企业信息化绩效评价系统软件源程序

# manage.py

import os

import sys

if \_\_name\_\_ == "\_\_main\_\_":

os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "evasys.settings")

try:

from django.core.management import execute\_from\_command\_line

except ImportError:

try:

import django

except ImportError:

raise ImportError(

"Couldn't import Django. Are you sure it's installed and "

"available on your PYTHONPATH environment variable? Did you "

"forget to activate a virtual environment?"

)

raise

execute\_from\_command\_line(sys.argv)

# admin.py

from \_\_future\_\_ import unicode\_literals

from django.contrib import admin

from .models import users,company,industry,evaluator,period,senior\_index,junior\_index

from .models import interval,ind\_model,val\_model

admin.site.register(users)

admin.site.register(company)

admin.site.register(industry)

admin.site.register(evaluator)

admin.site.register(senior\_index)

admin.site.register(junior\_index)

admin.site.register(period)

admin.site.register(ind\_model)

admin.site.register(val\_model)

admin.site.register(interval)

# views.py

from \_\_future\_\_ import unicode\_literals

from django.shortcuts import render,redirect,HttpResponse

import time

import os

from sympy import \*

import xlrd

from evasysapp.forms import UploadExcelForm

from random import Random

from django.core.mail import send\_mail

from evasysapp.models import EmailVerifyRecord

from evasys.settings import EMAIL\_FROM

from evasysapp.utils\_ import \*

from io import BytesIO

import base64

from xpinyin import Pinyin

from evasysapp.utils import ExtAssMethod

import json

import matplotlib.pyplot as plt

import matplotlib

matplotlib.rcParams['font.sans-serif'] = ['SimHei']

matplotlib.rcParams['font.family']='sans-serif'

matplotlib.rcParams['axes.unicode\_minus'] = False

class MyEncoder(json.JSONEncoder):

def default(self, obj):

if isinstance(obj, np.integer):

return int(obj)

elif isinstance(obj, Float):

return float(obj)

elif isinstance(obj, np.ndarray):

return obj.tolist()

else:

return super(MyEncoder, self).default(obj)

def login(request):

error\_msg = ''

if request.method =='POST':

username = request.POST.get('username',None)

password = request.POST.get('password',None)

if models.users.objects.filter(user\_name = username,password = password):

user\_id = models.users.objects.filter(user\_name = username).values("user\_id")

user\_id = user\_id[0]['user\_id']

request.session['username'] = username

request.session['user\_id'] = user\_id

request.session['is\_login'] = True

return redirect('http://127.0.0.1:8000/evasys/home/')

else:

error\_msg = '用户名或密码错误'

return render(request,'login.html',{'error\_msg':error\_msg})

def home(request):

msgs = "游客您好,您现在使用的是试用版，若要使用完整功能请您："

ors = '或者'

logins = '登录'

register = '注册'

hidden = "hidden"

if request.method=='GET':

try:

request.session['is\_login']

except:

return render(request, 'home.html', {'huanying': msgs, 'ors': ors, 'login': logins, 'register': register,"hidden":hidden})

if request.session['is\_login']:

username = request.session['username']

msg = '欢迎您,' + str(username)

return render(request, 'home.html', {'huanying': msg})

else:

return render(request, 'home.html', {'huanying': msgs, 'ors': ors, 'login': logins, 'register': register,"hidden":hidden})

else:

pass

def detail(request):

nid = request.GET.get('nid')

return HttpResponse(nid)

def register(request):

industry\_list = models.industry.objects.all()

users = models.users.objects.all()

if request.method =='POST':

companyname = request.POST.get('companyname', None)

liscom\_code = request.POST.get('liscom\_code', None)

industry\_id = int(request.POST.get('industry', None))

telephone = request.POST.get('telephone', None)

company\_address = request.POST.get('company\_address', None)

introduction = request.POST.get('introduction', None)

website = request.POST.get('website', None)

docs = request.FILES.get('docs',None)

imgs = request.FILES.get('imgs',None)

username = request.POST.get('username', None)

email = request.POST.get('email', None)

password = request.POST.get('password', None)

ecode = request.POST.get('emailcode',None)

obj = models.EmailVerifyRecord.objects.get(email=email)

vcode = obj.vcode

send\_time = obj.send\_time.replace(tzinfo=None)

user\_name = obj.user\_name

nowTime = datetime.datetime.now()

interval\_time = nowTime-send\_time

if ecode == vcode and interval\_time.seconds < 300:

company\_docs\_path = None

company\_imgs\_path = None

for item in users:

if item.user\_name == user\_name:

msg = '该用户已经存在，请重新输入用户名'

return render(request, 'register.html',{'users\_msg':msg})

if docs:

company\_docs\_path = os.path.join('upload/company/docs', docs.name)

docfile = open(company\_docs\_path, 'wb')

for i in docs.chunks():

docfile.write(i)

docfile.close()

if imgs:

company\_imgs\_path = os.path.join('upload/company/imgs',imgs.name)

imgfile = open(company\_imgs\_path,'wb')

for i in imgs:

imgfile.write(i)

imgfile.close()

times = time.strftime('%Y%m%d%H', time.localtime())

p = Pinyin()

upy = p.get\_initials(username, u'')

cpy = p.get\_initials(companyname, u'')

company\_id = times+upy

user\_id = times+cpy

models.company.objects.create(company\_id = company\_id,company\_name = companyname,liscom\_code = liscom\_code,

industry\_id\_id = industry\_id,telephone = telephone,company\_address = company\_address,

introduction = introduction,website = website,img\_address = company\_imgs\_path,

intro\_address = company\_docs\_path)

try:

user = models.users(user\_id = user\_id,user\_name =username,email = email,password= password,

user\_type = 1)

user.save()

except:

models.company.objects.filter(company\_id=company\_id).delete()

return redirect('/evasys/sucregister/')

elif vcode!= ecode:

return render(request,'register.html',{'err\_msg':'验证码错误','industry':industry\_list})

else:

return render(request,'register.html',{'err\_msg':'验证码失效','industry':industry\_list})

elif request.method =='GET':

return render(request,'register.html',{'industry':industry\_list})

def sucregister(request):

if request.method == "GET":

meta = request.META

if meta.get("HTTP\_REFERER")== "http://127.0.0.1:8000/evasys/register/":

return render(request,'sucregister.html')

else:

return render(request,'NoPage.html')

else:

pass

def sendecode(request):

user\_name = request.POST.get("user\_name",None)

if len(user\_name) ==0:

msg = "用户名不能为空"

return HttpResponse(msg)

email = request.POST.get("email")

users = models.users.objects.all()

for item in users:

if user\_name == item.user\_name:

msg = '该用户已经存在，请重新输入用户名'

return HttpResponse(msg)

if email == item.email:

msg = '邮箱已被注册'

return HttpResponse(msg)

def random\_vcode(randomlength=6):

vcode = ''

chars = 'AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz0123456789'

length = len(chars) - 1

random = Random()

for i in range(randomlength):

vcode += chars[random.randint(0, length)]

return vcode

def send\_register\_email(email, send\_type=0):

vcode = random\_vcode(6)

email\_record = EmailVerifyRecord()

email\_record.vcode\_id = len(models.EmailVerifyRecord.objects.all())+1

email\_record.email = email

email\_record.user\_name = user\_name

email\_record.send\_type = send\_type

email\_record.vcode = vcode

obj = models.EmailVerifyRecord.objects.all()

email\_list = []

for item in obj:

email\_list.append(item.email)

if email in email\_list:

eobj = models.EmailVerifyRecord.objects.get(email=email\_record.email)

eobj.username = user\_name

eobj.vcode = vcode

eobj.send\_type = send\_type

eobj.save()

else:

email\_record.save()

email\_title = ""

email\_body = ""

if send\_type == 0:

email\_title = "注册验证码"

email\_body = "您的注册验证码是{0}，请妥善保管，不要透露给他人。验证码有效期为五分钟。".format(email\_record.vcode)

if send\_type == 1:

email\_title = "忘记密码验证码"

email\_body = "您的验证码是{0}，请妥善保管，不要透露给他人。验证码有效期为五分钟。".format(email\_record.vcode)

if send\_type == 2:

email\_title = "修改邮箱验证码"

email\_body = "您的验证码是{0}，请妥善保管，不要透露给他人。验证码有效期为五分钟。".format(email\_record.vcode)

send\_status = send\_mail(email\_title, email\_body, EMAIL\_FROM, [email])

if send\_status:

msg = "验证码发送成功"

else:

msg = "验证码发送失败"

return msg

try:

msg = send\_register\_email(email, send\_type=0)

return HttpResponse(msg)

except Exception as e:

return HttpResponse(e)

def defindex(request):

meta = request.META

if request.method== "GET" and meta.get("HTTP\_REFERER") == "http://127.0.0.1:8000/evasys/home/":

junior\_index = models.junior\_index.objects.all()

user\_id = request.session.get('user\_id',None)

companylist = models.company.objects.filter(user\_id=user\_id)

res\_list = []

for i in range(len(junior\_index)):

junior\_index\_id = junior\_index[i].junior\_index\_id

memo = junior\_index[i].memo

junior\_index\_name = junior\_index[i].junior\_index\_name

senior\_index\_id = junior\_index[i].senior\_index\_id

evaluator\_id = junior\_index[i].evaluator\_id

period\_id = junior\_index[i].period\_id

senior\_index\_name = models.senior\_index.objects.get(senior\_index\_id=senior\_index\_id).senior\_index\_name

evaluator\_name = models.evaluator.objects.get(evaluator\_id=evaluator\_id).evaluator\_name

period\_name = models.period.objects.get(period\_id=period\_id).period\_name

res\_dic = {"junior\_index\_id":junior\_index\_id,'junior\_index\_name':junior\_index\_name,'senior\_index\_name':senior\_index\_name,'evaluator\_name':evaluator\_name,'period\_name':period\_name,'memo':memo}

res\_list.append(res\_dic)

return render(request,'defindex.html',{'res\_list':res\_list,'companylist':companylist})

elif request.method =="POST":

now = datetime.datetime.now()

now\_str = now.strftime('%Y%m%d%H%M%S')

data = request.POST.get("data")

type = request.POST.get("type")

company\_id = request.POST.get("company\_id")

company\_name=models.company.objects.get(company\_id=company\_id).company\_name

num = str(len(json.loads(data)))

user\_id = request.session['user\_id']

ind\_model\_id = user\_id + now\_str

user\_id = models.users.objects.get(user\_id=user\_id)

ind\_model\_name = request.session['username']

obj = models.ind\_model.objects.filter(user\_id=user\_id)

ind\_model\_cat = str('No.' + str(len(obj)+1))

models.ind\_model.objects.create(ind\_model\_id=ind\_model\_id, ind\_model\_name=ind\_model\_name, user\_id=user\_id,

data=data, ind\_model\_type=type, ind\_model\_num=num,ind\_model\_cat=ind\_model\_cat,

company\_id\_id=company\_id)

return redirect('/evasys/defindex/')

else:

return render(request,'NoPage.html')

def chanindex(request):

meta = request.META

if request.method == "GET" and meta.get("HTTP\_REFERER") == "http://127.0.0.1:8000/evasys/home/":

user\_id = request.session.get('user\_id', None)

companylist = models.company.objects.filter(user\_id=user\_id)

def\_index = models.def\_index.objects.all()

junior\_index = models.junior\_index.objects.all()

res\_list = []

for i in range(len(junior\_index)):

memo = junior\_index[i].memo

junior\_index\_id = junior\_index[i].junior\_index\_id

junior\_index\_name = junior\_index[i].junior\_index\_name

senior\_index\_id = junior\_index[i].senior\_index\_id

evaluator\_id = junior\_index[i].evaluator\_id

period\_id = junior\_index[i].period\_id

senior\_index\_name = models.senior\_index.objects.get(senior\_index\_id=senior\_index\_id).senior\_index\_name

evaluator\_name = models.evaluator.objects.get(evaluator\_id=evaluator\_id).evaluator\_name

period\_name = models.period.objects.get(period\_id=period\_id).period\_name

res\_dic = {"junior\_index\_id":junior\_index\_id,'junior\_index\_name':junior\_index\_name,'senior\_index\_name':senior\_index\_name,'evaluator\_name':evaluator\_name,'period\_name':period\_name,'memo':memo}

res\_list.append(res\_dic)

return render(request,'chanindex.html',{'res\_list':res\_list,'def\_index':def\_index,'companylist':companylist})

elif request.method =="POST":

now = datetime.datetime.now()

now\_str = now.strftime('%Y%m%d%H%M%S')

data = request.POST.get("data")

type = request.POST.get("type")

num = str(len(json.loads(data)))

user\_id = request.session['user\_id']

ind\_model\_id = user\_id + now\_str

user\_id = models.users.objects.get(user\_id = user\_id)

ind\_model\_name = request.session['username']

obj = models.ind\_model.objects.filter(user\_id=user\_id)

ind\_model\_cat = 'No.'+str(len(obj))

models.ind\_model.objects.create(ind\_model\_id = ind\_model\_id,ind\_model\_name = ind\_model\_name,user\_id = user\_id,data = data,ind\_model\_type=type,ind\_model\_num=num,ind\_model\_cat=ind\_model\_cat)

return redirect('/evasys/chanindex/')

else:

return render(request, "NoPage.html")

def creaindex(request):

meta = request.META

if request.method == "GET" and meta.get("HTTP\_REFERER") == "http://127.0.0.1:8000/evasys/home/":

user\_id = request.session.get('user\_id', None)

companylist = models.company.objects.filter(user\_id=user\_id)

def\_index = models.def\_index.objects.all()

junior\_index = models.junior\_index.objects.all()

res\_list = []

for i in range(len(junior\_index)):

memo = junior\_index[i].memo

junior\_index\_id = junior\_index[i].junior\_index\_id

junior\_index\_name = junior\_index[i].junior\_index\_name

senior\_index\_id = junior\_index[i].senior\_index\_id

evaluator\_id = junior\_index[i].evaluator\_id

period\_id = junior\_index[i].period\_id

senior\_index\_name = models.senior\_index.objects.get(senior\_index\_id=senior\_index\_id).senior\_index\_name

