

2018 年“花旗杯”金融创新应用大赛

专项计划仿真流程报告



参赛题目：砺金—住房租赁资产证券化 REITs 平台

大赛队长：褚天硕

辅导老师：隋 聪、高 明

隶属学校：东北财经大学

目录

1. 第一部分：引言	3
1.1 编写目的	3
1.2 背景	3
2. 第二部分：金融理论背景	3
2.1 修正后的 score 模型	3
2.2 改进的 KMV 模型	4
2.3 REITs 估值模型-营运现金流贴现法	4
3. 第三部分：仿真流程	5
3.1 企业端流程	5
3.2 后端流程	8

1. 第一部分：引言

1.1 编写目的

专项计划流程仿真报告是在项目开发之后，针对砺金平台具体功能的预期再现。专项计划流程仿真，检验砺金平台以大数据技术为基础开发的动态风险评控功能；同时也有助于检查砺金平台设计的机器学习模型与一系列金融模型是否能够按预期流程正常交接运作，花旗 API 的调用是否完善，即能否实现企业准入、风险评控、以及 REITs 基金估值、客户信息获取等一系列流程。

1.2 背景

本平台（砺金平台）是一个致力于拓宽国内住房租赁企业的融资渠道、丰富并完善房地产价值链，为个体投资者提供投资于房地产市场的全新方式的综合性服务平台。平台一端连接企业，一端连接投资者，以私募基金加专项计划的方式，既为企业提供了优质低成本的融资又为投资者提供了低风险高收益的投资项目。

砺金平台共有两个端口，一个后台，两个端口分别连接着申请 REITs 专项计划的企业和投资者。平台由四部分构成，分别是企业审核、风险评控、设立 REITs 专项计划和资金划转。SPV 端在接收企业端提交过来的财务数据后，通过机器学习对企业进行融资资格准入审核；再利用修正后的 Z-score 模型和改进后的 KMV 模型、“爬虫”技术抓取企业实时信息计算出企业的风险指数；再经由修正好的美国 REITs 估值模型进行估值得出基金发行规模。

2. 第二部分：金融理论背景

2.1 修正后的 score 模型

Z-score 模型于 20 世纪 60 年代由 altman 首次提出，其分析对象为美国破产和非破产企业。但是，在我国，由于经济体制的不同，破产不等同于停业，所以将 Z-score 模型用于判别企业是否存在财务危机更加适合我国国情。除此外，我国多位学者对 Z-score 模型在我国的适用性进行了大量分析研究，发现不同行

业的 Z-score 模型具有显著差异，且随时间推移，用于量化风险的财务指标也应进行修正。本模块旨在量化我国房地产企业财务风险，故对 Z-score 模型的修正是十分有必要的。

2.2 改进的 KMV 模型

KMV 模型因其诸多优点而成为目前国际金融界最流行的风险管理模型之一。首先，KMV 可以充分利用资本市场上的信息，对所有公开上市企业进行信用风险的量化和分析；其次，由于该模型所获取的数据来自股票市场的资料，因而更能反映企业当前的信用状况，具有前瞻性，其预测能力更强、更及时，也更准确；另外，KMV 模型建立在当代公司理财理论和期权理论的基础之上，具有很强的理论基础做依托。因此我们采用 KMV 模型来预测单个企业违约风险。

2.3 REITs 估值模型-营运现金流贴现法

营运现金流贴现法由现金流贴现法演化而来。该估值方法是对物业产生的营运现金流建立一定假设构建财务模型，贴现加总物业后续各期现金流量，预测期末终值的估计金额。

其公式如下：

$$V = \sum_{i=1}^n \frac{FFO_i}{(1+r)^i}$$

$$WACC \equiv KD \times \frac{TD}{V} + KE \times \frac{E}{V} \times (1 - T_C)$$

净收入=所有收入（包括资本利得）-运营费用-折旧-摊销-利息支出-一般化管理费用

FFO=净收入-房地产出售中的资本利得+房地产折旧费用

其中， FFO_i 是第 i 年的营运现金流； r 是折现率，一般用 WACC 表示； KD 是债务资本成本； KE 是权益资本成本； V 是总成本； TD 是债务资本总额； E 是权益资本总额， T_C 是税率，一般取企业所得税税率。

