

Определение доходности облигаций по данным ММВБ



Куликова Наталия Владимировна

октябрь'21



Проект ПЕРЕЗАПУСК

О себе

- Куликова Наталия Владимировна
- г. Москва
- Высшее экономическое
- Опыт в Сбере (Корпоративный блок - Московский и Среднерусский ТБ, кредитный инспектор, рассмотрение сделок по кредитованию юридических лиц, финансовый и комплексный анализ Заемщика/Группы/Холдинга, презентация кредитных сделок на КПКИ ТБ , ЦА , работа с проблемной задолженностью. Учетные системы - СПАРК, CRM, ЕКС , РЕГА, ЕФС / процессы - НКП 2.0, К7М, Простые сделки
-  8 960 506 13 54
-  kulikova.n.vladimiro@sberbank.ru

Описание проекта

Суть проекта :

Провести оценку рынка рублевых облигаций по распределению купонной доходности, провести кластеризацию ценных бумаг по купонной доходности, предложить графическое представление полученных групп/кластеров.

Используемый функционал:

- Python 3
- Jupyter Notebook

Ссылка на репозиторий с кодом:

https://github.com/Natalia251976/PROJECT_DA

План

1. Загрузка данных с сайта ММВБ

<https://iss.moex.com/iss/apps/infogrid/emission/rates.csv>

2. Работа с полученными данными:

- *Анализ*
- *Очистка и перевод данных в нужный формат*

3. Расчет показателей доходности

4. Построение диаграмм

Модель данных

Результат запроса к <https://iss.moex.com/iss/apps/infogrid/emission/rates.csv>
Dataframe (Bond)

Ввод [1179]: `bond.head()`

Out[1179]:

	SECID	SHORTNAME	NAME	TYPENAME	ISIN	REGNUMBER	LISTLEVEL	FACEVALUE	FACEUNIT	ISSUESIZE	...	RTH2
0	AMUNIBB2AER2	UBANK11/22	UNIBANK OJSC Series 1	Корпоративные еврооблигации	AMUNIBB2AER2	NaN	3.0	100,0	USD	10000.0	...	14457,44
1	AMUNIBB2DER6	UBANK02/24	UNIBANK OJSC Series 1 21	Корпоративные еврооблигации	AMUNIBB2DER6	NaN	3.0	100,0	USD	33478.0	...	14983,58
2	CH0248531110	VTB-24 CHF	VTB CAPITAL S.A. 24 CHF	Корпоративные еврооблигации	CH0248531110	NaN	NaN	5000,0	CHF	70000.0	...	808520,36
3	CH0346828400	GAZ-21 CHF	GAZ CAPITAL SA 2.75 30/11/21	Корпоративные еврооблигации	CH0346828400	NaN	NaN	5000,0	CHF	100000.0	...	463890,63 3
4	CH0374882816	GAZ-22 CHF	GAZ CAPITAL SA 2.25 19/07/22	Корпоративные еврооблигации	CH0374882816	NaN	NaN	5000,0	CHF	100000.0	...	460076,83 3

```
bond.shape
```

```
(2695, 53)
```

Оставляем только рублевые облигации:

```
bond_rub = bond.drop(bond[bond['FACEUNIT'] != 'RUB'].index)
```

Модель данных

Результат анализа используемого Dataframe:

~ 80% - object
~ 20% - float64

Процент строк, содержащих Nan: **100%**

Вывод:

1. df.dropna() - не используем.

Удалять Nan будем только из полей, которые используются в расчетах.

```
bond_rub = bond_rub.dropna(subset = ['COUPONLENGTH', 'COUPONPERCENT', 'COUPONFREQUENCY', 'PRICE', 'COUPONVALUE'])  
bond_rub = bond_rub.reset_index(drop=True)
```

2. Приводим поля, необходимые для расчетов к соответствующим типам (float64, datetime)

bond_rub2.isna().sum()