evaluator\_name = models.evaluator.objects.get(evaluator\_id=evaluator\_id).evaluator\_name

period\_name = models.period.objects.get(period\_id=period\_id).period\_name

res\_dic = {"junior\_index\_id":junior\_index\_id,'junior\_index\_name':junior\_index\_name,'senior\_index\_name':senior\_index\_name,'evaluator\_name':evaluator\_name,'period\_name':period\_name,'memo':memo}

res\_list.append(res\_dic)

return render(request,'creaindex.html',{'res\_list':res\_list,'def\_index':def\_index,'companylist':companylist})

elif request.method =="POST":

now = datetime.datetime.now()

now\_str = now.strftime('%Y%m%d%H%M%S')

data = request.POST.get("data")

data\_type = request.POST.get("type")

company\_id = request.POST.get("companyid")

company\_name = models.company.objects.get(company\_id=company\_id).company\_name

num = str(len(json.loads(data)))

user\_id = request.session.get('user\_id',None)

ind\_model\_id = user\_id + now\_str

user\_id = models.users.objects.get(user\_id = user\_id)

ind\_model\_name = request.session.get('username',None)

obj = models.ind\_model.objects.filter(user\_id=user\_id)

ind\_model\_cat = str('No.' + str(len(obj) + 1))

models.ind\_model.objects.create(company\_id\_id=company\_id, ind\_model\_id=ind\_model\_id,

ind\_model\_name=ind\_model\_name, user\_id=user\_id, data=data,

ind\_model\_type=data\_type, ind\_model\_num=num, ind\_model\_cat=ind\_model\_cat)

models.def\_index.objects.all().delete()

return render(request, 'creaindex.html')

else:

return render(request, "NoPage.html")

def newindex(request):

ret = {'status': True, 'error': None, 'data': None}

try:

memo = request.POST.get('memo')

evaluator\_id\_id = request.POST.get('evaluator\_id\_id')

print('evaluator\_id\_id',evaluator\_id\_id)

junior\_index\_name = request.POST.get('junior\_index\_name')

period\_id\_id = request.POST.get('period\_id\_id')

print('period\_id\_id',period\_id\_id)

senior\_index\_name = request.POST.get('senior\_index\_name')

exist\_senior = models.senior\_index.objects.filter(senior\_index\_name=senior\_index\_name)

exist\_def\_senior = models.def\_index.objects.filter(senior\_index\_name=senior\_index\_name)

exist\_junior = models.junior\_index.objects.filter(junior\_index\_name=junior\_index\_name)

exist\_def\_junior = models.def\_index.objects.filter(junior\_index\_name=junior\_index\_name)

evaluator\_name = models.evaluator.objects.get(evaluator\_id=evaluator\_id\_id).evaluator\_name

period\_name = models.period.objects.get(period\_id=period\_id\_id).period\_name

if exist\_junior:

ret['status'] = False

ret['error'] = "二级指标已存在"

elif exist\_def\_junior:

ret['status'] = False

ret['error'] = "自定义二级指标重复"

elif senior\_index\_name=="":

ret['status'] = False

ret['error'] = "一级指标不能为空"

elif junior\_index\_name=="":

ret['status'] = False

ret['error'] = "二级指标不能为空"

else:

if exist\_senior:

senior\_index\_id = exist\_senior[0]

print(senior\_index\_id)

elif exist\_def\_senior:

index\_id = exist\_def\_senior[0]

senior\_index\_id = index\_id[2:4]

else:

senior\_index\_id = len(models.def\_index.objects.all())+21

junior\_index\_id =period\_id\_id + evaluator\_id\_id+str(senior\_index\_id) +str(len(models.def\_index.objects.all())+len(models.junior\_index.objects.all()))

junior\_index\_name = junior\_index\_name

res\_dic = {"junior\_index\_id": junior\_index\_id, 'junior\_index\_name': junior\_index\_name,

'senior\_index\_name': senior\_index\_name, 'evaluator\_name': evaluator\_name,

'period\_name': period\_name, 'memo': 'None'}

try:

models.def\_index.objects.create(\*\*res\_dic)

except Exception as e:

print(e)

except Exception as e:

ret['status'] = False

ret['error'] = '请求错误'

return HttpResponse(json.dumps(ret))

def indexval(request):

ind\_model\_name = request.session.get("username", None)

obj = models.ind\_model.objects.filter(ind\_model\_name=ind\_model\_name)

if request.method =="GET":

intervals = models.interval.objects.all()

res\_list = []

datas = []

for i in range(len(obj)):

data = obj[i].data

datas.append(json.loads(data))

return render(request, "indexval.html", {'res\_list': obj, "datas": datas,'intervals':intervals})

elif request.method =="POST":

if request.POST.get("data",None):

datatime = request.POST.get("cycle", None)

datetype = request.POST.get("datetype", None)

interval\_id = request.POST.get("interval\_id", None)

interval = models.interval.objects.get(interval\_id=interval\_id)

now = datetime.datetime.now()

now\_str = now.strftime('%Y%m%d%H%M%S')

user\_id = request.session['user\_id']

val\_model\_id = user\_id + "ValModel" + now\_str

user\_id = models.users.objects.get(user\_id=user\_id)

ind\_model\_id = request.POST.get("id")

ind\_model = models.ind\_model.objects.get(ind\_model\_id=ind\_model\_id)

ind\_model\_cat = ind\_model.ind\_model\_cat

company\_id = ind\_model.company\_id\_id

data = json.loads(request.POST.get("data"))

val\_model\_name = request.session['username']

pre\_data = json.loads(ind\_model.data)

for i,item in enumerate(pre\_data):

if item in data:

pre\_data[item].append(data[item])

try:

models.val\_model.objects.create(company\_id\_id=company\_id,ind\_model\_cat\_id=ind\_model\_cat

,val\_model\_id = val\_model\_id,ind\_model\_id = ind\_model

,val\_model\_name = val\_model\_name,user\_id = user\_id

,data = json.dumps(pre\_data),datetype=datetype

,datatime=datatime,interval\_id=interval)

except Exception as e:

print(e)

return HttpResponse("ok")

elif request.POST.get("ids",None):

ids = request.POST.get("ids")

ids\_json = json.loads(ids)

for item,i in enumerate(ids\_json):

ind\_model\_id = ids\_json[i]

models.ind\_model.objects.get(ind\_model\_id = ind\_model\_id).delete()

return HttpResponse("OK")

def calculate(request):

if request.method =="GET":

val\_model\_name = request.session.get("username", None)

algorithms = models.Algorithm\_description.objects.all()

obj = models.val\_model.objects.filter(val\_model\_name = val\_model\_name)

ii = models.val\_model.objects.filter(val\_model\_name = val\_model\_name).values\_list('val\_model\_id','data')

datas = []

for i in range(len(obj)):

data = obj[i].data

datas.append(json.loads(data))

return render(request, "calculate.html", {'res\_list': obj, "datas": datas,'algorithms':algorithms})

elif request.method =="POST":

if request.POST.get("ids",None):

ids = request.POST.get("ids")

ids\_json = json.loads(ids)

for item,i in enumerate(ids\_json):

val\_model\_id = ids\_json[i]

models.val\_model.objects.get(val\_model\_id = val\_model\_id).delete()

return redirect("/evasys/resvisualization/")

elif request.POST.get("cal\_ids",None):

user\_id = request.session.get("user\_id",None)

userInstance = models.users.objects.get(user\_id=user\_id)

val\_model\_id = json.loads(request.POST.get("cal\_ids"))

algorithm\_id = request.POST.get("alg\_id")

algorithmInstance = models.Algorithm\_description.objects.get(algorithm\_id=algorithm\_id)

now = datetime.datetime.now()

now\_str = now.strftime('%Y%m%d%H%M%S')

val\_res\_id = user\_id+"ValRes"+now\_str

if algorithm\_id=='1':

for i in range(len(val\_model\_id)):

valModelInstance = models.val\_model.objects.get(val\_model\_id=val\_model\_id[i])

companyId = valModelInstance.company\_id

ind\_model\_cat\_id = valModelInstance.ind\_model\_cat\_id

interval\_id = valModelInstance.interval\_id

intervalObj = models.interval.objects.get(interval\_id=interval\_id)

intervalStr = intervalObj.interval\_data

intervalList = list(map(int, intervalStr.split(",")))

evaDF, seniorResDF, evaluatorResDF, periodResDF, totalScoreDict= ExtAssMethod(dataJson=valModelInstance.data,

jingDianYuList=intervalList).addresDict()

print(periodResDF)

try:

periodResJson = json.dumps(periodResDF.to\_dict(), cls=MyEncoder)

evaluatorResJson =json.dumps(evaluatorResDF.to\_dict(), cls=MyEncoder)

seniorResJson = json.dumps(seniorResDF.to\_dict(), cls=MyEncoder)

except Exception as e:

print(e, "calculate wrong")

periodResJson, evaluatorResJson, seniorResJson, totalScoreDict = None, None,None, None

if periodResJson is not None:

try:

models.val\_res.objects.create(final\_res=totalScoreDict["t"], period\_res=periodResJson,

evaluator\_res=evaluatorResJson,senior\_res=seniorResJson,

user\_id=userInstance, val\_model\_id=valModelInstance,

algorithm\_id=algorithmInstance,company\_id=companyId,

val\_res\_id=val\_res\_id,ind\_model\_cat\_id=ind\_model\_cat\_id)

except Exception as e:

print(e, "model wrong")

return HttpResponse("Bad Request")

else:

print("calculate Wrong")

return HttpResponse("Bad Request")

return HttpResponse("OK")

else:

value\_list = []

for i in range(len(val\_model\_id)):

obj = models.val\_model.objects.get(val\_model\_id=val\_model\_id[i])

data\_dic = json.loads(obj.data)

v\_list=[]

global k\_list

k\_list = list(data\_dic.keys())

for v in data\_dic.values():

v\_list.append(float(v[4]))

value\_list.append(v\_list)

df = pd.DataFrame(value\_list, columns=k\_list).T

data\_values=df.values

if algorithm\_id == '2':

cv\_w = CV(data\_values)

print('变异系数法权重：',cv\_w)

cv\_v = CaculateValue(data\_values, cv\_w)

print('变异系数法结果:',cv\_v)

res = {'cv\_w':list(cv\_w),'cv\_v':cv\_v}

if algorithm\_id == '3':

en\_w = EntropyMethod(data\_values)

print('熵值法权重：',en\_w)

en\_v = CaculateValue(data\_values, en\_w)

print('熵值法结果:',en\_v)

res = {'w':list(en\_w),'v':en\_v}

if algorithm\_id == '6':

cr\_w = Critic(data\_values)

print('Critic法权重：',cr\_w)

cr\_v = CaculateValue(data\_values, cr\_w)

print('Critic法结果:',cr\_v)

res = {'w':list(cr\_w),'v':cr\_v}

if algorithm\_id == '6':

me\_w = MeanError(df.values)

print('标准离差法权重：',me\_w)

me\_v = CaculateValue(df.values, me\_w)

print('标准离差法结果:',me\_v)

res = {'w':list(me\_w),'v':me\_v}

now = datetime.datetime.now()

now\_str = now.strftime('%Y%m%d%H%M%S')

val\_res\_id = user\_id + now\_str

result = json.dumps(res)

models.val\_res.objects.create(junior\_res=result,algorithm\_id\_id=algorithm\_id,val\_model\_id=val\_model\_id,val\_res\_id=val\_res\_id,user\_id\_id=user\_id,ind\_model\_cat\_id=ind\_model\_cat\_id)

return HttpResponse('ok')

def resvisualization(request):

if request.method == "GET":

if request.GET.get("id",None):

val\_res\_id = request.GET.get("id",None)

valResInstance = models.val\_res.objects.get(val\_res\_id=val\_res\_id)

companyName = models.company.objects.get(company\_id=valResInstance.company\_id).company\_name

algorithmName = models.Algorithm\_description.objects.get(algorithm\_id=valResInstance.algorithm\_id).algorithm\_name

totalRes, periodRes, evaluatorRes, seniorRes = valResInstance.final\_res, valResInstance.period\_res, valResInstance.evaluator\_res, valResInstance.senior\_res

periodResDict, evaluatorResDict, seniorResDict = json.loads(periodRes),json.loads(evaluatorRes),json.loads(seniorRes)

valModelId = valResInstance.val\_model\_id

evaData = models.val\_model.objects.get(val\_model\_id=valModelId).data

evaDataJson = json.loads(evaData)

jingDianYuId = models.val\_model.objects.get(val\_model\_id=valModelId).interval\_id

jingDianYu = models.interval.objects.get(interval\_id=jingDianYuId).interval\_data

jingDianYuList = list(map(int, jingDianYu.split(",")))

jingDianYuTuple = tuple((jingDianYuList[i], jingDianYuList[i + 1]) for i in range(len(jingDianYuList) - 1))

mapJingDianYuList = [1 if float(totalRes)<=x[1] and float(totalRes)>x[0] else 0 for x in jingDianYuTuple]

mapScoreDict = {0:"很差", 1:"较差", 2:"一般", 3:"较好", 4:"很好"}

index = np.argmax(mapJingDianYuList)

totalEvaluation = mapScoreDict[index]

return render(request, "resvisualization.html",

{"totalRes": totalRes, "totalEvaluation": totalEvaluation,

"periodResDict": periodResDict,

"evaluatorResDict": evaluatorResDict, "seniorResDict":seniorResDict,

"companyName":companyName, "algorithmName":algorithmName, "val\_res\_id":val\_res\_id})

else:

return HttpResponse("BAD REQUEST")

elif request.method == "POST":

name = request.POST.get("senior\_name",None)

val\_res\_id = request.POST.get("val\_res\_id",None)

obj = models.val\_res.objects.get(val\_res\_id=val\_res\_id)

senior\_index = pd.read\_json(obj.senior\_res)

senior\_index = senior\_index.sort\_values(by='senior\_index\_id')

junior\_res = pd.read\_json(obj.junior\_res)

junior\_res = junior\_res.T

junior\_res['senior\_index\_id'] = junior\_res['senior\_index\_id'].astype('int64')

def senior\_plot(senior\_name, senior\_index, junior\_res):