该方法在理论上是十分合理的，同时研究结果也显示，过去 20 年中，美国权益型 REITs 的价格指数与租金指数的相关系数达到 0.72。对于一家财务规范

的 REITs 公司来说，做 FF0 的预测并不困难，因为预测的结果基于公司既有的财务数据，可信度较高。但以这种方式折现 FF0 在一定程度上夸大了价值，因为投资者并没有像这种方法所暗含的一样提前收到所有未来 FF0，投资者收到的只是 REITs 的现金分红，剩余 FF0 则被留存以促进未来增长。由于现实中债务利率通常低于收益率，REITs 很容易通过提高负债率特别是低成本可变利率债务来“购买”FF0 增长，从而使得 REITs 价值被高估。

3. 第三部分：仿真流程

3.1 企业端流程

3.1.1 仿真流程

我们采用了 13 项指标，根据 57 家公司的财报数据进行机器学习。

```

1 - A = xlsread('D:\pingji.xlsx');
2 - n=13;
3 - for i=1:n
4 -     A(i,:)=(A(i,:)-min(A(i,:)))/(max(A(i,:))-min(A(i,:)));
5 - end
6 - A=A';    %%%取转置矩阵
7 - AA=A([1:1:30],:);
8 - net=newsom(minmax(AA),[1 3]);
9
10 - a=[1000 2000 3000 4000 5000 6000];
11 - for i=1:6
12 -     net.trainParam.epochs=a(i);
13 -     net=train(net,AA);
14 -     y=sim(net,AA);
15 -     yc=vec2ind(y);
16 - end
17
18 - pt=AA;
19 - C=yc;
20 - T=ind2vec(C);
21 - n1=1;
22 - n2=1;n3=1;n4=1;
23 - n5=1;n6=1;n7=1;
24 - n8=1;n9=1;n10=1;
25 - n11=1;n12=1;n13=1;
26 - a1=[];a2=[];a3=[];
27 - a4=[];a5=[];a6=[];
28 - a7=[];a8=[];a9=[];

```

```

56 -         if C(i)==7
57 -             a7(n7)=i;
58 -             n7=n7+1;
59 -         end
60 -         if C(i)==8
61 -             a8(n8)=i;
62 -             n8=n8+1;
63 -         end
64 -         if C(i)==9
65 -             a9(n9)=i;
66 -             n9=n9+1;
67 -         end
68 -         if C(i)==10
69 -             a10(n10)=i;
70 -             n10=n10+1;
71 -         end
72 -         if C(i)==11
73 -             a11(n11)=i;
74 -             n11=n11+1;
75 -         end
76 -         if C(i)==12
77 -             a12(n12)=i;
78 -             n12=n12+1;
79 -         end
80 -         if C(i)==13
81 -             a13(n13)=i;
82 -             n13=n13+1;

