Лабораторная работа №3

Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных

Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.) Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи: обработку пропусков в данных; кодирование категориальных признаков; масштабирование данных.

В чем состоит проблема: Если в данных есть пропуски, то большинство алгоритмов машинного обучения не будут с ними работать. Даже корреляционная матрица не будет строиться корректно. Большинство алгоритмов машинного обучения требуют явного перекодирования категориальных признаков в числовые. Даже если алгоритм не требует этого явно, такое перекодирование возможно стоит попробовать, чтобы повысить качество модели. Большинство алгоритмов показывает лучшее качество на отмасштабированных признаках, в особенности алгоритмы, использующие методы градиентного спуска.

Загружаем окружение

12 cells hidden

Работа с данными

```
1 companies = pd.read_csv('acquisitions.csv', sep=',')
2 companies.head(10)
```

	AcquisitionID	AcquisitionMonth	AcquisitionMonthDate	AcquisitionYear
0	ACQ99	November	11.0	2015

```
1 companies.shape
```

```
(916, 10)
```

```
1 companies.dtypes
```

```
AcquisitionID            object
AcquisitionMonth         object
AcquisitionMonthDate    float64
AcquisitionYear           int64
Company                  object
Business                 object
Country                  object
Value (USD)             float64
Derived products         object
ParentCompany            object
dtype: object
```

Проверка на пустые значения:

```
1 companies.isnull().sum()
2 # for column in companies.columns:
3 #     buf_null = companies[companies[column].isnull()].shape[0]
4 #     print ('{}-{}'.format(column, buf_null))
5
6 # acquisition - приобретение, овладение
7 # derived products - производные продукты
```

```
AcquisitionID             0
AcquisitionMonth          6
AcquisitionMonthDate     33
AcquisitionYear           0
Company                   0
Business                  0
Country                  46
Value (USD)             671
Derived products        515
ParentCompany             0
dtype: int64
```

Вывод: по полям AcquisitionMont, AcquisitionMonthDate, Country-46 - пропуски данных небольшие, это не сильно повлияет на анализ

- По полям Value (USD) и Derived products пропуски более 50% от dataset,сильное влияние

```
1 total_count = companies.shape[0]
2 print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 916
```

1. Обработка пропусков в данных

1.1. Простые стратегии - удаление или заполнение нулями Удаление колонок, содержащих пустые значения

```
data_new_1 = companies.dropna(axis=1, how='any')
(companies.shape, data_new_1.shape)
```

```
((916, 10), (916, 5))
```

```
data_new_1.head(5)
```

	AcquisitionID	AcquisitionYear	Company	Business	ParentCompany
0	ACQ99	2015	bebop	Cloud software	Google
1	ACQ98	2015	Fly Labs	Video editing	Google
2	ACQ97	2015	Clearleap	Cloud-based video management	IBM
3	ACQ96	2015	Metanautix	Big Data Analytics	Microsoft
4	ACQ95	2015	Talko Inc.	Mobile	Microsoft

```
data_new_1.shape
```

```
(916, 5)
```

Удаление строк, содержащих пустые значения

```
data_new_2 = companies.dropna(axis=0, how='any')
(companies.shape, data_new_2.shape)
```

```
((916, 10), (114, 10))
```

```
data_new_2.head(5)
```

	AcquisitionID	AcquisitionMonth	AcquisitionMonthDate	AcquisitionYear	C
0	ACQ99	November	11.0	2015	
38	ACQ889	February	7.0	1997	
47	ACQ880	October	8.0	1997	

```
data_new_2.shape
```

```
(114, 10)
```

Заполнение всех пропущенных значений нулями

- В данном случае это некорректно, так как нулями заполняются в том числе категориальные колонки

```
data_new_3 = companies.fillna(0)
data_new_3.isnull().sum()
```

```
AcquisitionID           0
AcquisitionMonth        0
AcquisitionMonthDate    0
AcquisitionYear         0
Company                 0
Business                0
Country                 0
Value (USD)             0
Derived products        0
ParentCompany           0
dtype: int64
```