if senior\_name not in senior\_index['senior\_index\_name'].values:

print('非法指标')

return 0

index = senior\_index[senior\_index['senior\_index\_name'] == senior\_name].index

title = senior\_name + ':' + str(

round(senior\_index.loc[index, 'value'].values[0], 2))

senior\_id=senior\_index.loc[index, 'senior\_index\_id'].values

df = junior\_res[junior\_res['senior\_index\_id'] == senior\_id[0]]

plt.figure(figsize=(12, 9))

labels = df[['junior\_index\_name', 'value']].values

sizes = [round(x, 2) for x in df['value'].values]

labels1 = []

labels2 = df['junior\_index\_name']

for i in labels:

labels1.append(i[0] + ':' + str(round(i[1], 2)))

explode = None

plt.pie(sizes, explode=explode, labels=labels1,

labeldistance=1.1,

shadow=False, startangle=90, pctdistance=0.8, textprops={'fontsize': 15})

plt.title(title, fontsize=20)

plt.axis('equal')

plt.legend(labels2, fontsize=10,ncol=1,loc='upper right',bbox\_to\_anchor=(1.1,1.05),borderaxespad=0.3)

sio = BytesIO()

plt.savefig(sio, format='png')

data = base64.encodebytes(sio.getvalue()).decode()

return data

data = senior\_plot(name, senior\_index, junior\_res)

return HttpResponse(data)

def resshow(request):

if request.method=="GET":

user\_id = request.session.get("user\_id",None)

username = request.session.get("username",None)

obj = models.val\_res.objects.filter(user\_id=user\_id)

res\_list = []

for i in range(len(obj)):

val\_res\_id = obj[i].val\_res\_id

val\_model = obj[i].val\_model\_id

final\_res = obj[i].final\_res

ctime = obj[i].ctime

res\_dict = {"val\_res\_id":val\_res\_id,"val\_model\_id":val\_model,"final\_res":final\_res,"ctime":ctime,"username":username}

res\_list.append(res\_dict)

return render(request,"resshow.html",{"res\_list":obj})

elif request.method =="POST":

if request.POST.get("ids",None):

ids = request.POST.get("ids")

ids\_json = json.loads(ids)

for item,i in enumerate(ids\_json):

val\_res\_id = ids\_json[i]

models.val\_res.objects.get(val\_res\_id = val\_res\_id).delete()

return redirect("/evasys/resvisualization/")

def modscope(request):

try:

inf = request.META['HTTP\_REFERER']

except:

return HttpResponse("BAD REQUEST")

if inf == 'http://127.0.0.1:8000/evasys/home/':

return render(request,'modscope.html')

def modalgorithm(request):

return render(request,'modalgorithm.html')

def moddescribe(request):

return render(request,'moddescribe.html')

def modevaluation(request):

return render(request,'modevaluation.html')

def EXEMtheory(request):

return render(request,'EXEMtheory.html')

def EXEMfeature(request):

return render(request,'EXEMfeature.html')

def EXEMmodel(request):

return render(request,'EXEMmodel.html')

def EWMtheory(request):

return render(request,'EWMtheory.html')

def EWMfeature(request):

return render(request,'EWMfeature.html')

def EWMmodel(request):

return render(request,'EWMmodel.html')

def AHPtheory(request):

return render(request,'AHPtheory.html')

def AHPfeature(request):

return render(request,'AHPfeature.html')

def AHPmodel(request):

return render(request,'AHPmodel.html')

def CMtheory(request):

return render(request,'CMtheory.html')

def CMfeature(request):

return render(request,'CMfeature.html')

def CMmodel(request):

return render(request,'CMmodel.html')

def getexcel(request):

user\_id = request.session.get('user\_id')

companylist = models.company.objects.filter(user\_id\_id=user\_id)

companyid=request.POST.get("companyid",None)

intervals = models.interval.objects.all()

if request.method == "POST" and companyid is None:

form = UploadExcelForm(request.POST, request.FILES)

res\_list = []

if form.is\_valid():

wb = xlrd.open\_workbook(filename=None, file\_contents=request.FILES['excel'].read())

table = wb.sheets()[0]

row = table.nrows

val\_list=[]

index\_list=[]

for i in range(1,row):

col = table.row\_values(i)

period\_name = col[0]

evaluator\_name = col[1]

senior\_index\_name = col[2]

junior\_index\_name = col[3]

val = col[4]

res\_dic = {'period\_name':period\_name,'evaluator\_name':evaluator\_name,'senior\_index\_name':senior\_index\_name,'junior\_index\_name':junior\_index\_name,'val':val}

res\_list.append(res\_dic)

val\_list.append(val)

index\_list.append(col[:4])

ind\_model\_obj = models.ind\_model.objects.filter(user\_id=user\_id)

def get\_index\_id(obj, index\_list):

for i in range(len(obj)):

data = json.loads(obj[i].data)

if list(data.values()) == index\_list:

ind\_model\_id = obj[i].ind\_model\_id

pre\_data = json.loads(obj[i].data)

ind\_model = obj[i]

return ind\_model\_id, pre\_data, ind\_model

ind\_model\_id, pre\_data, ind\_model = get\_index\_id(ind\_model\_obj, index\_list)

ind\_model = models.ind\_model.objects.get(ind\_model\_id=ind\_model\_id)

pre\_data = json.loads(ind\_model.data)

for i, item in enumerate(pre\_data):

res\_list[i]['junior\_index\_id']=item

return render(request, 'getexcel.html', {'res\_list': res\_list,'ind\_model\_id':ind\_model\_id

, 'companylist': companylist,"intervals":intervals})

if companyid:

datatime = request.POST.get("datatime",None)

datetype = request.POST.get("datetype",None)

ind\_model\_id = request.POST.get("ind\_model\_id",None)

interval\_id = request.POST.get("interval\_id",None)

data = request.POST.get("data",None)

val\_model\_name = request.session['username']

ind\_model = models.ind\_model.objects.get(ind\_model\_id=ind\_model\_id)

now = datetime.datetime.now()

now\_str = now.strftime('%Y%m%d%H%M%S')

val\_model\_id = user\_id + now\_str

models.val\_model.objects.create(val\_model\_id=val\_model\_id, ind\_model\_id\_id=ind\_model\_id,

val\_model\_name=val\_model\_name, user\_id\_id=user\_id,

data=data, ind\_model\_cat\_id=ind\_model.ind\_model\_cat

,datatime=datatime,datetype=datetype,interval\_id\_id=interval\_id

,company\_id\_id=companyid)

return render(request,'getexcel.html',{'companylist':companylist,"intervals":intervals})

elif request.method == "GET":

return render(request,'getexcel.html',{'companylist':companylist,'intervals':intervals})

def sendecode\_email\_change(request):

user\_id = request.session['user\_id']

user\_name=models.users.objects.get(user\_id=user\_id).user\_name

email = request.POST.get("email")

users = models.users.objects.all()

for item in users:

if email == item.email:

msg = '邮箱已被注册'

return HttpResponse(msg)

def random\_vcode(randomlength=6):

vcode = ''

chars = 'AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz0123456789'

length = len(chars) - 1

random = Random()

for i in range(randomlength):

vcode += chars[random.randint(0, length)]

return vcode

def send\_register\_email(email, send\_type=0):

vcode = random\_vcode(6)

email\_record = EmailVerifyRecord()

email\_record.vcode\_id = len(models.EmailVerifyRecord.objects.all()) + 1

email\_record.email = email

email\_record.user\_name = user\_name

email\_record.send\_type = send\_type

email\_record.vcode = vcode

obj = models.EmailVerifyRecord.objects.all()

email\_list = []

for item in obj:

email\_list.append(item.email)

if email in email\_list:

eobj = models.EmailVerifyRecord.objects.get(email=email\_record.email)

eobj.username = user\_name

eobj.vcode = vcode

eobj.send\_type = send\_type

eobj.save()

else:

email\_record.save()

email\_title = ""

email\_body = ""

if send\_type == 0:

email\_title = "注册验证码"

email\_body = "您的注册验证码是{0}，请妥善保管，不要透露给他人。验证码有效期为五分钟。".format(email\_record.vcode)

if send\_type == 1:

email\_title = "忘记密码验证码"

email\_body = "您的验证码是{0}，请妥善保管，不要透露给他人。验证码有效期为五分钟。".format(email\_record.vcode)

if send\_type == 2:

email\_title = "修改邮箱验证码"

email\_body = "您的验证码是{0}，请妥善保管，不要透露给他人。验证码有效期为五分钟。".format(email\_record.vcode)

send\_status = send\_mail(email\_title, email\_body, EMAIL\_FROM, [email])

if send\_status:

msg = "验证码发送成功"

else:

msg = "验证码发送失败"

return msg

try:

msg = send\_register\_email(email, send\_type=0)

return HttpResponse(msg)

except Exception as e:

print(e)

return HttpResponse(e)

def Usercenter(request):

user\_id = request.session['user\_id']

users = models.users.objects.get(user\_id=user\_id)

if request.method=="GET":

user\_name=users.user\_name

email=users.email

creat\_time=users.createtime

uptime=users.updatetime

return render(request,'Usercenter.html',{'user\_name':user\_name,'email':email,'creat\_time':creat\_time,'uptime':uptime})

def company\_info(request):

industry = models.industry.objects.all()

user\_id = request.session['user\_id']

company = models.company.objects.filter(user\_id=user\_id)

if request.method=="GET":

return render(request,'company\_info.html',{'industry':industry,'companys':company,})

if request.method=="POST":

count = len(models.company.objects.all())

companyname = request.POST.get('companyname', None)

liscom\_code = request.POST.get('liscom\_code', None)

industry\_id = request.POST.get('industry', None)

telephone = request.POST.get('telephone', None)

company\_address = request.POST.get('company\_address', None)

introduction = request.POST.get('introduction', None)

website = request.POST.get('website', None)

company\_del\_id = request.POST.get('company\_del\_id',None)

if industry\_id:

for item in company:

if item.company\_name == companyname:

error\_msg = '该企业已经关联，请重新输入企业名称'

return HttpResponse(error\_msg)

times = time.strftime('%Y%m%d%H%M%S', time.localtime())

company\_id = times + str(count)

industry = models.industry.objects.get(industry\_id=industry\_id)

user = models.users.objects.get(user\_id=user\_id)

models.company.objects.create(company\_id=company\_id, company\_name=companyname, liscom\_code=liscom\_code,

industry\_id=industry, telephone=telephone,

company\_address=company\_address,

introduction=introduction, website=website, user\_id=user)

else:

models.company.objects.filter(company\_id=company\_del\_id).delete()

company = models.company.objects.filter(user\_id=user\_id)

return redirect('/evasys/company\_info/')

def company\_del(request):

if request.method=="POST":

company\_del\_id=request.POST.get('company\_del\_id')

models.company.objects.filter(company\_id=company\_del\_id).delete()

return HttpResponse('删除成功')

def email\_info(request):

user\_id = request.session['user\_id']

users = models.users.objects.get(user\_id=user\_id)

email = users.email

if request.method=="GET":

email = users.email

return render(request,'email\_info.html',{'email':email})

if request.method=="POST":

ecode=request.POST.get('Verification\_Code')

email = request.POST.get('new\_email', None)

obj = models.EmailVerifyRecord.objects.get(email=email)

vcode = obj.vcode

send\_time = obj.send\_time.replace(tzinfo=None)

nowTime = datetime.datetime.now()

interval\_time = nowTime - send\_time

if ecode == vcode and interval\_time.seconds < 300:

users.email=email

users.save()

return HttpResponse("修改成功")

elif vcode!=ecode:

return HttpResponse('验证码错误')

else:

return HttpResponse("验证码失效")

def code\_info(request):

user\_id = request.session['user\_id']

users=models.users.objects.get(user\_id=user\_id)

code = users.password

if request.method=="POST":

original\_code=request.POST.get('original\_code',None)

new\_code=request.POST.get('repassword',None)

if code==original\_code:

users.password=new\_code

users.save()

err\_msg = "修改成功"

return render(request, 'code\_info.html', {"success\_msg": err\_msg})

else:

err\_msg="原来密码输入错误"

return render(request,'code\_info.html',{"success\_msg":err\_msg})

if request.method=="GET":

return render(request,'code\_info.html')

def user\_feedback(request):

if request.method=="GET":

return render(request,'user\_feedback.html')

if request.method=="POST":

user\_id = request.session['user\_id']

times = time.strftime('%Y%m%d%H%M%S', time.localtime())

feedback\_id=user\_id+times

evaluation\_content=request.POST.get('evaluation',None)

models.user\_feedback.objects.create(feedback\_id=feedback\_id,evaluation\_content=evaluation\_content,user\_id\_id=user\_id)

return HttpResponse('提交成功')

def historical\_eval(request):

if request.method=="GET":

user\_id = request.session['user\_id']

historical\_eval=models.user\_feedback.objects.filter(user\_id=user\_id)

return render(request, 'historical\_eval.html',{'historical\_eval':historical\_eval})

if request.method=="POST":

feedback\_id=request.POST.get("evaluation",None)

models.user\_feedback.objects.get(feedback\_id=feedback\_id).delete()

return HttpResponse('删除成功')

# apps.py

from \_\_future\_\_ import unicode\_literals

from django.apps import AppConfig

class EvasysappConfig(AppConfig):

name = 'evasysapp'

# forms.py

from django import forms

from django.utils.translation import gettext as \_

from django.core.exceptions import ValidationError

def validate\_excel(value):

if value.name.split('.')[-1] not in ['xls','xlsx']:

raise ValidationError(\_('Invalid File Type: %(value)s'),params={'value': value},)

class UploadExcelForm(forms.Form):

excel = forms.FileField(validators=[validate\_excel])

