9518	0.9563	0.9357	**0.9628**	0.9686	0.9700
k-means	0.8464	0.8167	0.8432	0.9424	**0.9608**	**0.9423**	**0.9628**	0.9686	0.9700
ZET	0.8466	0.8097	0.8134	0.9516	**0.9630**	0.9218	0.9571	0.9686	0.9700

Добавление бинарного признака

Добавление дополнительного бинарного признака почти ничего не меняет.



Содержание

Выводы

- Ни один из методов не превосходит все остальные. Иногда не имеет разницы, какой метод использовать.
- Замена модой, средним или специальным значением показывает неплохие результаты.
- На данных с натуральными пропусками среди продвинутых методов чаще других показывал лучший результат метод, основанный на методе k средних.

Ссылки

Реализация всех описанных методов:
`https://github.com/emilkayumov/missing-value`