SECID	0
SHORTNAME	0
NAME	0
FACEVALUE	0
FACEUNIT	0
DURATION	670
IS_QUALIFIED_INVESTORS	0
HIGH_RISK	0
COUPONFREQUENCY	355
COUPONDAYSAPASSED	5
COUPONPERCENT	60
COUPONLENGTH	5
COUPONVALUE	63
OFFERDATE	1802
EMITENTNAME	142
INN	142
PRICE	157
PRICE_RUB	157
COUPONDATE	5
ISSUEDATE	142
dtype: int64	

bond.dtypes

SECID	object
SHORTNAME	object
NAME	object
TYPENAME	object
ISIN	object
REGNUMBER	object
LISTLEVEL	float64
FACEVALUE	object
FACEUNIT	object
ISSUESIZE	float64
IS_COLLATERAL	int64
IS_EXTERNAL	int64
PRIMARY_BOARDID	object
PRIMARY_BOARD_TITLE	object
MATDATE	object
IS_RII	object
ISSUEDATE	object
EVENINGSESSION	int64
DURATION	float64
IS_QUALIFIED_INVESTORS	int64
HIGH_RISK	int64
COUPONFREQUENCY	float64
YIELDATWAP	object
COUPONDATE	object
COUPONPERCENT	object
COUPONVALUE	object
COUPONDAYSAPASSED	float64
COUPONDAYSREMAIN	float64
COUPONLENGTH	float64

Модель данных

При приведении полей к типу float64 возникла ошибка - наличие "мусорных" символов в полях Dataframe

Решение:

Очистка полей ['PRICE_RUB','PRICE','FACEVALUE','COUPONPERCENT','COUPONVALUE'] с использованием регулярного выражения, замена «,» на «.» и приведение к соответствующему типу

```
bond_rub[i].replace(regex=True, inplace=True, to_replace=r'^0-9,\\.\\-]', value=r'')  
bond_rub[i] = bond_rub[i].str.replace(',', '.').astype(float)
```

Приводим формат столбцов с датами к типу *datetime*

```
bond_rub['COUPONDATE'] = pd.to_datetime(bond_rub['COUPONDATE'])
```

```
bond_rub[['date_NEW', 'ISSUEDATE', 'COUPONDATE']].dtypes
```

```
date_NEW      datetime64[ns]  
ISSUEDATE     datetime64[ns]  
COUPONDATE    datetime64[ns]  
dtype: object
```

Модель данных

На основании анализа рынка доходных облигаций в 2021 (длинных и коротких) определено, что эффективная доходность к погашению в % годовых колеблется от 4.2 до 8.5 %. На основании вышеприведенных данных, оставляем значения эффективной доходности не менее ключевой ставки ЦБ (KeyRate).

Проводим импорт Ключевой ставки с сайта ЦБ РФ и обрабатываем данные

```
bond_rub = bond_rub[(bond_rub['COUPONPERCENT'] >= KeyRate) & (bond_rub['PROFIT'] > 0)]
```

```
bond_rub['PROFIT'].max()
```

```
14.117652068022926
```

Результат обработки исходного Datframe:

```
bond_rub.shape
```

```
(904, 61)
```


Расчеты

На основании полученных данных проводим расчет показателя «Доходность ценных бумаг» через промежуточные показатели:

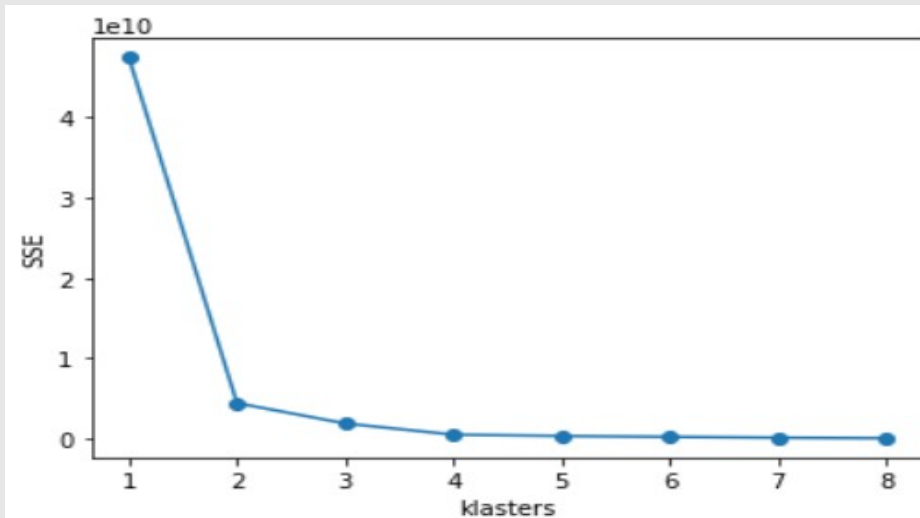
- *Купонный доход в рублях*
- *НКД в рублях*
- *Прибыль/убыток в рублях*
- *Купонный доход с уплатой 13% НДФЛ в рублях*

$$\text{Процентный доход} = \frac{\left(\text{Цена}_{\text{Номинальная}} - \text{Цена}_{\text{покупки}} - \text{НКД} + 0.87 * \text{Купонный}_{\text{доход}} * X_{\text{лет}} \right) * 100 \%}{\left(\text{Цена}_{\text{покупки}} + \text{НКД} \right)}$$