# utils.py

import pandas as pd

import numpy as np

from sympy import \*

from sympy.abc import x

import json

class ExtAssMethod:

def \_\_init\_\_(self, dataJson, jingDianYuList):

self.dataJson = dataJson

self.jingDianYuList = jingDianYuList

def extendJingDianYu(self, jingDianYuList, direction = "left"):

if direction=="left":

negativePoint = jingDianYuList[0]-(jingDianYuList[2]-jingDianYuList[1])

jingDianYuList.insert(0, negativePoint)

elif direction=="right":

advancePoint = jingDianYuList[-1]+(jingDianYuList[-2]-jingDianYuList[-3])

jingDianYuList.append(advancePoint)

else:

pass

return jingDianYuList

def getJieYuTuple(self, jingDianYuList):

return (min(jingDianYuList), max(jingDianYuList))

def testData(self):

if isinstance(self.jingDianYuList, list):

flag = 0

else:

flag = 1

if isinstance(self.dataJson, dict):

flag = 0

else:

flag = 1

return flag

def addressIntervalData(self,jingDianYuList):

return (tuple(((jingDianYuList[i] + jingDianYuList[i+1])/2,

(jingDianYuList[i+1] - jingDianYuList[i])/2) for i in range(len(jingDianYuList)-1)),

tuple((jingDianYuList[i], jingDianYuList[i+1]) for i in range(len(jingDianYuList)-1)))

def extDistance(self, singleJingDianYuTuple, singleEvaDataNum):

return abs(singleEvaDataNum-((singleJingDianYuTuple[0]+singleJingDianYuTuple[1])\*0.99)/2)\

-(singleJingDianYuTuple[1]-singleJingDianYuTuple[0])/2

def extWeight(self, jingDianYuTuple, evaDataList):

absoluteWeightTuple = tuple(tuple((2\*(x-y[0])+0.1)/(y[1]-y[0]) if x<(y[0]+y[1])/2 else (2\*(y[1]-x)+0.1)/(y[1]-y[0]) for y in jingDianYuTuple) for x in evaDataList)

maxAbsoluteWeightNP = np.max(np.array(absoluteWeightTuple), axis = 1)

relativeWeightNP = maxAbsoluteWeightNP/sum(maxAbsoluteWeightNP)

return relativeWeightNP

def correlation(self, jingDianYuTuple, jieYuTuple, evaDataList):

return tuple(tuple(self.extDistance(singleJingDianYu, singleEvaNum)/(self.extDistance(jieYuTuple,singleEvaNum)-self.extDistance(singleJingDianYu,singleEvaNum)+singleJingDianYu[0]-singleJingDianYu[1]) \

if singleEvaNum<= singleJingDianYu[1] and singleEvaNum>singleJingDianYu[0] else \

self.extDistance(singleJingDianYu, singleEvaNum)/(self.extDistance(jieYuTuple,singleEvaNum)-self.extDistance(singleJingDianYu,singleEvaNum)) \

for singleJingDianYu in jingDianYuTuple) for singleEvaNum in evaDataList)

def totalCorrelation(self, correlationNP, relativeWeightNP):

"""

:param correlationNP: (number of evaluation data, number of jingdianyu)

:param relativeWeightNP: (1,number of evaluation data)

:return: totalCorelationNP,()

"""

return np.dot(relativeWeightNP, correlationNP)

def \_checkData(self,minLimit,maxLimit,data):

if data is not None:

if minLimit<=data and data<=maxLimit:

return True

elif data<minLimit:

print("数据值小于最低下限", data, minLimit)

return False

else :

print("数据值大于最大上限", data, maxLimit)

return False

else:

return False

def checkData(self, x1, x2, x3, maxCorrelaiton, \*limits):

if maxCorrelaiton>0:

if len(limits)==3:

minLimit, middleLimit, maxLimit = limits

if not self.\_checkData(minLimit, middleLimit, x1):

x1 = None

else:

pass

if not self.\_checkData(middleLimit, maxLimit, x2):

x2 = None

else:

pass

elif len(limits)==4:

minLimit, middle1Limit, middle2Limit, maxLimit = limits

if not self.\_checkData(minLimit, middle1Limit, x1):

x1 = None

else:

pass

if not self.\_checkData(middle1Limit, middle2Limit, x2):

x2 = None

else:

pass

if not self.\_checkData(middle2Limit, maxLimit, x3):

x3 = None

else:

pass

else:

pass

else:

pass

return x1,x2,x3

def checkCommonEndPoint(self, tuple1, tuple2):

if tuple1[0]==tuple2[0] :

return 0

else:

if tuple1[1]==tuple2[1]:

return 1

else:

return False

def subExcuteTotalCorrelation(self, jingDianYuList, evaDataList):

computeJingDianYuTuple, jingDianYuTuple = self.addressIntervalData(jingDianYuList)

jieYuTuple = self.getJieYuTuple(jingDianYuList)

relativeWeightTuple = self.extWeight(jingDianYuTuple,evaDataList)

correlationTuple = self.correlation(jingDianYuTuple, jieYuTuple, evaDataList)

correlationMat = np.matrix(correlationTuple)

relativeWeightMat = np.matrix(relativeWeightTuple)

totalCorrelationMat = self.totalCorrelation(correlationMat, relativeWeightMat)

totalCorrelationNP = np.asarray(totalCorrelationMat).reshape(totalCorrelationMat.shape[1])

return totalCorrelationNP, jingDianYuTuple

def totalScore(self, totalCorrelationNp, jingDianYuTuple, jieYuTuple, evaDataList):

jingDianYuIndexNP = int(np.argmax(totalCorrelationNp))

belongJinDianYuTuple = jingDianYuTuple[jingDianYuIndexNP]

maxCorrelationNum = np.max(totalCorrelationNp)

aStar,bStar = belongJinDianYuTuple[0],belongJinDianYuTuple[1]

a,b = jieYuTuple[0], jieYuTuple[1]

if bStar<=(a+b)/2:

x1 = solve(((aStar+bStar)/2-x-(bStar-aStar)/2)/(

(a+b)/2-x-(b-a)/2-((bStar+aStar)/2-x)-(bStar-aStar)/2))

x2 = solve((x-(aStar+bStar)/2-(bStar-aStar)/2)/(

(a+b)/2-x-(b-a)/2-(x-(bStar+aStar)/2)-(bStar-aStar)/2))

x3 = None

x1Num, x2Num, x3Num = self.checkData(x1[0],x2[0], x3, maxCorrelationNum, aStar, (aStar+bStar)/2, bStar)

elif aStar>=(a+b)/2:

x1 = solve(((aStar + bStar) / 2 - x - (bStar - aStar) / 2) / (

x-(a + b) / 2 - (b - a) / 2 - ((bStar + aStar) / 2 - x) - (

bStar - aStar) / 2)-maxCorrelationNum)

x2 = solve((x-(aStar + bStar) / 2 - (bStar - aStar) / 2) / (

x-(a + b) / 2 - (b - a) / 2 - (x - (bStar + aStar) / 2) - (

bStar - aStar) / 2)-maxCorrelationNum)

x3 = None

x1Num, x2Num, x3Num = self.checkData(x1[0], x2[0], x3, maxCorrelationNum, aStar, (aStar + bStar) / 2, bStar)

else:

if (aStar+bStar)<=(a+b):

x1 = solve(((aStar + bStar) / 2 - x - (bStar - aStar) / 2) / (

(a + b) / 2 - x - (b - a) / 2 - ((bStar + aStar) / 2 - x) - (

bStar - aStar) / 2) - maxCorrelationNum \* (bStar - aStar) / (

2 \* (bStar - a)))

x2 = solve((x - (aStar + bStar) / 2 - (bStar - aStar) / 2) / (

(a + b) / 2 - x - (b - a) / 2 - (x - (bStar + aStar) / 2) - (

bStar - aStar) / 2) - maxCorrelationNum \* (bStar - aStar) / (

2 \* (bStar - a)))

x3 = solve((x -(aStar + bStar) / 2 - (bStar - aStar) / 2) / (

x - (a + b) / 2 - (b - a) / 2 - (x - (bStar + aStar) / 2) - (

bStar - aStar) / 2) - maxCorrelationNum \* (bStar - aStar) / (

2 \* (bStar - a)))

x1Num, x2Num, x3Num = self.checkData(x1[0], x2[0], x3[0], maxCorrelationNum, aStar, (aStar + bStar) / 2,(a+b)/2, bStar)

else:

x1 = solve(((aStar + bStar) / 2 - x - (bStar - aStar) / 2) / (

(a + b) / 2 - x - (b - a) / 2 - ((bStar + aStar) / 2 - x) - (

bStar - aStar) / 2) - maxCorrelationNum \* (bStar - aStar) / (

2 \* (bStar - a)))

x2 = solve(((aStar + bStar) / 2-x - (bStar - aStar) / 2) / (

x-(a + b) / 2 - (b - a) / 2 - ((bStar + aStar) / 2-x) - (

bStar - aStar) / 2) - maxCorrelationNum \* (bStar - aStar) / (

2 \* (bStar - a)))

x3 = solve((x -( aStar + bStar) / 2 - (bStar - aStar) / 2) / (

x - (a + b) / 2 - (b - a) / 2 - (x - (bStar + aStar) / 2) - (

bStar - aStar) / 2) - maxCorrelationNum \* (bStar - aStar) / (

2 \* (bStar - a)))

x1Num, x2Num, x3Num = self.checkData(x1[0], x2[0], x3[0], maxCorrelationNum, aStar, (a + b) / 2, (aStar + bStar) / 2, bStar)

xList = []

if x1Num is not None:

xList.append(x1Num)

else:

pass

if x2Num is not None:

xList.append(x2Num)

else:

pass

if x3Num is not None:

xList.append(x3Num)

else:

pass

if len(xList)==2:

if jingDianYuIndexNP == 0:

extendJingDianYuList = self.extendJingDianYu(self.jingDianYuList)

checkTotalCorrelationNP, extendJingDianYuTuple = self.subExcuteTotalCorrelation(

jingDianYuList=extendJingDianYuList, evaDataList = evaDataList)

if checkTotalCorrelationNP[0]<=checkTotalCorrelationNP[2]:

resScore = xList[0]

else:

resScore = xList[1]

elif jingDianYuIndexNP==len(jingDianYuTuple)-1:

extendJingDianYuList = self.extendJingDianYu(self.jingDianYuList, direction="right")

checkTotalCorrelationNP, extendJingDianYuTuple = self.subExcuteTotalCorrelation(

jingDianYuList=extendJingDianYuList, evaDataList = evaDataList)

if checkTotalCorrelationNP[-1]<=checkTotalCorrelationNP[-3]:

resScore = xList[0]

else:

resScore = xList[1]

else:

if totalCorrelationNp[jingDianYuIndexNP-1]<=totalCorrelationNp[jingDianYuIndexNP+1]:

resScore = xList[1]

else:

resScore = xList[0]

elif len(xList)==1:

resScore = xList[0]

else:

print("计算失败")

resScore = False

return resScore

def oneLayerExcute(self, evaDataList):

totalCorrelationNP, jingDianYuTuple = self.subExcuteTotalCorrelation(jingDianYuList=self.jingDianYuList, evaDataList=evaDataList)

jieYuTuple = self.getJieYuTuple(self.jingDianYuList)

totalScore = self.totalScore(totalCorrelationNP,jingDianYuTuple, jieYuTuple, evaDataList)

return totalScore

def splitEvaData(self, dataDict, removeStringNum = 2):

newDict = {}

for key, valueList in dataDict.items():

removeKey = key[0:-removeStringNum]

if removeKey in newDict:

if isinstance(valueList, list):

newDict[removeKey].append(float(valueList[-1]))

else:

newDict[removeKey].append(float(valueList))

else:

if isinstance(valueList, list):

newDict[removeKey] = [float(valueList[-1])]

else:

newDict[removeKey] = [float(valueList)]

return newDict

def oneLayerMultiExcute(self, newDict):

scoreTuple = tuple(tuple((key, self.oneLayerExcute(evaDataList=valueList))) \

if len(valueList)>1 else tuple((key,valueList[0])) \

for key,valueList in newDict.items())

scoreDict = dict(scoreTuple)

return scoreDict

def addresDict(self):

dataJson = json.loads(self.dataJson)

valuesLi = []

for key, valueList in dataJson.items():

valueList.append(key)

valuesLi.append(valueList)

evaDF = pd.DataFrame(valuesLi, columns = ["评价阶段名称", "评价者名称", "一级评价指标名称", "二级评价指标名称", "二级评价指标量值", "二级评价指标编码"])

evaDF["一级评价指标编码"] = evaDF["二级评价指标编码"].map(lambda x:x[:-2])

evaDF["评价者编码"] = evaDF["一级评价指标编码"].map(lambda x:x[:-2])

evaDF["评价阶段编码"] = evaDF["评价者编码"].map(lambda x:x[:-1])

seniorResDF = evaDF.drop(columns=["二级评价指标名称", "二级评价指标编码", "二级评价指标量值"])

seniorResDF.drop\_duplicates(subset=["一级评价指标编码"], inplace=True)

seniorScoreDict, evaluatorScoreDict, peroidScoreDict, totalScoreDict = self.excute()

seniorResDF["一级评价指标结果"] = seniorResDF["一级评价指标编码"].map(seniorScoreDict)

evaluatorResDF = seniorResDF.drop(columns=["一级评价指标编码","一级评价指标结果", "一级评价指标名称"])

evaluatorResDF.drop\_duplicates(subset=["评价者编码"],inplace=True)

evaluatorResDF["评价者结果"] = evaluatorResDF["评价者编码"].map(evaluatorScoreDict)

periodResDF = evaluatorResDF.drop(columns=["评价者编码", "评价者结果", "评价者名称"])

periodResDF.drop\_duplicates(subset=["评价阶段编码"], inplace=True)

periodResDF["评价阶段结果"] = periodResDF["评价阶段编码"].map(peroidScoreDict)

return evaDF, seniorResDF,evaluatorResDF,periodResDF,totalScoreDict

def excute(self):

dataJson = json.loads(self.dataJson)

juniorIndexDict = self.splitEvaData(dataJson)

seniorScoreDict = self.oneLayerMultiExcute(juniorIndexDict)

seniorIndexDict = self.splitEvaData(seniorScoreDict)

evaluatorScoreDict = self.oneLayerMultiExcute(seniorIndexDict)

evaluatorIndexDict = self.splitEvaData(evaluatorScoreDict,removeStringNum=1)

peroidScoreDict = self.oneLayerMultiExcute(evaluatorIndexDict)

totalData = {"t":list(peroidScoreDict.values())}

totalScoreDict = self.oneLayerMultiExcute(totalData)