1.2. "Внедрение значений" - импьютация (imputation)

1.2.1. Обработка пропусков в числовых данных

- Импьютация - процесс замены пропущенных, некорректных или несостоятельных значений другими значениями
- Выберем числовые колонки с пропущенными значениями
- Цикл по колонкам датасета
- Выберем числовые колонки с пропущенными значениями
- Цикл по колонкам датасета

```
num_cols = []
for col in companies.columns:
    # Количество пустых значений
    temp_null_count = companies[companies[col].isnull()].shape[0]
    dt = str(companies[col].dtype)
    total_count = companies.shape[0]
    if temp_null_count>0 and (dt=='float64' or dt=='int64'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format
```

```
Колонка AcquisitionMonthDate. Тип данных float64. Количество пустых значений 33, 3.
Колонка Value (USD). Тип данных float64. Количество пустых значений 671, 73.25%.
```

Фильтр по колонкам с пропущенными значениями

```
data_num = companies[num_cols]
data_num
```

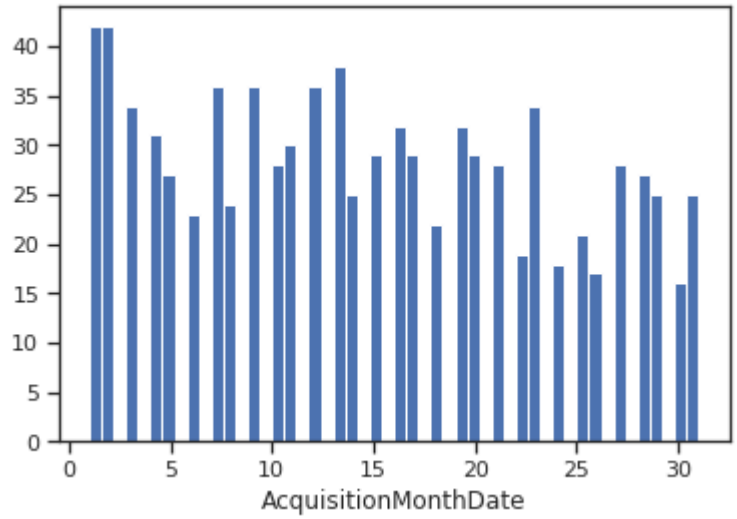
	AcquisitionMonthDate	Value (USD)
0	11.0	3.800000e+08
1	11.0	NaN
2	8.0	NaN
3	18.0	NaN
4	21.0	NaN
5	7.0	NaN
6	15.0	NaN
7	19.0	NaN
8	30.0	1.400000e+07
9	2.0	NaN
10	7.0	NaN
11	27.0	NaN
12	11.0	NaN
13	3.0	NaN
14	21.0	NaN
15	31.0	NaN
16	29.0	NaN
17	28.0	NaN
18	27.0	NaN
19	1.0	NaN
20	15.0	NaN
21	23.0	NaN
22	10.0	NaN
23	17.0	NaN
24	6.0	NaN
25	28.0	NaN
26	16.0	NaN
27	12.0	NaN
28	16.0	1.330000e+08
29	6.0	NaN
...	...	...
886	23.0	NaN
887	31.0	1.600000e+08

888	3.0	NaN
889	6.0	1.000000e+09
890	NaN	NaN
891	5.0	NaN
892	NaN	NaN
893	NaN	NaN
894	3.0	NaN
895	10.0	NaN
896	11.0	NaN
897	21.0	NaN
898	28.0	NaN
899	28.0	NaN
900	30.0	NaN
901	2.0	NaN
902	9.0	NaN
903	3.0	NaN
904	17.0	NaN
905	21.0	NaN
906	21.0	NaN
907	28.0	NaN
908	NaN	NaN
909	3.0	NaN
910	5.0	NaN
911	6.0	1.309000e+09
912	9.0	NaN
913	11.0	NaN

