Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ**

Отчёт по курсовой работе «Исследование факторов влияющих на успешность стартапов при выходе на международный рынок»

по курсу “Методы машинного обучения”

Исполнитель:
Студент группы ИУ5-22М
Желанкина А.С.
20.05.2021

Москва, 2021

Постановка задачи машинного обучения

Рассмотрим статистику стартапов за 10 лет, которые были созданы в с 2011 года. Выбор этого периода объясняется тем, что за это время часть исследуемых стартапов с большой вероятностью достигла поставленных целей. Для построения модели была использована база стартапов Crunchbase. Из неё был сформирован датасет, состоящий из 11324 строк и 24 столбцов.

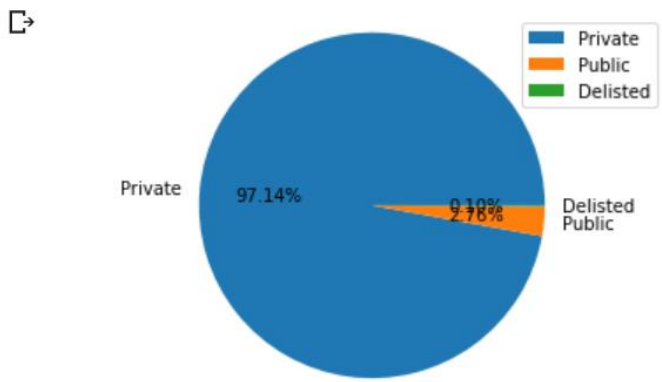
Необходимо провести исследование для определения факторов, влияющих на успешность стартапов на рынке, и разработать модель, которая могла бы предсказать возможные успешные стартапы по имеющемуся набору характеристик.

Решение задачи

Для начала было решено просмотреть состав двух переменных, из которых собирается целевая. Первой была рассмотрена переменная 'IPO Status'.

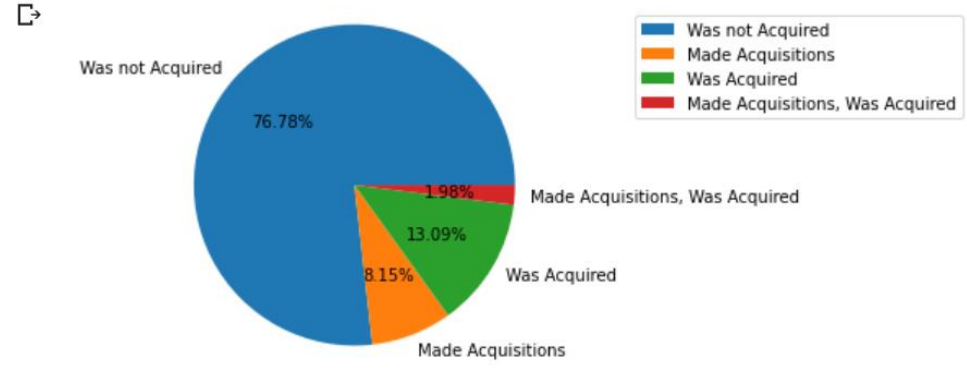
IPO может принимать такие значения, как «Private», «Delisted» и «Public». В случае успешного IPO повышается рыночная стоимость компании. Рассмотрим подробнее принимаемые переменной значения. «Public» статус означает, что стартапу удалось успешно разместить акции на бирже, «Private» – компания ещё не успела провести IPO и до этого момента считается частной. «Delisted» – компания перестала выставляться как публичная, возможно ей не удалось выйти на IPO или же она предпочла вид частного капитала. Обзор распределения в представленных данных показал, что большую часть рынка (97,14%) занимают компании с «Private» статусом, то есть еще не разместившие свои акции на бирже (рис. 1). Публичных компаний значительно меньше – 2,76%, в то время как стартапов со статусом «Delisted» всего 0,1%. Успешными стартапами в данном случае будут считаться компании, вышедшие на IPO («Public»).

```
fig1, ax1 = plt.subplots()

wedges, texts, autotexts = ax1.pie([10933, 311, 11], labels=labels, autopct='%1.2f%%')
ax1.axis('equal')
ax1.legend(loc='upper right', bbox_to_anchor=(1.0, 1.0))
plt.show()
```