return seniorScoreDict, evaluatorScoreDict,peroidScoreDict,totalScoreDict

# models.py

# -\*- coding: utf-8 -\*-

from \_\_future\_\_ import unicode\_literals

from django.db import models

class industry(models.Model):

industry\_id\_choice = (

(1,'农林牧渔业'),

(2,'采矿业'),

(3,'制造业'),

(4,'住宿和餐饮业'),

(5,'金融业'),

(6, '房地产业'),

(7, '租赁和商务服务'),

(8, '电力、燃气及水的生产和供应'),

(9, '建筑业'),

(10, '水利、环境和公共设施管理'),

(11, '居民服务和其他服务'),

(12, '教育'),

(13, '信息传输、计算机服务和软件业'),

(14, '卫生、社会保障和社会福利'),

(15, '文化、体育和娱乐业'),

(16, '公共管理和社会组织'),

(17, '国际组织'),

(18, '科学研究、技术服务和地质勘查'),

(19, '交通运输、仓储和邮政'),

(20, '批发和零售'),

)

industry\_id = models.IntegerField(choices=industry\_id\_choice,primary\_key= True)

industry\_name = models.CharField(max\_length=50)

memo = models.CharField(max\_length=100,null=True,blank=True)

def \_\_str\_\_(self):

return self.industry\_name

class company(models.Model):

company\_id = models.CharField(max\_length=18,primary\_key=True)

company\_name = models.CharField(max\_length=40,unique=True)

liscom\_code = models.CharField(max\_length=6,default=0)

company\_address = models.CharField(max\_length=100)

industry\_id = models.ForeignKey("industry",on\_delete = models.CASCADE,to\_field='industry\_id',null = True)

user\_id = models.ForeignKey("users",to\_field='user\_id',on\_delete=models.CASCADE,null=True)

telephone = models.CharField(max\_length=20)

introduction = models.CharField(max\_length=500)

website = models.CharField(max\_length=100,null=True)

img\_address = models.CharField(max\_length=100,null=True)

intro\_address = models.CharField(max\_length=100,null=True)

def \_\_str\_\_(self):

return self.company\_id

class users(models.Model):

usertype\_list = (

(0,'临时用户'),

(1, '用户'),

(2, '超级用户'),

)

user\_id = models.CharField(max\_length=18,primary\_key=True)

user\_name = models.CharField(max\_length=6,unique=True)

email = models.CharField(max\_length=30,unique=True)

mobilphone = models.IntegerField(null=True,unique=True)

password = models.CharField(max\_length=20)

user\_type = models.IntegerField(choices=usertype\_list)

user\_authority = models.IntegerField(null=True)

createtime = models.DateTimeField(auto\_now\_add=True,null=True)

updatetime = models.DateTimeField(auto\_now=True,null=True)

def \_\_str\_\_(self):

return self.user\_id

class EmailVerifyRecord(models.Model):

vcode\_id = models.CharField(max\_length=5,primary\_key=True)

user\_name = models.CharField(max\_length=6,null=True)

email = models.CharField(max\_length=30, unique=True)

vcode = models.CharField(max\_length=18)

send\_type = models.CharField(max\_length=10,choices=((0,u"注册"), (1,u"找回密码"),(2,u"修改邮箱")),default=0)

send\_time = models.DateTimeField(auto\_now=True,null=True)

class Algorithm\_description(models.Model):

algorithm\_id=models.CharField(max\_length=20,primary\_key=True)

algorithm\_name=models.CharField(max\_length=100)

algorithm\_type=models.CharField(max\_length=4,null=True)

algorithm\_img=models.ImageField(null=True)

algorithm\_text=models.CharField(max\_length=10000,null=True)

def \_\_str\_\_(self):

return self.algorithm\_id

class period(models.Model):

period\_id = models.CharField(max\_length=1,primary\_key=True)

period\_name = models.CharField(max\_length=30,unique=True)

memo = models.CharField(max\_length=300,null=True,blank=True)

def \_\_str\_\_(self):

return self.period\_id

class evaluator(models.Model):

evaluator\_id = models.CharField(max\_length=1,primary\_key = True)

evaluator\_name = models.CharField(max\_length=30,unique=True)

memo = models.CharField(max\_length=300,null=True,blank=True)

def \_\_str\_\_(self):

return self.evaluator\_id

class senior\_index(models.Model):

senior\_index\_id = models.CharField(max\_length=2,primary\_key = True)

senior\_index\_name = models.CharField(max\_length=30,unique=True)

memo = models.CharField(max\_length=300,null=True,blank=True)

def \_\_str\_\_(self):

return self.senior\_index\_id

class junior\_index(models.Model):

junior\_index\_id = models.CharField(max\_length=6,primary\_key = True)

junior\_index\_name = models.CharField(max\_length=30,unique=True)

period\_id = models.ForeignKey("period", to\_field="period\_id", null=True,on\_delete = models.CASCADE)

evaluator\_id = models.ForeignKey("evaluator", to\_field="evaluator\_id", null=True,on\_delete = models.CASCADE)

senior\_index\_id = models.ForeignKey('senior\_index',to\_field='senior\_index\_id',null=True,on\_delete = models.CASCADE)

memo = models.CharField(max\_length=300,null=True,blank=True)

def \_\_str\_\_(self):

return self.junior\_index\_name

class ind\_model(models.Model):

ind\_model\_id = models.CharField(primary\_key=True,max\_length=100)

ind\_model\_name = models.CharField(max\_length=20,null=True)

ind\_model\_num = models.CharField(max\_length=2)

ind\_model\_type = models.CharField(max\_length=20)

ind\_model\_cat = models.CharField(max\_length=20,null=True,unique=True)

user\_id = models.ForeignKey('users',to\_field='user\_id',null=True,on\_delete = models.CASCADE)

company\_id = models.ForeignKey("company", to\_field="company\_id",null=True,on\_delete = models.CASCADE,related\_name='id\_ind')

data = models.CharField(max\_length=10000)

ctime = models.DateTimeField(auto\_now\_add=True,null = True)

def \_\_str\_\_(self):

return self.ind\_model\_name

class val\_model(models.Model):

datetype\_choices =(

(0, u'年份'),

(1, u"季度"),

(2, u"月份"),

)

val\_model\_id = models.CharField(max\_length=50,primary\_key=True)

val\_model\_name = models.CharField(max\_length=30)

ind\_model\_id = models.ForeignKey("ind\_model",to\_field="ind\_model\_id",on\_delete = models.CASCADE,related\_name='ind\_val')

ind\_model\_cat = models.ForeignKey("ind\_model",to\_field="ind\_model\_cat",on\_delete = models.CASCADE,related\_name='cat\_val',null=True)

user\_id = models.ForeignKey("users",to\_field="user\_id",on\_delete = models.CASCADE)

company\_id=models.ForeignKey("company",to\_field="company\_id",null=True,on\_delete = models.CASCADE,related\_name='id\_val')

data = models.CharField(max\_length=10000)

datatime = models.CharField(max\_length=8,null=True)

datetype = models.IntegerField(choices=datetype\_choices,null=True)

interval\_id = models.ForeignKey('interval', to\_field='interval\_id', on\_delete=models.CASCADE, null=True)

ctime = models.DateField(auto\_now\_add=True,null=True)

def \_\_str\_\_(self):

return self.val\_model\_id

class val\_res(models.Model):

val\_res\_id = models.CharField(max\_length=50,primary\_key=True)

val\_model\_id = models.ForeignKey("val\_model", to\_field="val\_model\_id", on\_delete=models.DO\_NOTHING,)

ind\_model\_cat\_id = models.CharField(max\_length=30,null=True)

final\_res = models.CharField(max\_length=30,null=True)

e\_res = models.CharField(max\_length=30,null=True)

p\_res = models.CharField(max\_length=30,null=True)

user\_id = models.ForeignKey("users",to\_field="user\_id",on\_delete = models.CASCADE)

algorithm\_id = models.ForeignKey("Algorithm\_description",to\_field="algorithm\_id",null=True,on\_delete = models.CASCADE)

company\_id = models.ForeignKey("company", to\_field="company\_id",null=True,on\_delete = models.CASCADE,related\_name='id\_res')

period\_res = models.TextField(max\_length=1000,null=True)

evaluator\_res = models.TextField(max\_length = 1000,null=True)

senior\_res = models.TextField(max\_length=1000,null=True)

junior\_res = models.TextField(max\_length=1000,null=True)

p\_index\_res = models.TextField(max\_length=1000,null=True)

ctime = models.DateField(auto\_now\_add=True,null=True)

period\_plot=models.TextField(null=True)

evaluator\_plot=models.TextField(null=True)

senior\_detail\_plot=models.TextField(null=True)

class interval(models.Model):

interval\_id = models.CharField(max\_length=10,primary\_key=True)

interval\_data = models.CharField(max\_length=100,null=True)

def \_\_str\_\_(self):

return self.interval\_id

class def\_index(models.Model):

junior\_index\_id=models.CharField(max\_length=6,primary\_key=True)

period\_name=models.CharField(max\_length=30,null=True)

evaluator\_name=models.CharField(max\_length=30,null=True)

junior\_index\_name=models.CharField(max\_length=30,null=True)

senior\_index\_name=models.CharField(max\_length=30,null=True)

memo=models.CharField(max\_length=30,default=None)

def \_\_str\_\_(self):

return self.junior\_index\_id

class user\_feedback(models.Model):

feedback\_id = models.CharField(max\_length=32,primary\_key=True)

user\_id = models.ForeignKey("users",to\_field="user\_id",on\_delete = models.CASCADE)

evaluation\_content=models.TextField()

ctime = models.DateTimeField(auto\_now=True)

def \_\_str\_\_(self):

return self.feedback\_id

# urls.py

from evasysapp import views

from django.conf.urls import url,include

urlpatterns = [

url(r'^login/', views.login),

url(r'^home/', views.home),

url(r'^defindex/', views.defindex),

url(r'^chanindex/', views.chanindex),

url(r'^creaindex/',views.creaindex),

url(r'^detail/',views.detail),

url(r'^sendecode/',views.sendecode),

url(r'^register/',views.register),

url(r'^sucregister/',views.sucregister),

url(r'^modscope/',views.modscope),

url(r'^modalgorithm/',views.modalgorithm),

url(r'^moddescribe/',views.moddescribe),

url(r'^modevaluation/',views.modevaluation),

url(r'^indexval/',views.indexval),

url(r'^newindex/',views.newindex),

url(r'^calculate/',views.calculate),

url(r'^resvisualization/',views.resvisualization),

url(r'^resshow/',views.resshow),

url(r'^EXEMtheory/',views.EXEMtheory),

url(r'^EXEMfeature/',views.EXEMfeature),

url(r'^EXEMmodel/',views.EXEMmodel),

url(r'^EWMtheory/',views.EWMtheory),

url(r'^EWMfeature/',views.EWMfeature),

url(r'^EWMmodel/',views.EWMmodel),

url(r'^AHPtheory/',views.AHPtheory),

url(r'^AHPfeature/',views.AHPfeature),

url(r'^AHPmodel/',views.AHPmodel),

url(r'^CMtheory/',views.CMtheory),

url(r'^CMfeature/',views.CMfeature),

url(r'^CMmodel/',views.CMmodel),

url(r'^getexcel/',views.getexcel),

url(r'^Usercenter/', views.Usercenter),

url(r'^sendecode\_email\_change/', views.sendecode\_email\_change),

url(r'^company\_info/', views.company\_info),

url(r'^company\_del/', views.company\_del),

url(r'^email\_info/', views.email\_info),

url(r'^code\_info/', views.code\_info),

url(r'^user\_feedback/', views.user\_feedback),

url(r'^historical\_eval/', views.historical\_eval),

]

# calculate.js

$(function () {

var oTable = new TableInit();

oTable.Init();

var oButtonInit = new ButtonInit();

oButtonInit.Init();

});

var TableInit =function(){

var oTableInit = new Object();

oTableInit.Init = function () {

$('#table').bootstrapTable({

method: 'get',

url: "/evasys/calculate/",

striped: true,

cache: false,

toolbar:"#toolbar",

clickToSelect: true,

showExport: true,

exportTypes: ['excel'],

exportDataType: 'basic',

Icons: 'glyphicon-export',

search: true,

advancedSearch: true,

idTable: 'advancedtable',

showRefresh: true,

showToggle: true,

showColumns: true,

toolbarAlign: 'right',

buttonsAlign: 'left',

searchOnEnterKey: true,

singleSelect:false,

onEditableSave: function (field, row, oldValue, $el) {

$.ajax({

type: "post",

url: "/Editable/chanindex/",

data: {strJson: JSON.stringify(row)},

success: function (data, status) {

if (status == "success") {

alert("编辑成功");

}

},

error: function () {

alert("Error");

},

complete: function () {

}

});

}

});

$('#table1').bootstrapTable({

method: 'get',

url:"/evasys/calculate/",

striped: true,

cache: false,

clickToSelect: true,

showExport: true,

exportTypes: ['excel'],

exportDataType: 'basic',

Icons: 'glyphicon-export',

search: true,

advancedSearch: true,

idTable: 'advancedtable',

showRefresh: true,

showToggle: true,

showColumns: true,

toolbarAlign: 'right',

buttonsAlign: 'left',

searchOnEnterKey: true,

singleSelect:true,

onEditableSave: function (field, row, oldValue, $el) {

$.ajax({

type: "post",

url: "/Editable/chanindex/",

data: {strJson: JSON.stringify(row)},

success: function (data, status) {

if (status == "success") {

alert("编辑成功");

}

},

error: function () {

alert("Error");

},

complete: function () {

}

});

}

});

};

oTableInit.queryParams = function (params) {

var temp = {

limit: params.limit,

offset: params.offset,

departmentname: $("#txt\_search\_departmentname").val(),

statu: $("#txt\_search\_statu").val()

};

return temp;

};

return oTableInit;

};

var ButtonInit = function () {

var oInit = new Object();

var postdata = {};

oInit.Init = function () {

};

return oInit;

};

# chanindex.js

$(function () {

var oTable = new TableInit();

oTable.Init();

var oButtonInit = new ButtonInit();

oButtonInit.Init();