```

在读取 57 家公司的数据后，用 SOM 进行机器学习。然后用 LVQ 方法进行进一步信用分级：

```

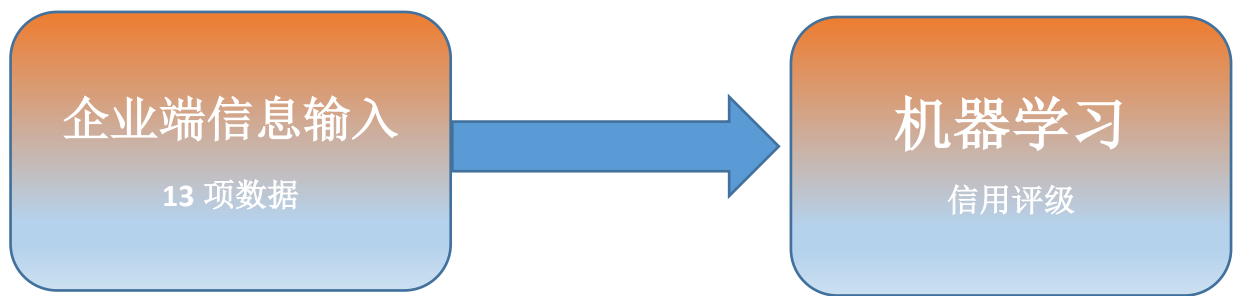
1 - Pt=PP;
2 - C=yc;
3 - T=ind2vec(C);
4 - n1=1;
5 - n2=1;n3=1;n4=1;
6
7 - a1=[];a2=[];a3=[];
8 - a4=[];
9 - for i=1:57
10 -     if C(i)==1
11 -         a1(n1)=i;
12 -         n1=n1+1;
13 -     end
14 -     if C(i)==2
15 -         a2(n2)=i;
16 -         n2=n2+1;
17 -     end
18 -     if C(i)==3
19 -         a3(n3)=i;
20 -         n3=n3+1;
21 -     end
22 -     if C(i)==4
23 -         a4(n4)=i;
24 -         n4=n4+1;
25 -     end
26 - end
27 - B=[(n1-1);(n2-1);(n3-1);(n4-1)]'/57;

28 - lvqnet=newlvq(minmax(Pt),4,B);
29 - lvqnet.IW{1,1}=net.IW{1,1};
30 - a=[20 40 60 80 100];
31 - for i=1:5
32 -     lvqnet.trainParam.epochs=a(i);
33 -     lvqnet=train(lvqnet,Pt,T);
34 -     yy=sim(lvqnet,Pt);
35 -     C=vec2ind(yy);
36 - end
37
38

```

如果达到信用标准，将进入下一流程；未达到标准的，将进行线下审核。

3.1.2 流程图



3.2 后端流程

3.2.1 仿真流程

当企业端的信息传过来之后，进行如下流程：

- 1、调用 KMV 算法，根据传入的数据计算风险值：

```

1  function F=KMVfun(EtoD, r, T, EquityTheta, x)
2  —   d1=(log(x(1)*EtoD)+(r+0.5*x(2)^2)*T)/(x(2)*sqrt(T));
3  —   d2=d1-x(2)*sqrt(T);
4  —   F=[x(1)*normcdf(d1)-exp(-r*T)*normcdf(d2)/EtoD-1;
5     —   normcdf(d1)*x(1)*x(2)-EquityTheta];
6  —   end

1  function [Va, AssetTheta]=KMVOptSearch(E, D, r, T, EquityTheta)
2  —   EtoD=E/D;
3  —   x0=[1, 1];
4  —   VaThetaX=fsolve(@(x) KMVfun(EtoD, r, T, EquityTheta, x), x0);
5  —   Va=VaThetaX(1)*E;
6  —   AssetTheta=VaThetaX(2);
7
8  —   end
    
```



```

1 -      r=0.03551;
2 -      T=1 ;
3 -      SD=31800014304.2900;%流动负债
4 -      LD=18189406581.0500;%长期负债
5 -      DP=1.108*SD+1.126*LD;%修正后DP
6 -      D=DP;|
7 -      PriceTheta=0.074676757;%日波动率
8 -      EquityTheta=0.074676757*sqrt(252);%年化波动率
9 -      E=4150345758.1967;%总市值均值
10 -      [Va,AssetTheta]=KMV0ptSearch(E,D,r,T,EquityTheta);
11 -      DD=(Va-DP)/(Va*AssetTheta); %计算违约距离
12 -      EDF=normcdf(-DD); %计算违约率
13

```

2、计算 Z 值，分析财务状况：

```

1 package com.Servlet;
2
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import javax.servlet.annotation.WebServlet;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9
10 @WebServlet("/listZscoreServlet")
11 public class ListZscoreServlet extends HttpServlet {
12     private static final long serialVersionUID = 1L;
13
14     private double x;
15     private double y;
16     private double z;
17
18     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
19         // TODO Auto-generated method stub
20
21         x=(double) request.getAttribute("A");
22         y=(double) request.getAttribute("B");
23         z=(double) request.getAttribute("C");
24
25         request.setAttribute("zscore",zs(x,y,z) );
26         request.getRequestDispatcher("zscore.jsp");
27     }
28
29     public String zs(double a, double b,double c) {
30         double z;
31         z=-3.812+4.721*a+9.078*b+0.039*c;
32
33         if(z>0.747048)
34             return ("财务健康");
35         else if(-0.74173<z)
36             return ("灰色地带");
37         else if(z<-0.74173 )
38             return ("财务危机");
39         return null;
40     }
41