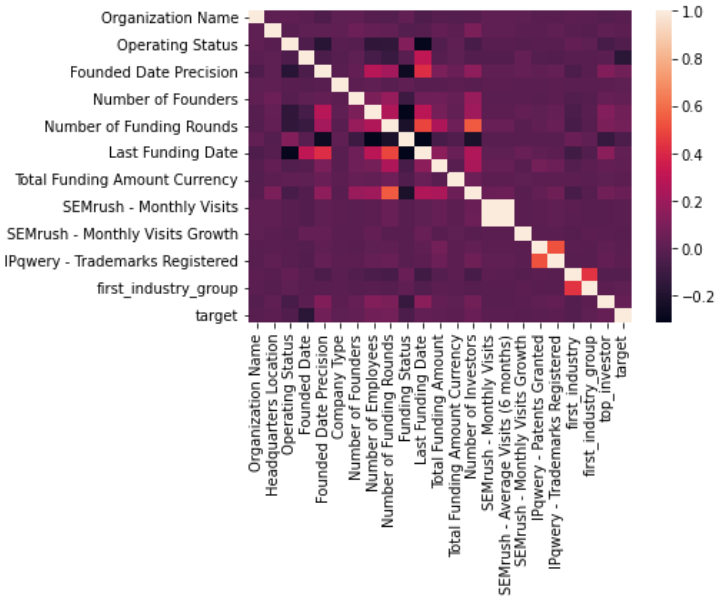


Вторая переменная, Acquisition Status, обозначает статус приобретения стартапа и так же имеет четыре значения: не была продана (“Was not Acquired”), была продана (“Was Acquired”), приобрела другую компанию (“Made Acquisitions”), приобрела другую компанию и была куплена (“Made Acquisitions, Was Acquired”). Большую часть рынка (76,78%) занимают стартапы, которые еще не были приобретены (рис. 2). Приобретенные компании составляют 13,09% от общего числа. Стартапы, совершившие покупку других компаний составляют 8,15%, а стартапы с обеими операциями насчитывают всего лишь 1,98%. Статус «Was Acquired» используется в случае, если компания была продана, что является одним из параметров оценки успешности стартапа. Также можно считать успешной компанию, чей статус равен “Made Acquisitions, Was Acquired”, так как это означает, что компания была продана и при этом успела приобрести стартап. Статус «Made Acquisitions» как правило связан с покупкой другого стартапа.

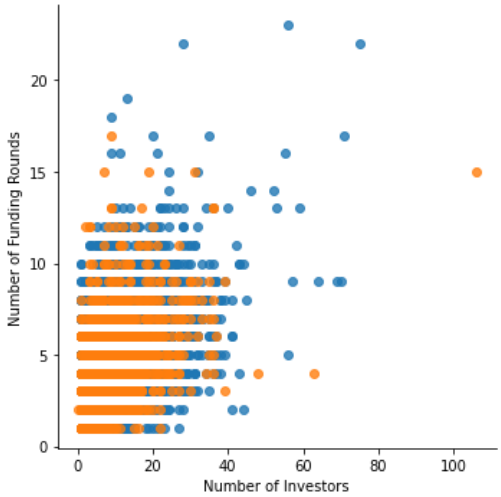
```
fig1, ax1 = plt.subplots()

wedges, texts, autotexts = ax1.pie([8642, 917, 1473, 223], labels=labels, autopct='%1.2f%%')
ax1.axis('equal')
ax1.legend(loc='upper left', bbox_to_anchor=(1.0, 1.0))
plt.show()
```



Переменные IPO Status и Acquisition Status будут рассмотрены в паре. Поэтому из них будет создана целевая переменная ‘target’. Посмотрим, есть ли явная корреляция целевой переменной с какой-либо другой из набора. Можно заметить, что целевая переменная ни с одной другой не имеет сильной связи. Однако сильно взаимосвязаны оказались число раундов инвестиций и количество инвесторов, а также зарегистрированные торговые марки и патенты, которыми владеет компания.