});

var TableInit =function(){

var oTableInit = new Object();

oTableInit.Init = function () {

$('#table').bootstrapTable({

method: 'get',

url: "/evasys/chanindex/",

toolbar: '#toolbar',

striped: true,

cache: false,

clickToSelect: true,

showExport: true,

exportTypes: ['excel','xlsx'],

exportDataType: 'basic',

Icons: 'glyphicon-export',

search: true,

advancedSearch: true,

idTable: 'advancedtable',

showRefresh: true,

showToggle: true,

showColumns: true,

toolbarAlign: 'right',

buttonsAlign: 'left',

searchOnEnterKey: true,

onEditableSave: function (field, row, oldValue, $el) {

$.ajax({

type: "post",

url: "/Editable/chanindex/",

data: {strJson: JSON.stringify(row)},

success: function (data, status) {

if (status == "success") {

alert("编辑成功");

}

},

error: function () {

alert("Error");

},

complete: function () {

}

});

}

});

};

if(!cycle){

alert('请选择数据周期！');

return false

}

for(var i in data\_rec){

if(!datas[i]){

datas[i] = new Array();

var obj = document.getElementById(i);

var val = obj.value

datas[i] = val

}

if(val == ""){

alert("请将二级 指标取值填写完整");

}else if(parseInt(val)>10 || parseInt(val)<0 ){

alert("请填写1-10的数字");

}

}

$.ajax({

url:"/evasys/indexval/",

type:"POST",

data:{"data":JSON.stringify(datas),"id":ids,"cycle":cycle,"datetype":datetype,"interval\_id":interval\_id},

headers:{"X-CSRFtoken": $.cookie("csrftoken")},

success:function (arg) {

alert("保存成功")

},

error:function () {

alert("保存失败")

}

})

}

function showtable(target) {

var index = target.parentNode.parentNode.getAttribute("data-index");

ids = target.id;

var res\_data = {{ datas|safe }};

var table = document.getElementById("table1");

$("#table1 tr:not(:first)").empty("");

table.deleteRow(1);

for(var i = 0;i<res\_data.length;i++){

if(i == index){

data\_rec = res\_data[i];

console.log(data\_rec);

for(var m in data\_rec){

var newTr = table.insertRow();

for(j=0;j<=data\_rec[m].length;j++){

if(j===4){

var newTd1 = newTr.insertCell();

newTd1.innerText= '';

newTd1.innerHTML = "<input type = 'text' id = "+m+" style = 'width:140px' class='form-control' placeholder='请输入1-10的数字'>";

}else{

var newTd = newTr.insertCell();

newTd.innerText= data\_rec[m][j];

}

}

}

}

}

}

</script>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/bootstrap.min.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-zh-CN.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-export.js"></script>

<script type="text/javascript" src="/static/js/xlsx.core.min.js"></script>

<script type="text/javascript" src="/static/js/tableExport.min.js"></script>

<script type="text/javascript" src="/static/js/FileSaver.min.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-toolbar.js"></script>

<script type="text/javascript" src="/static/js/jquery.cookie.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-editable.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-datetimepicker.min.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-datetimepicker.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-editable.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-select.js"></script>

<script type="text/javascript" src="/static/js/indexval.js" ></script>

<script type="text/javascript">

$('#year').datetimepicker({

language: 'zh-CN',

format:'yyyy',

weekStart: 1,

autoclose: 1,

todayHighlight: 1,

startView: 4,

minView: 4,

forceParse: 0

});

$('#month').datetimepicker({

language: 'zh-CN',

format:'yyyymm',

weekStart: 1,

autoclose: 1,

todayHighlight: 1,

startView: 3,

minView: 3,

forceParse: 0

});

function show() {

var showid = document.getElementById('datetype').value;

var div = document.getElementsByName('dtype');

for(var i=0;i<3;i++){

div[i].style.display="none";

div[showid].style.display = "inline-block"

}

}

</script>

</body>

</html>

# login.html

<!DOCTYPE html>

<html>

<head>

<title>登录</title>

<meta charset="UTF-8" />

<META HTTP-EQUIV="pragma" CONTENT="no-cache">

<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache, must-revalidate">

<META HTTP-EQUIV="expires" CONTENT="0">

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/static/themes/icon.css" />

<link rel="stylesheet" type="text/css" href="/static/css/login.css" />

</head>

<body>

<div class="container" style="margin: 100px">

<div class="row">

<div class="col-lg-5 col-lg-offset-3">

<div class="panel panel-default">

<div class="panel-heading">

<h1 class="panel-title" style="padding: 0 190px">登录</h1>

</div>

<div class="panel-body">

<form action="/evasys/login/" method = 'post' class="form-horizontal" >

{% csrf\_token %}

<div class = 'form-group' style="padding: 10px 0">

<label for="username" class="col-lg-3 control-label">用户名：</label>

<div class="col-lg-7">

<input type="text" id="username" class="form-control" name = 'username' placeholder="请输入您的用户名">

</div>

</div>

<div class = 'form-group' style="padding: 30px 0px">

<label for = 'password' class="col-lg-3 control-label">密 &nbsp 码 ：</label>

<div class="col-lg-7">

<input type="password" id = 'password' name = 'password' class="form-control" placeholder="请输入密码">

</div>

<span>

<span style="color: red;">{{ error\_msg }}</span>

</span>

</div>

<div class="form-group">

<div class="" style="text-align: center">

<button type="submit" id = '' class="btn btn-primary" style="position: relative;left: 50px">登录</button>

</div>

</div>

<a type="button" id="register" class="btn btn-default" href="/evasys/register/" style="position: relative;left: 130px; bottom: 48px">注册</a>

</form>

</div>

</div>

</div>

</div>

</div>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/bootstrap.min.js"></script>

<script type="text/javascript" src="/static/js/easyui-lang-zh\_CN.js" ></script>

<script type="text/javascript" src="/static/js/index.js" ></script>

</body>

</html>

# modalgorithm.html

<!DOCTYPE html>

<html >

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>企业信息化绩效评价系统</title>

<link href="/static/css/moddescribe.css" rel="stylesheet" />

</head>

<body>

<div>

<div class="top1">

<p>模型算法</p>

<div class="top2">

<span>当前位置：</span>

<a href="">首页</a>

<span >></span>

<a href="">模型描述</a>

<span >></span>

<a href="">模型算法</a>

</div>

</div>

<div class="middle1">

<p>基于企业信息化绩效评价问题，我们采用可拓学中的可拓优度评价法作为评价方法。它是一种定性与定量相结合的评价方法，该方法允许评价指标既可以为定量指标也可以为定性指标，使得评价过程灵活易用。另外，该方法不仅可以对待评对象进行部分评价，也可以对待评对象的整体进行评价。同时该方法利用关联函数确定的关联度，衡量了企业信息化绩效所隶属的评价等级，并可以对属于同一等级内不同企业的信息化绩效进行比较。我们以求解一级指标企业战略c<sub>11</sub>的绩效值为例，该方法的具体评价步骤如下：</p>

<span class="middle2">

<p class="text1">1.确定评价指标体系</p>

<p>评价一个对象，如果没有评价标准，那么评价就没有依据，也就无法辨别评价结果的优劣，因此，需要确定待评对象的评价指标。一级指标企业战略c<sub>11</sub>所属二级指标分别为企业战略一致性c<sub>111</sub>、信息化的迫切性c<sub>112</sub>、信息化领导力c<sub>113</sub>。</p>

</span>

<span class="middle2">

<p class="text1">2.收集并处理指标数据</p>

<p>为了获取各指标变量的值，为企业信息化绩效评价提供数据支撑，以企业信息化绩效评估指标为依据，对于可直接获取的数据，可从系统中直接抓取；对于不能直接获取的数据，可从系统中抓取关联数据进行变化、换算等转化工作，获取指标数据值。然后对各指标数据进行归一化处理，得到企业信息化各指标的实际取值。具体地，我们得到二级指标企业战略一致性c<sub>111</sub>、信息化的迫切性c<sub>112</sub>、信息化领导力c<sub>113</sub>的归一化标准值分别为v<sub>111</sub>、v<sub>112</sub>、v<sub>113</sub>。</p>

</span>

<span class="middle2">

<p class="text1">3.确定指标经典域和节域</p>

<p>确定各评价指标关于各等级的经典域区间和指标取值范围的节域区间。我们假设当前一级指标有 t 个等级，经典域表示各指标关于某一特定评价等级时的取值范围，可由行业规范、专家经验等给出；节域表示指标实际可取值的区间范围，可以简单地认为是各个经典域的并集。由此，得到二级指标c<sub>111</sub>、c<sub>112</sub>、c<sub>113</sub>的经典域、节域如下：</p>

<div><img src="/static/img/3-1.jpg"></div>

<span class="middle2">

<p class="text1">4.确定指标权重</p>

<p>评价指标集中的各评价指标有轻有重，用权重代表各评价指标的相对重要性程度。为了更加科学准确地计算出各评价指标的权重，我们提供了可拓优度法、熵权法、层次分析法以及组合法等各种方法来确定评价指标的权重。我们得到二级指标c<sub>111</sub>、c<sub>112</sub>、c<sub>113</sub>的权重分别为α<sub>111</sub>、α<sub>112</sub>、α<sub>113</sub>。</p>

</span>

<span class="middle2">

<p class="text1">5.建立关联函数与计算关联度</p>

<p>首先引入可拓距的概念，设x0是实轴上的任意点，X=<a,b>是实域R上的任意区间，称</p>

<div><img src="/static/img/5-1.jpg"></div>

<p>为点x\_0区间X之间的可拓距。取值为负表明点在区间内，具体的数值表示点在区间中的位置。</p>

<p>以二级指标c<sub>111</sub>为例，其归一化标准值分别为v<sub>111</sub>，等级j(j=1,2…t)的经典域 ，节域 ，V<sub>j</sub>⊂V，当V<sub>j</sub>与V有公共端点v<sub>z</sub>时，若v<sub>111</sub>≠v<sub>z</sub>，则指标c<sub>11</sub>1关于等级j的关联函数为：</p>

<div><img src="/static/img/5-2.jpg"></div>

<p>其中，D(v<sub>111</sub>,V<sub>j</sub>,V)= ρ(v<sub>111</sub>,V)- ρ(v<sub>111</sub>,V<sub>j</sub> )。</p>

<p>当V<sub>j</sub>与V无公共端点v<sub>z</sub>时，则指标c<sub>111</sub>关于等级j的关联函数为：</p>

<div><img src="/static/img/5-3.jpg"></div>

<p>其中，D(v<sub>111</sub>,V<sub>j</sub>,V)= ρ(v<sub>111</sub>,V)- ρ(v<sub>111</sub>,V<sub>j</sub> )。</p>

<p>关联函数计算公式是由评价指标实际值和评价指标当前所处的评价等级共同决定的，通过建立关联函数计算出各评价指标关于各评价等级的关联度，反应了评价指标隶属于某评价等级的程度，如果关联度大于零，说明该评价指标属于此评价等级，如果关联度小于零，说明该评价指标不在此评价等级内。由此，我们得到二级指标c<sub>111</sub>关于各等级的关联度K<sub>j</sub> (v<sub>111</sub>)，j=1,2…t。</p>

</span>

<span class="middle2">

<p class="text1">6.计算综合关联度</p>

<p>若待评价对象关于各个评价指标的关联度已经计算出来，则对其进行归一化处理，令</p>

<div><img src="/static/img/6-1.jpg"></div>

<p>其中，K<sub>j</sub> (v<sub>111</sub>)表示二级指标c<sub>11</sub>1关于评价等级 j 的关联度。</p>

<p>同理，重复上述步骤，可分别得到二级指标c<sub>112</sub> 、c<sub>113</sub>关于评价等级 j 的关联度K<sub>j</sub> (v<sub>112</sub> )、K<sub>j</sub> (v<sub>113</sub> )，j=1,2…t。</p>

<p>随后计算一级指标c<sub>11</sub>关于各评价等级的综合关联度，其计算公式为：</p>

<div><img src="/static/img/6-2.jpg"></div>

<p>其中，α<sub>11i</sub>表示指标c<sub>11i</sub>的权重。</p>

</span>

<span class="middle2">

<p class="text1">7.确定该级指标的评价等级</p>

<p>根据上一步骤求解得到的待评价对象关于等级t的综合关联度之后，若待评价对象的综合关联度为</p><div><img src="/static/img/7-1.jpg"></div><p>，那么该评价绩效评价等级为：j\*级。</p>

</span>

<span class="middle2">

<p class="text1">8.确定下一级指标的量值</p><p>由步骤6），得到一级指标c<sub>11</sub>关于评价等级 j 的关联度K<sub>j</sub> (v<sub>11</sub> )，j=1,2…t，且0≤K<sub>j</sub> (v<sub>11</sub> )≤1。其中K<sub>j\* </sub> (v<sub>11</sub> )=max⁡(K<sub>j </sub>(v<sub>11</sub> ))</p>

<p>我们已知一级指标c<sub>11</sub>所属的评价等级为j^\*级，则一级指标c<sub>11</sub>的量值v<sub>11</sub> ϵV\_(j^\* )，且我们已知一级指标c<sub>11</sub>关于该等级的综合关联度K<sub>j\* </sub>(v<sub>11</sub> )，则根据关联函数公式当v<sub>11</sub> ϵV<sub>j\* </sub>时，</p><div><img src="/static/img/8-1.jpg"></div>

<p>求得v<sub>11</sub>。</p>

<p>①当V<sub>j</sub>与V有公共端点时，即j=1或t，我们以j=1为例。我们画出非线性函数y=ρ(x,V<sub>1</sub> )/(D(x,V<sub>1</sub>,V)+a<sub>111</sub>-b<sub>111</sub>) x∈V<sub>1</sub>且V<sub>1</sub>=[0,2]的图像，如图所示：</p>