(комиссией брокера пренебрегаем)

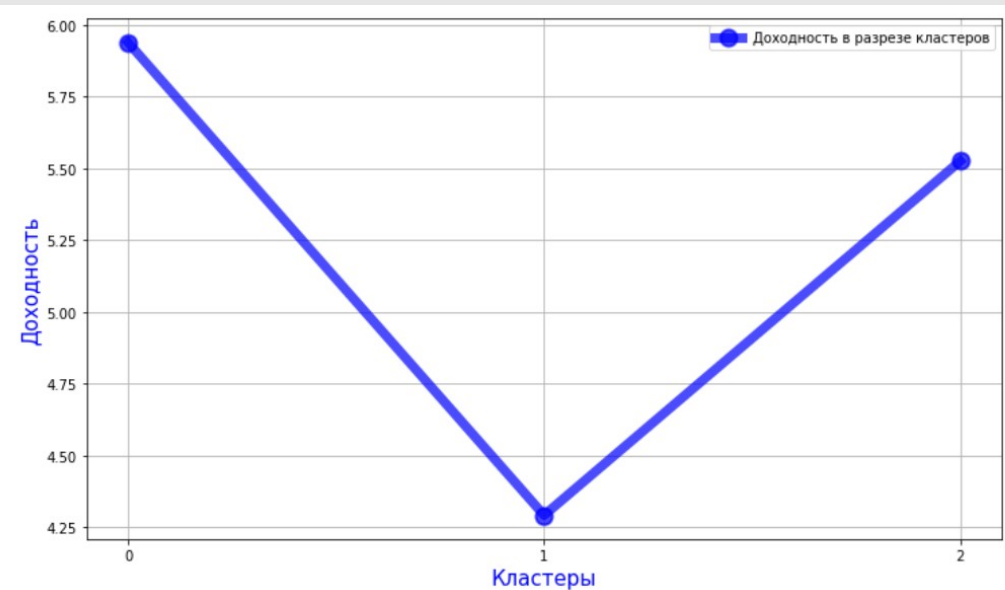
Расчеты

Проводим кластеризацию анализируемых данных , используя «Метод локтя» на основании показателей доходности :



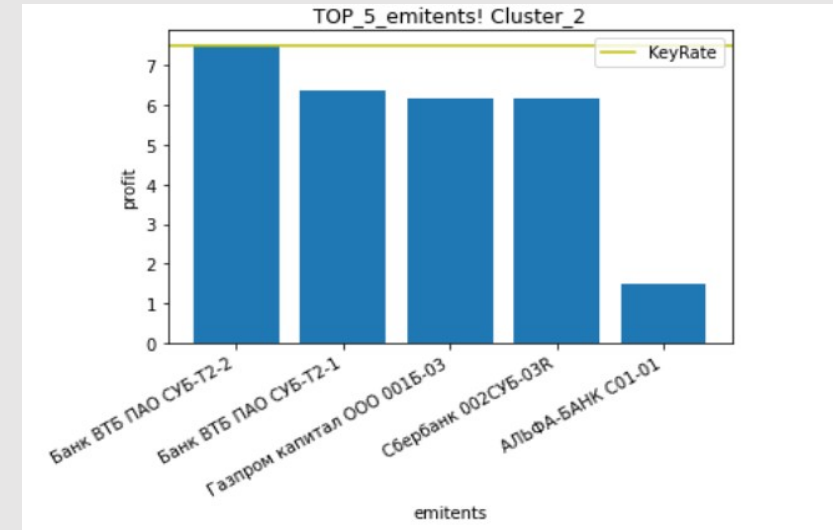
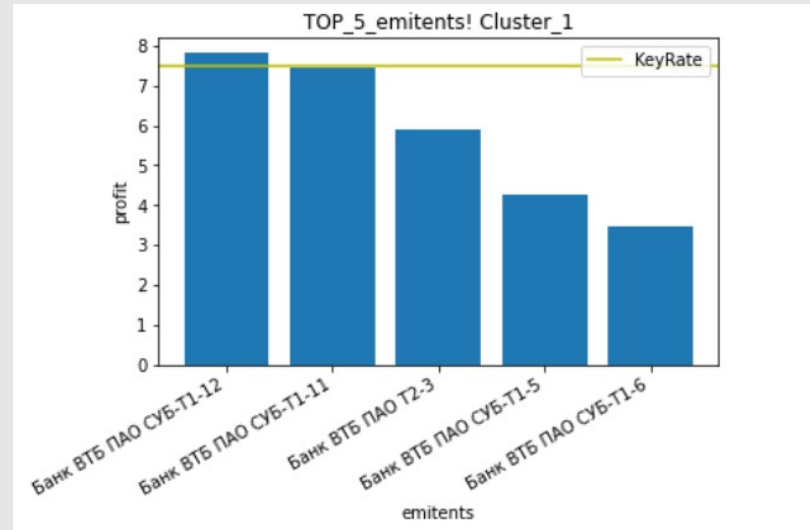
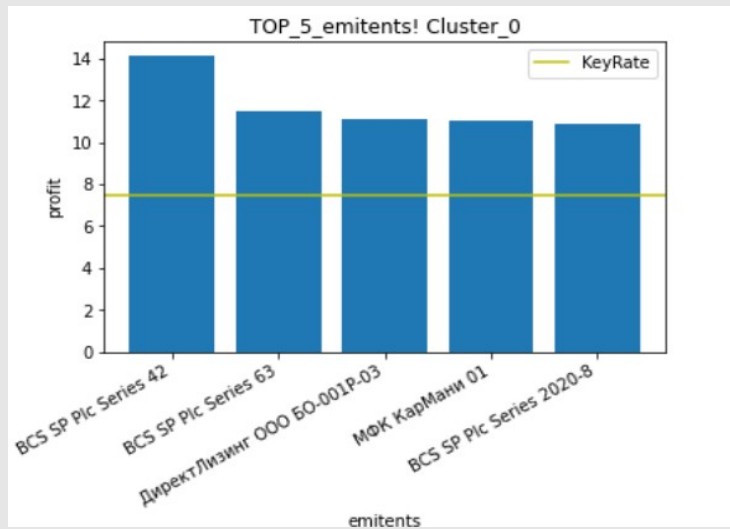
Вывод - делим на 3 кластера