Рассмотрим найденные корреляции поподробнее. Большинство успешных стартапов имеет число раундов инвестиций не более 10 и количество инвесторов до 20.



Целевая переменная имеет только два значения: 0 – неуспешный стартап, 1 – успешный стартап. С помощью следующего графика можно проиллюстрировать, что успешных стартапов в несколько раз меньше.

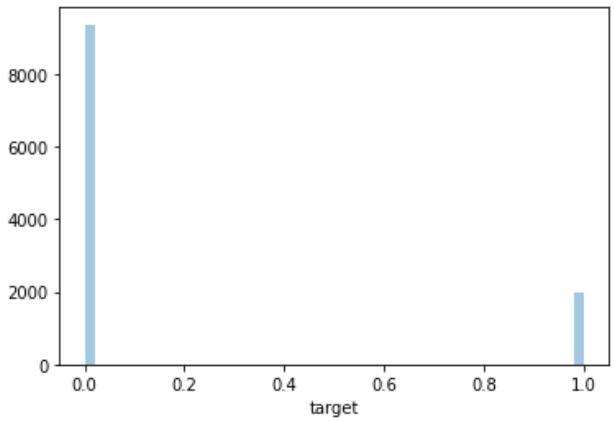
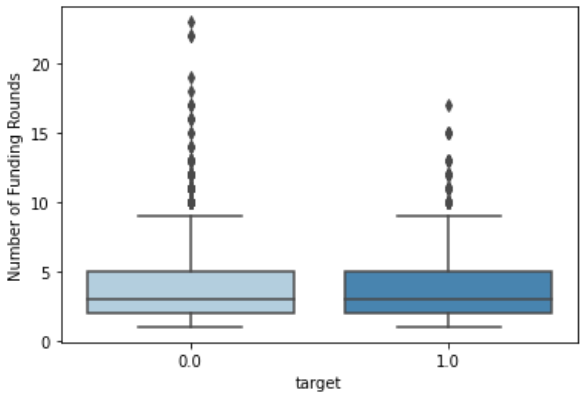
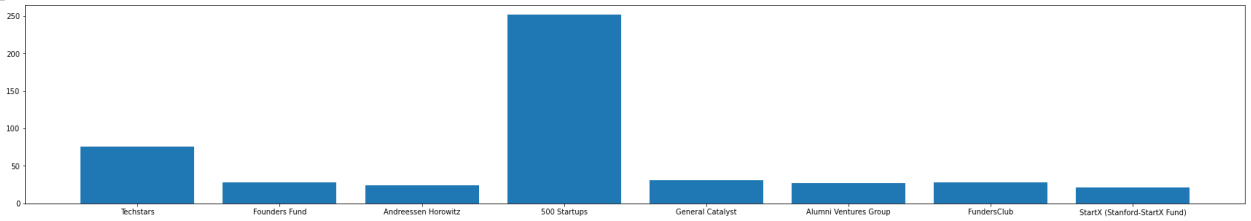


Диаграмма ящик с усами в удобной форме показывает медиану, нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы. Рассмотрим ящик для целевой переменной по числу раундов инвестиций. Медиана успешных стартапов по числу инвесторов находится в районе 4.



Отношение успешных компаний по целевой переменной к инвесторам. На график выведем восьмёрку лидеров. Лидирующую позицию с большим отрывом от остальных занимают стартапы, в которые инвестировали «500 Startups».