<div><img src="/static/img/8-2.jpg"></div>

<p>如图所示，x∈[0,2]时，函数y=ρ(x,V<sub>1</sub> )/(D(x,V<sub>1</sub>,V)+a<sub>11</sub>1-b<sub>111</sub>)的取值范围为[0,0.5]，而K<sub>j</sub> (v<sub>11</sub> )∈[0,1]。因此我们先将K<sub>1</sub> (v<sub>11</sub> )与函数y=ρ(x,V<sub>1</sub> )/(D(x,V<sub>1</sub>,V)+a<sub>11</sub>1-b<sub>111</sub>)的取值范围对应，令K<sub>1</sub>' (v<sub>11</sub> )=0.5\*K<sub>1</sub> (v<sub>11</sub> )，然后代入关联函数公式，即0.5\*K<sub>1</sub> (v<sub>11</sub> )=ρ(v<sub>11</sub>,V<sub>1</sub> )/(D(v<sub>11</sub>,V<sub>1</sub>,V)+a<sub>11</sub>1-b<sub>111</sub>)，求得v<sub>11</sub>。若K<sub>2</sub> (v<sub>11</sub> )≤0.5，则表明一级指标c<sub>11</sub>距离等级2较远，则取v<sub>11</sub>∈[0,1]；若K<sub>2</sub> (v<sub>11</sub> )>0.5，则表明一级指标c<sub>11</sub>距离等级2较近，则取v<sub>11</sub>∈(1,2]。</p>

<p>②当V<sub>j</sub>与V无公共端点时，即j=2,3…t-1，我们以j=2为例</p>

<p>我们画出非线性函数y=ρ(x,V<sub>2</sub> )/(D(x,V<sub>2</sub>,V)+a<sub>112</sub>-b<sub>112</sub> ) x∈V<sub>2</sub>且V<sub>2</sub>=[2,4]的图像，如图所示:</p>

<div><img src="/static/img/8-3.jpg"></div>

<p>如图所示，x∈[2,4]时，函数y=ρ(x,V<sub>2</sub> )/(D(x,V<sub>2</sub>,V)+a<sub>112</sub>-b<sub>112</sub> )的取值范围为[0,0.25]，而K<sub>j</sub> (v<sub>11</sub> )∈[0,1]。因此我们先将K<sub>2</sub> (v<sub>11</sub> )与函数y=ρ(x,V<sub>2</sub> )/(D(x,V<sub>2</sub>,V)+a<sub>112</sub>-b<sub>112</sub> )的取值范围对应，令K<sub>2</sub>' (v<sub>11</sub> )=0.25\*K<sub>2</sub> (v<sub>11</sub> )，然后代入关联函数公式，即0.25\*K<sub>2</sub> (v<sub>11</sub> )=ρ(x,V<sub>2</sub> )/(D(x,V<sub>2</sub>,V)+a<sub>112</sub>-b<sub>112</sub> )，求得v<sub>11</sub>。若K<sub>1</sub> (v<sub>11</sub> )≤K\_3 (v<sub>11</sub> )，则表明一级指标c<sub>11</sub>距离等级3较近，则取v<sub>11</sub>∈[3,4]；若K<sub>1</sub> (v<sub>11</sub> )>K\_3 (v<sub>11</sub> )，则表明一级指标c<sub>11</sub>距离等级1较近，则取v<sub>11</sub>∈[2,3)。</p>

<p class="text1">9.计算企业信息化全过程的综合得分</p><p>将第8）步获取的上一层指标的输入量值，循环第3）-8）步骤，循环迭代至最上层指标，确定企业信息化绩效整体综合得分和评价等级。</p>

</ul>

</div>

</body>

</html>

# moddescribe.html

<!DOCTYPE html>

<html >

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>企业信息化绩效评价系统</title>

<link href="/static/css/moddescribe.css" rel="stylesheet" />

</head>

<body>

<div>

<div class="top1">

<p>模型描述</p>

<div class="top2">

<span>当前位置：</span>

<a href="">首页</a>

<span >></span>

<a href="">模型描述</a>

<span >></span>

<a href="">模型描述</a>

</div>

</div>

<div class="middle1">

<p>根据绩效评价多个不同角色评价者的特征，以及企业信息化多阶段性的特征，并结合可拓理论，我们分别构建了选型、试用、租用三个阶段的多元多维可拓评价模型。其中多元是指多个评价者，多维是每类评价者有多个评价指标。该模型表示如下：

</p>

<span class="middle2">

<p class="text1">1.选型阶段</p>

<img src="/static/img/xxjd.png" >

<p>其中， P<sub>1</sub>表示选型阶段的企业信息化绩效，O<sub>1</sub>，O<sub>2</sub>，O<sub>3</sub>分别为选型阶段管理者、内控者和使用者，其一级评价指标为c<sub>mn</sub>，评价指标的量值为v<sub>mn</sub>。</p>

<p>对于一级指标c<sub>11</sub>，同样可以构建一元多维可拓模型。</p>

<img src="/static/img/xxjd2.png">

<p>V<sub>11</sub>表示指标c<sub>11</sub>的绩效值，c<sub>11</sub>的二级评价指标为c<sub>mnq</sub>，评价指标的量值为v<sub>mnq</sub>。这种两级评价指标结构不可改变，其值由云系统和大数据获得。</p>

</span>

<span class="middle2">

<p class="text1">2.试用阶段</p>

<img src="/static/img/syjd.png">

<p>其中， P<sub>2</sub>表示试用阶段的企业信息化绩效，O<sub>4</sub>，O<sub>5</sub>，O<sub>6</sub>分别为试用阶段管理者、内控者和使用者，其一级评价指标为c<sub>mn</sub>，评价指标的量值为v<sub>mn</sub>。</p>

</span>

<span class="middle2">

<p class="text1">3.租用阶段</p>

<img src="/static/img/zyjd.png">

<p>其中， P<sub>3</sub>表示租用阶段的企业信息化绩效，O<sub>7</sub>，O<sub>8</sub>，O<sub>9</sub>分别为租用阶段管理者、内控者和使用者，其一级评价指标为c<sub>mn</sub>，评价指标的量值为v<sub>mn</sub>。</p>

</span>

</div>

</div>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/jquery.easyui.min.js"></script>

<script type="text/javascript" src="/static/js/easyui-lang-zh\_CN.js" ></script>

<script type="text/javascript" src="/static/js/common.js" ></script>

<script type="text/javascript" src="/static/js/index.js" ></script>

</body>

</html>

# modevaluation.html

<!DOCTYPE html>

<html >

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>企业信息化绩效评价系统</title>

<link href="/static/css/moddescribe.css" rel="stylesheet" />

</head>

<body>

<div>

<div class="top1">

<p>模型评价</p>

<div class="top2">

<span>当前位置：</span>

<a href="">首页</a>

<span >></span>

<a href="">模型描述</a>

<span >></span>

<a href="">模型评价</a>

</div>

</div>

<div class="middle1">

<p>可拓评价模型是应用可拓学理论，将物元理论和可拓集合论相结合。它具有如下的特征：</p>

<ol>

<li>它提出了突破人脑习惯领域的方法，使人们能够按照一定的程序提出开拓的路径。</li>

<li>它采用了物元模型来描述问题，物元模型是利用物元和物元关系描述现象或问题的模型。</li>

<li>它采用了物元模型来描述问题，物元模型是利用物元和物元关系描述现象或问题的模型。</li>

<li>可拓方法利用菱形思维方法去描述人们进行创新性思维的过程，菱形思维方法包括发散过程和收敛过程。</li>

<li>发散过程是把问题用物元模型表达，然后从某一物元出发，沿不同的途径，开拓出一批物元，为解决问题提供丰富的信息。</li>

<li>收敛过程则是在发散过程的基础上，根据客观条件的限制和解决不同问题的不同需要，从可行性、优劣性、真伪性和相容性出发，对发散过程得到的大量物元进行物元变换或综合处理，从而得出评价结果。

</li>

</ol>

<span class="middle2">

<p>

正是基于这些特征，同时结合企业信息化绩效评价多个不同角色评价者的特征，以及企业信息化多阶段性的特征，可拓理论与企业信息化绩效评价问题有机地结合在一起，构建出多元多维可拓评价模型，从而有效地改善了现有信息化绩效评价模型只能反映绩效某个侧面或视角、无法反映绩效整体和不同评价者观点的问题，对企业信息化绩效进行了全方位、深层次的综合评价。

</p>

</span>

</div>

</div>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/jquery.easyui.min.js"></script>

<script type="text/javascript" src="/static/js/easyui-lang-zh\_CN.js" ></script>

</body>

</html>

# modscope.html

<!DOCTYPE html>

<html >

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>企业信息化绩效评价系统</title>

<link href="/static/css/moddescribe.css" rel="stylesheet" />

</head>

<body>

<div>

<div class="top1">

<p>模型范围</p>

<div class="top2">

<span>当前位置：</span>

<a href="">首页</a>

<span >></span>

<a href="">模型描述</a>

<span >></span>

<a href="">模型范围</a>

</div>

</div>

<div class="middle1">

<p>可拓学是一种解决矛盾间题的形式化数学工具。基于可拓原理的优度评价方法是一种定性与定量相结合的评价方法。该方法在评价条件集中加入了“必须满足的条件”，使得评价更加切合实际；而且它允许评价条件集的评价指标，既可为数量型的定量指标描述，又可为非数量型的定性指标描述，使评价方法更加灵活易用；同时该方法利用关联函数所确定的合格度和优度，来反映一个方案或策略的利弊程度。再者，可拓集合成功的给出了用形式化的语言描述矛盾转化及量变与质变的规律，使矛盾问题可以用数学方法解决。可拓学给出的“点与区间之距”的概念，也是经典数学中的“点与区间之距离”的概念的扩展，精确表达了点在区间内不同位置时与区间的位置关系，而不像经典数学一样认为“当点在区间内时，点与区间的距离为0”。这为从“类内即为同”发展为“类内也为异”提供了定量化的工具。</p>

</div>

</div>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/jquery.easyui.min.js"></script>

<script type="text/javascript" src="/static/js/easyui-lang-zh\_CN.js" ></script>

</body>

</html>

# NoPage.html

<!DOCTYPE html>

<html>

<head>

<title>网页不存在</title>

<meta charset="UTF-8" />

<META HTTP-EQUIV="pragma" CONTENT="no-cache">

<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache, must-revalidate">

<META HTTP-EQUIV="expires" CONTENT="0">

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/static/css/bootstrapValidator.css" />

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap-select.min.css" />

</head>

<body> &nbsp没有该网页

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/bootstrapValidator.js"></script>

<script type="text/javascript" src="/static/js/bootstrap.min.js"></script>

</body>

</html>

# register.html

<!DOCTYPE html>

<html>

<head>

<title>注册</title>

<meta charset="UTF-8" />

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/static/css/register.css" />

<link rel="stylesheet" type="text/css" href="/static/css/bootstrapValidator.css" />

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap-select.min.css" />

</head>

<body>

<div class="container">

<div class="row">

<div class="col-lg-8 col-lg-offset-2">

<div class="panel panel-default">

<div class="panel-heading">

<h1 class="panel-title">注册</h1>

</div>

<div class="panel-body">

<form id = 'defaultForm' class="form-horizontal" action="/evasys/register/" method = 'post' enctype="multipart/form-data">

<div class="page-header">

<h4>企业信息</h4>

</div>

<div class = 'form-group'>

<label for="companyname" class="col-lg-3 control-label">企业名称：</label>

<div class="col-lg-5">

<input type="text" id="companyname" class="form-control" name = 'companyname' placeholder="请输入您所属企业名称">

</div>

</div>

<div class = 'form-group'>

<label for="liscom\_code" class="col-lg-3 control-label">上市公司代码：</label>

<div class="col-lg-5">

<input type="text" id="liscom\_code" class="form-control" name = 'liscom\_code' placeholder="请输入上市公司代码" value="000000">

</div>

</div>

<div class = 'form-group'>

<label for="industry\_id" class="col-lg-3 control-label">所属行业：</label>

<div class="col-lg-5">

<select class="form-control" name = 'industry'>

{% for row in industry %}

<option value= "{{ row.industry\_id }}">{{ row.industry\_name }}</option>

{% endfor %}

</select>

</div>

</div>

<div class = 'form-group'>

<label for="company\_address" class="col-lg-3 control-label">企业地址：</label>

<div class="col-lg-5">

<input type="text" id="company\_address" class="form-control" name = 'company\_address' placeholder="请输入企业地址">

</div>

</div>

<div class = 'form-group'>

<label for="telephone" class="col-lg-3 control-label">企业热线：</label>

<div class="col-lg-5">

<input type="tel" id="telephone" class="form-control" name = 'telephone' placeholder="请输入企业热线电话">

</div>

</div>

<div class = 'form-group'>

<label for="introduction" class="col-lg-3 control-label">简介：</label>

<div class="col-lg-5">

<input type="text" id="introduction" class="form-control" name = 'introduction' placeholder="请输入公司简介">

</div>

</div>

<div class = 'form-group'>

<label for="website" class="col-lg-3 control-label">网址：</label>

<div class="col-lg-5">

<input type="text" id="website" class="form-control" name = 'website' placeholder="请输入公司网址，可以为空">

</div>

</div>

<div class = 'form-group'>

<label for="docs" class="col-lg-3 control-label">企业介绍文档：</label>

<span class="col-lg-5">

<input type="file" id="docs" class="form-control" name = 'docs' placeholder="请输入企业热线电话">

</span>

</div>

<div class = 'form-group'>

<label for="imgs" class="col-lg-3 control-label">企业logo图片：</label>

<span class="col-lg-5">

<input type="file" id="imgs" class="form-control" name = 'imgs' placeholder="请输入企业热线电话">

</span>

</div>

<div class="page-header">

<h4>用户信息</h4>

</div>

<div class = 'form-group'>

<label for="username" class="col-lg-3 control-label">用户名：</label>

<div class="col-lg-5">

<input type="text" id="username" class="form-control" name = 'username' placeholder="请输入您的用户名" value="">

<span style="color: red;">{{ users\_msg }}</span>

</div>

<div style="color: gray;font-size: smaller">用户名只支持中文，<br/>长度为二到十之间</div>

</div>

<div class = 'form-group'>

<label for="password" class="col-lg-3 control-label">密码：</label>

<div class="col-lg-5">

<input type="password" id="password" class="form-control" name = 'password' placeholder="请输入您的密码">