Строим диаграмму распределения кластеров по средним значениям доходности



Результаты

Строим диаграммы по Топ 5 облигаций/эмитентов в разрезе кластеров



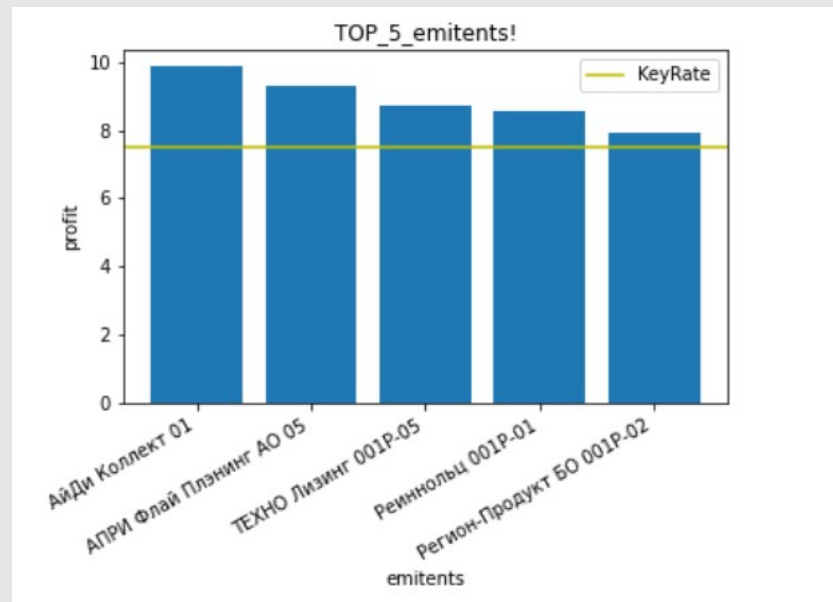
Результаты

Проводим расчет доходности последующих купонов по низкорисковым рублевым облигациям с датой промежуточной выплаты по купонам не более 20% купонного периода назад до текущей даты и облигациям выпущенным в аналогичный период с учетом разницы номинала и цены продажи, а также с учетом налоговых отчислений на период год с текущей дата.

Решение:

Оставляем данные, удовлетворяющие условию: "COUPONDATE" и "ISSUEDATE" меньше или равны "date_NEW " и "HIGH_RISK" = 0

Определяем Топ 5 эмитентов по доходности и строим диаграмму.



Используемые технологии

- **Используемые технологии:**

- Python 3
- Pandas
- Numpy
- Sklearn
- Matplotlib
- BeautifulSoup

СПАСИБО ЗА ВНИМАНИЕ!