Гистограмма по признакам

```
for col in data_num:
    plt.hist(companies[col], 50)
    plt.xlabel(col)
    plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/numpy/lib/histograms.py:824: RuntimeWa
  keep = (tmp_a >= first_edge)
/usr/local/lib/python3.6/dist-packages/numpy/lib/histograms.py:825: RuntimeWa
  keep &= (tmp_a <= last_edge)
```



Фильтр по пустым значениям поля AcquisitionMonthDate

```
1 companies[companies['AcquisitionMonthDate'].isnull()]
```

162	ACQ777	October	NaN	2003
166	ACQ773	January	NaN	2004
182	ACQ759	September	NaN	2004
184	ACQ757	October	NaN	2004
198	ACQ744	March	NaN	2005
205	ACQ738	April	NaN	2005
218	ACQ726	July	NaN	2005
233	ACQ712	November	NaN	2005
301	ACQ651	December	NaN	2006
474	ACQ496	August	NaN	2010
571	ACQ408	NaN	NaN	2012
629	ACQ356	NaN	NaN	2013
630	ACQ355	NaN	NaN	2013
641	ACQ345	March	NaN	2013
713	ACQ280	December	NaN	2013
733	ACQ262	NaN	NaN	2014
840	ACQ166	January	NaN	2015
858	ACQ15	October	NaN	2017
862	ACQ146	April	NaN	2015

Запоминаем индексы строк с пустыми значениями

```
flt_index = companies[companies['AcquisitionMonthDate'].isnull()].index
flt_index
```

```
Int64Index([ 45,  61,  99, 100, 144, 149, 150, 161, 162, 166, 182, 184, 198,
             205, 218, 233, 301, 474, 571, 629, 630, 641, 713, 733, 840, 858,
             862, 869, 872, 890, 892, 893, 908],
            dtype='int64')
```

Проверяем что выводятся нужные строки

```
companies[companies.index.isin(flt_index)]
```

162	ACQ777	October	NaN	2003
166	ACQ773	January	NaN	2004
182	ACQ759	September	NaN	2004
184	ACQ757	October	NaN	2004
198	ACQ744	March	NaN	2005
205	ACQ738	April	NaN	2005
218	ACQ726	July	NaN	2005
233	ACQ712	November	NaN	2005
301	ACQ651	December	NaN	2006
474	ACQ496	August	NaN	2010
571	ACQ408	NaN	NaN	2012

фильтр по колонке

```
data_num[data_num.index.isin(flt_index)]['AcquisitionMonthDate']
```

```
45     NaN
61     NaN
99     NaN
100    NaN
144    NaN
149    NaN
150    NaN
161    NaN
162    NaN
166    NaN
182    NaN
184    NaN
198    NaN
```

Будем использовать встроенные средства импьютации библиотеки scikit-learn - https://scikit-learn.org/stable/modules/impute.html#impute

```
1 data_num_AcquisitionMonthDate = data_num[['AcquisitionMonthDate']]
2 data_num_AcquisitionMonthDate.head()
```

	AcquisitionMonthDate
0	11.0
1	11.0
2	8.0
3	18.0
4	21.0

```
908    NaN
```

```
1 from sklearn.impute import SimpleImputer
2 from sklearn.impute import MissingIndicator
```

Фильтр для проверки заполнения пустых значений

```
1 indicator = MissingIndicator()
2 mask_missing_values_only = indicator.fit_transform(data_num_AcquisitionMonthDat
3 mask_missing_values_only
```

```
array([[False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```

```
[ True],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[ True],
[ True],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
```

```
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[ True],
[False],
[False],
[False],
[False],
[ True],
[ True],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[ True],
[ True],
[False],
[False],
[False],
[ True],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[ True],
[False],
[ True]
```

[True],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[True],
[False],
[False],
[False],
[False],
[False],
[False],
[True],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[True],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[True],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],

```
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[ True],
[False],
[False],
[False],
[False],
[False],
[False],
```

```
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
```

```
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```

```
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
[False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```