```

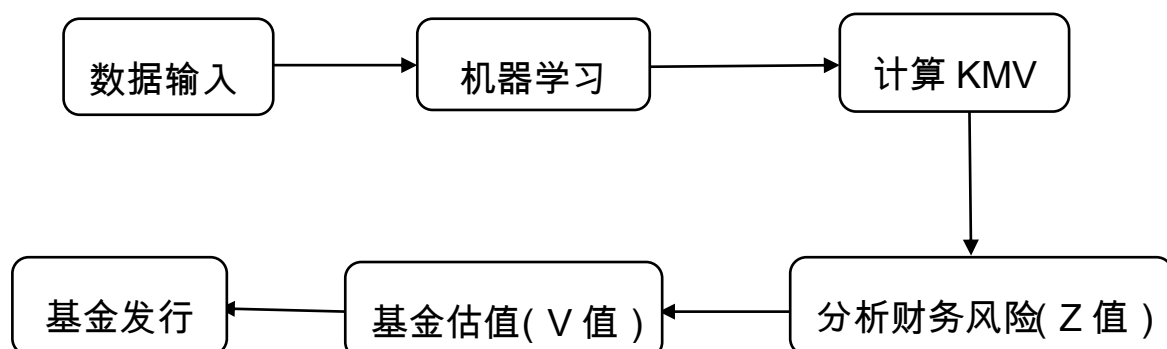
3、调用估值算法，计算基金预计价值 V:

```

1 package com.Servlet;
2
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import javax.servlet.annotation.WebServlet;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9
10
11 @WebServlet("/listV")
12 public class ListVServlet extends HttpServlet {
13     private static final long serialVersionUID = 1L;
14     private double RF;
15     private double Beta;
16     private double RM;
17     private double KD;
18     private double TD;
19     private double Tc;
20     private double V1;
21     private double TE;
22
23
24
25     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
26         // TODO Auto-generated method stub
27
28         RF=(double) request.getAttribute("RF");
29         Beta=(double) request.getAttribute("Beta");
30         RM=(double) request.getAttribute("RM");
31         KD=(double) request.getAttribute("KD");
32         TD=(double) request.getAttribute("TD");
33         Tc=(double) request.getAttribute("Tc");
34         V1=(double) request.getAttribute("V1");
35         TE=(double) request.getAttribute("TE");
36
37         double AFFO[] = new double[5];
38
39         request.setAttribute("v", comV(RF,Beta, RM, KD, TD, Tc, AFFO, V1, TE));
40         request.getRequestDispatcher(".jsp");
41     }
42
43
44     public double comV(double RF,double Beta,double RM,double KD,double TD,double Tc,double AFFO[],double V1,double TE)
45     {
46         double KE;
47         double r;
48         double V=0;
49         int i=AFFO.length;
50         KE=RF+Beta*(RM-RF);
51         r=KD*(TD/V1)+KE*(TE/V1)*(1-Tc);
52         for(int j=0;j<i;j++)
53         {
54             V+=AFFO[j]/Math.pow((1+r), j);
55         }
56         return V;
57     }
58 }
59
60