Такая информация выводится о датасете.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11355 entries, 0 to 11365
Data columns (total 24 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Organization Name                     11355 non-null  int64
 1   Headquarters Location                 11355 non-null  int64
 2   Operating Status                      11355 non-null  int64
 3   Founded Date                          11355 non-null  int64
 4   Founded Date Precision                11355 non-null  int64
 5   Company Type                          11355 non-null  int64
```

```
6   Number of Founders                    11355 non-null  float64
7   Number of Employees                   11355 non-null  int64
8   Number of Funding Rounds              11355 non-null  float64
9   Funding Status                        11355 non-null  int64
10  Last Funding Date                     11355 non-null  int64
11  Total Funding Amount                  11355 non-null  float64
12  Total Funding Amount Currency         11355 non-null  int64
13  Number of Investors                   11355 non-null  float64
14  Acquisition Status                    11355 non-null  object
15  IPO Status                            11355 non-null  object
16  SEMrush - Monthly Visits              11355 non-null  float64
17  SEMrush - Average Visits (6 months)   11355 non-null  float64
18  SEMrush - Monthly Visits Growth       11355 non-null  float64
19  IPqwery - Patents Granted             11355 non-null  float64
20  IPqwery - Trademarks Registered       11355 non-null  float64
21  first_industry                        11355 non-null  int64
22  first_industry_group                  11355 non-null  int64
23  top_investor                          11355 non-null  int64
    dtypes: float64(9), int64(13), object(2)
```

Так как присутствует большое количество колонок, в которых достаточно сложно заполнить пропуски (больше 50%), или коррелирующих между собой колонок, то такие данные было решено удалить.

```
[183] data = data.drop('Valuation at IPO Currency (in USD)', 1)
      data = data.drop('Price Currency (in USD)', 1)
      data = data.drop('Total Funding Amount Currency (in USD)', 1)
```

```
data = data.drop('Diversity Spotlight (US Only)', 1)
data = data.drop('Valuation at IPO', 1)
data = data.drop('Valuation at IPO Currency', 1)
data = data.drop('Acquisition Type', 1)
data = data.drop('Number of Acquisitions', 1)
data = data.drop('Price', 1)
data = data.drop('Price Currency', 1)
data = data.drop('Announced Date', 1)
data = data.drop('Announced Date Precision', 1)
data = data.drop('Organization Name URL', 1)
```

Пропуски в числовых значениях заменяли на значение медианы в данном столбце.

```
def repl(col):
    new_col = []
    for n in col:
        if type(n) == str:
            #print('{} = {}, {}'.format(n, n.replace(',', ''), i))
            n = n.replace(',', '')
            #print(n)
            n = float(n)
        new_col.append(n)
    new_col = pd.Series(new_col)
    return new_col

array = ['Total Funding Amount',
         'Number of Funding Rounds',
         'Number of Investors',
         'Number of Founders',
         'SEMrush - Monthly Visits',
         'SEMrush - Average Visits (6 months)',
         'IPqwery - Patents Granted',
         'SEMrush - Monthly Visits Growth',
         'IPqwery - Trademarks Registered']
for i in array:
    data[i] = repl(data[i])
    data[i] = data[i].fillna(data[i].median())
```

Заполнение категориальных пропусков зависит от столбца, в котором есть пропуски. Заполнялось либо наиболее вероятным значением, либо наиболее часто встречающимся, либо ничего не значащим.

In [11]:
```
data['Acquisition Status'] = data['Acquisition Status'].fillna('Was not Acquired')
```

In [12]:
```
simp = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
data[['Industry Groups']] = simp.fit_transform(data[['Industry Groups']])
data[['Number of Employees']] = simp.fit_transform(data[['Number of Employees']])
data[['Industries']] = simp.fit_transform(data[['Industries']])
```

Кодирование категориальных значений с помощью LabelEncoder.