```

3.2.2 流程图



附录一 爬虫 Python 代码实现

```

1 from bs4 import BeautifulSoup
2 import requests
3 from parsel import Selector
4 import pandas as pd
5 import time
6 from pandas.core.frame import DataFrame
7 from fake_useragent import UserAgent
8 ua = UserAgent()
9 headers={
10     "User-Agent":ua.ie
11 }
12 requests.adapters.DEFAULT_RETRIES = 40

14 def j_par(url):
15     wr=requests.get(url,headers=headers,stream=True)
16     sel=Selector(wr.text)
17     soup = BeautifulSoup(wr.text, 'lxml')
18     oth= soup.find_all(class_="special clearfix")
19     oth_2=soup.find_all(class_="des")
20
21     '''房源名'''
22     h_nam=sel.xpath('//div[@class="content"]/div[@class="wrap pb80"]/ul[@class="fyuan-list"]/li/a/div[@class="des"]/p[@class="tit"]/text()').extract()
23
24     '''房型'''
25     h_typ=sel.xpath('//div[@class="content"]/div[@class="wrap pb80"]/ul[@class="fyuan-list"]/li/a/div[@class="des"]/div[@class="room"]/span[1]/text()').extract()
26
27     '''面积'''
28     h_siz=sel.xpath('//div[@class="content"]/div[@class="wrap pb80"]/ul[@class="fyuan-list"]/li/a/div[@class="des"]/div[@class="room"]/span[2]/text()').extract()
29
30     '''楼层'''
31     h_flo=sel.xpath('//div[@class="content"]/div[@class="wrap pb80"]/ul[@class="fyuan-list"]/li/a/div[@class="des"]/div[@class="room"]/span[3]/text()').extract()
32
33     '''优势'''
34     h_sep=[]
35     for tag in oth:
36         if tag.get_text()=="\n":
37             h_sep.append(" ")
38         else:
39             h_sep.append(tag.get_text().strip().replace('\n','').replace('\n',''))
40
41     h_tra2=[]

```

```

43 for tag in oth_2:
44     if tag.find(class_="distance"):
45         h_tra2.append(tag.find(class_="distance").get_text().replace('\n',''))
46     else:
47         h_tra2.append(" ")
48
49 '''价格'''
50 h_pri=sel.xpath('//div[@class="content"]/div[@class="wrap pb80"]/ul[@class="fyuan-list"]/li/a/div[@class="price"]/span[@class="num"]/text()).extract()
51
52 '''房源号'''
53 a=soup.find_all("a",class_="clearfix")
54 h_cod=[]
55 for t in a:
56     h_cod.append(t.get('href'))
57
58 '''地址&交通及周边设施'''
59 h_add=[]
60 h_des=[]
61 for c in h_cod:
62     ur = "http://zufang.jiwu.com"+c
63     wr = requests.get(ur,headers=headers,stream=True)
64     wesoup=BeautifulSoup(wr.text,'lxml')
65     wesoup_1=wesoup.find_all('td') #用于地址
66     wesoup_2=wesoup.find('p',class_='fwms').get_text() #用于设施
67
68     ''' 地址 '''
69     td_content=[]
70     for i in wesoup_1:
71         td_content.append(i.text)
72     length=len(wesoup_1)
73     wetd=td_content[length-1]#筛选地址
74     h_add.append(wetd)
75
76     '''交通及周边设施'''
77     wesoup_2_des=wesoup_2
78     h_des.append(wesoup_2_des)
79
80 #将列表转换成字典
81 c={"h_cod" : h_cod,
82    "h_nam" : h_nam,
83    "h_loc" : h_add,
84    "h_typ" : h_typ,
85    "h_siz" : h_siz,
86    "h_pri" : h_pri,
87    "h_flo" : h_flo,
88    "h_sep" : h_sep,
89    "h_des" : h_des}
90
91 #将字典转换成数据框
92 pages_info=DataFrame(c)
93
94 return pages_info
95
96 def pp(pages):
97     count=0
98     for page in pages:
99         a=j_par(page)
100         count=count+1
101         print ('the '+str(count)+' page is sucessful')
102         time.sleep(5)
103         l=pd.DataFrame(columns=['h_cod','h_nam','h_loc','h_typ','h_siz','h_flo','h_pri','h_sep','h_des'])
104         l=pd.concat([l,a],ignore_index=True)
105     return l
106
107 for p in range(1,8):
108     pages=[]
109     if p==1:
110         pages.append('http://zufang.jiwu.com/gz/gongyu46/')
111     else:
112         pages.append('http://zufang.jiwu.com/gz/gongyu46/page'+str(p)+'/')
113
114     lj=pp(pages)
115     lj.to_csv('d:\\boyu_gz_futian_10__'+str(p)+'.csv')

```