```
label_enc = LabelEncoder()
obj_columns = ['Organization Name', 'Headquarters Location', 'Operating Status',
               'Founded Date', 'Founded Date Precision', 'Company Type',
               'Number of Employees', 'Funding Status', 'Last Funding Date',
               'Total Funding Amount Currency', 'first_industry', 'first_industry_group',
               'top_investor']

for obj in obj_columns:
  print(obj)
  data[obj] = label_enc.fit_transform(data[[obj]])
```

Нормализация обучающей выборки из датасета производилась с помощью MinMaxScaler.

In [37]:
```
min_max_sc = MinMaxScaler()

x_train = min_max_sc.fit_transform(x_train)
X_test = min_max_sc.transform(X_test)
```

Обработка нестандартного признака:

```
1 data['Industries']
```

```
0                      Biotechnology, Health Care, Medical
1                      Biotechnology, Health Care, Medical
2                      Biotechnology, Health Care, Medical
3                      Biotechnology, Health Care, Medical
4                      Biotechnology, Health Care, Medical
                                  ...
3994                                          Venture Capital
3995    Biotechnology, Health Care, Life Science, Phar...
3996                   Non Profit, STEM Education, Women's
3997    Biopharma, Biotechnology, Health Care, Pharmac...
3998    Advanced Materials, Health Diagnostics, Pharma...
Name: Industries, Length: 3999, dtype: object
```

```
def industries_cut(col):
    new_text = []
    for text in col:
        sep = ','
        text = text.split(sep, 1)[0]
        new_text.append(text)
    #print(val,' and ', Acquisition_Status[i], ' = ', trgt[i])

    new_text = pd.Series(new_text)
    return new_text
data['first_industry'] = industries_cut(data['Industries'])
data = data.drop('Industries', 1)
data['first_industry']
```

```
0              Biotechnology
1              Biotechnology
2              Biotechnology
3              Biotechnology
4              Biotechnology
                  ...
3994         Venture Capital
3995           Biotechnology
3996              Non Profit
3997               Biopharma
3998    Advanced Materials
Name: first_industry, Length: 3999, dtype: object
```

Затем к обработанным данным были применены различные методы машинного обучения.

```
[146] print_accuracy(target_logistic_regression, Y_test)

      accuracy = 0.8251912889935256, balanced accuracy = 0.5,
      precision = 0.0, F1-score = 0.0

[147] print_accuracy(target_random_forest, Y_test)

      accuracy = 0.8251912889935256, balanced accuracy = 0.5,
      precision = 0.0, F1-score = 0.0

      print_accuracy(target_naive_bayes, Y_test)

      accuracy = 0.20158917010005886, balanced accuracy = 0.5016312675014529,
      precision = 0.1752988047808765, F1-score = 0.296603577910293

[149] print_accuracy(target_gradient_boosting, Y_test)

      accuracy = 0.8248969982342554, balanced accuracy = 0.501148551612175,
      precision = 0.4, F1-score = 0.00667779632721202

Сравнение
Градиентый бустинг
accuracy = 0.8248969982342554, balanced accuracy = 0.501148551612175,
precision = 0.4, F1-score = 0.00667779632721202
Ада буст
accuracy = 0.8257798705120659, balanced accuracy = 0.5030103699861189,
precision = 0.6666666666666666, F1-score = 0.013333333333333332
XGB
accuracy = 0.8163625662154208, balanced accuracy = 0.5132266555233744,
precision = 0.32558139534883723, F1-score = 0.08235294117647059
Cat
accuracy = 0.8201883460859329, balanced accuracy = 0.5195253774069751,
precision = 0.4, F1-score = 0.10014727540500737
```

Выводы

Лучшие результаты показал бустинг, что говорит о возможном наличии сложных взаимосвязей в датасете. Далее будет продолжен подбор гиперпараметров для поиска наилучшей модели предсказания возможного успеха стартапа.