</div>

</div>

<div class = 'form-group'>

<label for="repassword" class="col-lg-3 control-label">确认密码:</label>

<div class="col-lg-5">

<input type="password" id="repassword" class="form-control" name = 'repassword' placeholder="请再次输入您的密码">

</div>

</div>

<div class = 'form-group'>

<label for="email" class="col-lg-3 control-label">邮箱：</label>

<div class="col-lg-5">

<input type="email" id="email" class="form-control" name="email" placeholder="请输入您的邮箱">

</div>

</div>

<div class = 'form-group'>

<label for="emailcode" class="col-lg-3 control-label">邮箱验证码：</label>

<div class="col-lg-5">

<div class="input-group m-bot15">

<input type="text" id="emailcode" class="form-control" name = 'emailcode' placeholder="请输入邮箱验证码">

<span class="input-group-btn">

<input type="button" id="sendecode" value="获取验证码" class="btn btn-success">

</span>

</div>

<span style="color: red;">{{ err\_msg }}</span>

</div>

<div style="color: gray;font-size: smaller">&nbsp;&nbsp;验证码区分大小写</div>

</div>

<div class="form-ggroup">

<div class="col-lg-1 col-lg-offset-4">

<button type="submit" id = 'btn' class="btn btn-primary">注册</button>

<span style="color: red;">{{ error\_msg }}</span>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

<script>

function sendecode() {

var email=$('#email').val();

var username = $('#username').val();

if (email==="") {

alert("email不能为空")

} else{

$.ajax({

url: "/evasys/sendecode/",

type: 'POST',

data: { 'email':email,'user\_name':username},

headers:{"X-CSRFtoken": $.cookie("csrftoken")},

success:function (msg) {

console.log(msg);

alert(msg);

}

})

}

}

var wait=60;

function time(o) {

if (wait === 0) {

o.removeAttribute("disabled");

o.value="获取验证码";

wait = 60;

} else {

o.setAttribute("disabled", true);

o.value=wait+"秒后可重发";

wait--;

setTimeout(function() {

time(o)

},

1000)

}

}

document.getElementById("sendecode").onclick=function(){sendecode(this);time(this)}

</script>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/bootstrapValidator.js"></script>

<script type="text/javascript" src="/static/js/bootstrap.min.js"></script>

<script type="text/javascript" src="/static/js/easyui-lang-zh\_CN.js" ></script>

<script type="text/javascript" src="/static/js/register.js" ></script>

<script type="text/javascript" src="/static/js/bootstrap-select.js" ></script>

<script type="text/javascript" src="/static/js/bootstrap.min.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-zh-CN.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-export.js"></script>

<script type="text/javascript" src="/static/js/tableExport.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-toolbar.js"></script>

<script type="text/javascript" src="/static/js/jquery.cookie.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-editable.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-editable.js"></script>

</body>

</html>

# resshow.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>企业信息化绩效评价系统</title>

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap-table.css" />

<link href="/static/themes/icon.css" rel="stylesheet" />

<link href="/static/css/bootstrap-editable.css" rel="stylesheet" type="text/css"/>

</head>

<body>

<div class="container" style="position: relative">

<h1>已提交评价指标体系模板</h1>

<table id = "table">

<thead>

<tr>

<th data-field="state" data-checkbox="true" ></th>

<th data-field = "operate">操作</th>

<th data-field = "company">企业名称</th>

<th data-field = "cat">指标体系</th>

<th data-field = "algorithm">算法</th>

<th data-field="evaluator\_name">创建时间</th>

<th data-field="senior\_index\_name">创建人</th>

</tr>

</thead>

<tbody>

{% for item in res\_list %}

<tr>

<td></td>

<td><a id = "{{ item.val\_res\_id }}" class="btn btn-primary btn-large" onclick="calculate(this)"><span class="glyphicon glyphicon-pencil" aria-hidden="true"></span>查看 </a></td>

<td>{{ item.company\_id.company\_name }}</td>

<td>{{ item.ind\_model\_cat\_id }}</td>

<td>{{ item.algorithm\_id.algorithm\_name }}</td>

<td>{{ item.ctime }}</td>

<td>{{ item.user\_id.user\_name}}</td>

</tr>

{% endfor %}

</tbody>

</table>

<div id="toolbar" class="btn-group" style="position: absolute;right: 250px">

<button id="btn\_delete" type="button" class="btn btn-primary" onclick="deleterow()">

<span class="glyphicon glyphicon-remove" aria-hidden="true"></span>删除

</button>

</div>

<script>

function calculate(target) {

ids = target.id;

console.log(ids);

window.location.href="/evasys/resvisualization/?id="+ids+"";

}

function deleterow() {

var table = $("table");

var table\_1 = document.getElementById("table");

var tbody = table\_1.children[1];

var trs = tbody.children;

var ids = {};

for(var i = 0;i<trs.length;i++ ){

var tr = trs[i];

console.log(tr)

if(tr.getAttribute("class") =='selected'){

var id = tr.children[1].firstChild.id;

ids[i] = id;

}

}

$.ajax({

url:"/evasys/resshow/",

type:"POST",

data:{"ids":JSON.stringify(ids)},

headers:{"X-CSRFtoken": $.cookie("csrftoken")},

success:function (arg) {

alert("删除成功")

},

error:function () {

alert("删除失败")

}

});

table.find("tr[class = 'selected']").hide();

}

</script>

</div>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/bootstrap.min.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-zh-CN.js"></script>

<script type="text/javascript" src="/static/js/jquery.cookie.js"></script>

<script type="text/javascript" src="/static/js/resshow.js" ></script>

</body>

</html>

# resvisualization.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>结果可视化</title>

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap.min.css" />

</head>

<body>

<h2 style="position: absolute;left: 450px">绩效评价结果</h2>

<div class="container" style="">

<button id = "download-paper" class="btn btn-default" style="position: absolute;top:50px;right: 30px">下载为PDF文件</button>

<h4 style="position: absolute;top: 50px;left: 55px">评价方法：{{ algorithmName }}</h4>

<h4 style="position: absolute;top: 50px;left: 355px">评价企业：{{ companyName }}</h4>

<h4 style="position: absolute;top: 80px;left: 55px">本次计算总体得分为{{ totalRes }}，评价映射为{{ totalEvaluation }}。</h4>

<h4 style="position: absolute; left: 55px;top: 110px;" class="glyphicon glyphicon-star">

总体展示:</h4>

<div id="main" style="width: 600px;height:400px;margin: 150px auto"></div>

<div id="main1" style="width: 500px;height:400px;margin: 100px auto;position: relative;top: -100px;"></div>

<div>

<h4 class="glyphicon glyphicon-star" style="display: inline-block;position: absolute;left: 55px;top: 1000px;">按评价阶段展示:</h4>

<select name="" id="selectPeriod" class="form-control" style="display: inline-block;position: relative; width: 150px;left: 180px;top: -195px">

</select>

</div>

<div id="main2" style="width: 500px;height:400px;margin: 100px auto;top: -180px"></div>

<div>

<h4 class="glyphicon glyphicon-star" style="display: inline-block;position: relative;left: -8px;top: -300px;">按评价者展示:</h4>

<select name="" id="selectEvaluator" class="form-control" style="display: inline-block;position: relative; width: 150px;left: 30px;top: -300px;">

</select>

</div>

<div id="main3" style="width: 500px;height:400px;margin: 50px auto;top: -350px"></div>

<h4 class="glyphicon glyphicon-star" style="display: inline-block;position: relative;left: -13px;top: -400px;">按一级评价指标展示:</h4>

<div id="main4" style="width: 800px;height:400px;margin: 50px auto;top: -380px;"></div>

</div>

<script src="/static/js/echarts.common.min.js"></script>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/bootstrap.min.js"></script>

<script src="/static/js/html2canvas.js" type="text/javascript" charset="utf-8"></script>

<script src="/static/js/jspdf.debug.js" type="text/javascript" charset="utf-8"></script>

<script src="/static/js/renderPDF.js" type="text/javascript" charset="utf-8"></script>

<script src="/static/js/paper\_download.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

var myChart = echarts.init(document.getElementById('main'));

var myChart1 = echarts.init(document.getElementById('main1'));

var myChart2 = echarts.init(document.getElementById('main2'));

var myChart3 = echarts.init(document.getElementById('main3'));

var myChart4 = echarts.init(document.getElementById('main4'));

var periodResDict = {{ periodResDict|safe }};

var evaluatorResDict = {{ evaluatorResDict|safe }};

var seniorResDict = {{ seniorResDict|safe }};

var periodName = periodResDict["评价阶段名称"];

var periodRes = periodResDict["评价阶段结果"];

var data = Array();

var periodNameList = Array();

for(index in periodName){

var valueArray = {};

var value = periodRes[index];

var valueNum = +value;

valueArray["name"] =periodName[index]+"阶段";

valueArray["value"] = valueNum;

data.push(valueArray);

periodNameList.push(periodName[index]+"阶段");

}

option = {

title : {

text: '评价得分总体情况',

x:'center'

},

tooltip : {

trigger: 'item',

formatter: "{a} <br/>{b} : {c} ({d}%)"

},

legend: {

orient : 'vertical',

x : 'left',

data:periodNameList

},

toolbox: {

show : true,

feature : {

mark : {show: true},

dataView : {show: true, readOnly: false},

magicType : {

show: true,

type: ['pie', 'funnel'],

option: {

funnel: {

x: '25%',

width: '50%',

funnelAlign: 'left',

max: 1548

}

}

},

restore : {show: true},

saveAsImage : {show: true}

}

},

series : [

{

type:'pie',

radius : '55%',

center: ['50%', '60%'],

data:data

}

]

};

myChart.setOption(option);

option1 = {

title : {

text: '评价得分总体情况',

x:'center'

},

tooltip : {

trigger: 'axis'

},

toolbox: {

show : true,

feature : {

mark : {show: true},

dataView : {show: true, readOnly: false},

magicType : {show: true, type: ['line', 'bar']},

restore : {show: true},

saveAsImage : {show: true}

}

},

calculable : true,

xAxis : [

{

type : 'category',

data : periodNameList

}

],

yAxis : [

{

type : 'value'

}

],

series : [{

itemStyle: {

normal: {

color: function(params) {

var colorList = [

'#C1232B','#B5C334','#FCCE10','#E87C25','#27727B',

'#FE8463','#9BCA63','#FAD860','#F3A43B','#60C0DD',

'#D7504B','#C6E579','#F4E001','#F0805A','#26C0C0'

];

return colorList[params.dataIndex]

},

label: {

show: true,

position: 'top',

formatter: '{b}\n{c}'

}

}

},

type:'bar',

radius : '55%',

center: ['50%', '60%'],

data:data

}]

};

myChart1.setOption(option1);

for(pN in periodNameList){

$("#selectPeriod").append("<option value='"+periodNameList[pN]+"'>"+periodNameList[pN]+"</option>")

}

var pName = $("#selectPeriod").val();

var periodNameDict = evaluatorResDict["评价阶段名称"];

var evaluatorNameDict = evaluatorResDict["评价者名称"];

var evaluatorResNumDict = evaluatorResDict['评价者结果'];

var evaluatorResArray = Array();

var evaluatorList = Array();

for(pIndex in periodNameDict){

if(periodNameDict[pIndex]+"阶段"==pName){

var subData = {};

subData["name"] = evaluatorNameDict[pIndex];

var valueString = evaluatorResNumDict[pIndex];

subData["value"] = +valueString;

evaluatorResArray.push(subData);

evaluatorList.push(evaluatorNameDict[pIndex])

}

}

option2 = {

title : {

text: pName+'得分情况',

x:'center'

},

tooltip : {

trigger: 'axis'

},

toolbox: {

show : true,

feature : {

mark : {show: true},

dataView : {show: true, readOnly: false},

magicType : {show: true, type: ['line', 'bar']},

restore : {show: true},

saveAsImage : {show: true}

}

},

calculable : true,

xAxis : [

{

type : 'category',

data : evaluatorList

}

],

yAxis : [

{

type : 'value'

}

],

series : [{

itemStyle: {

normal: {

color: function(params) {

var colorList = [

'#C1232B','#B5C334','#FCCE10','#E87C25','#27727B',

'#FE8463','#9BCA63','#FAD860','#F3A43B','#60C0DD',

'#D7504B','#C6E579','#F4E001','#F0805A','#26C0C0'

];

return colorList[params.dataIndex]

},

label: {

show: true,

position: 'top',

formatter: '{b}\n{c}'

}

}

},

type:'bar',

radius : '55%',

center: ['50%', '60%'],

data:evaluatorResArray

}]

};

myChart2.setOption(option2);

$("#selectPeriod").change(function () {

var pName = $("#selectPeriod").val();

var eName = $("#selectEvaluator").val();

var evaluatorResArray = Array();

var evaluatorList = Array();

for(pIndex in periodNameDict){

if(periodNameDict[pIndex]+"阶段"==pName){

var subData = {};

subData["name"] = evaluatorNameDict[pIndex];

subData["value"] = +evaluatorResNumDict[pIndex];

console.log(subData, "subData");

evaluatorResArray.push(subData);

evaluatorList.push(evaluatorNameDict[pIndex])

}else{continue}

document.getElementById("selectEvaluator").options.length=0;

for(pN in evaluatorList){

$("#selectEvaluator").append("<option value='"+evaluatorList[pN]+"'>"+evaluatorList[pN]+"</option>")

}

}

var subSeniorList = Array();

var subSeniorNameList = Array();

for(i in periodSeniorDict){

if(periodSeniorDict[i]+"阶段"==pName && evaluatorSeniorDict[i]==eName){

var sub = {};

sub["value"] = + seniorResNumDict[i];

sub["name"] = seniorNameDict[i];

subSeniorList.push(sub);

subSeniorNameList.push(seniorNameDict[i])

}

}

option3 = {

title : {

text: pName+eName+'得分情况',

x:'center'

},

tooltip : {

trigger: 'axis'

},

toolbox: {

show : true,

feature : {

mark : {show: true},

dataView : {show: true, readOnly: false},

magicType : {show: true, type: ['line', 'bar']},

restore : {show: true},

saveAsImage : {show: true}

}

},

calculable : true,

xAxis : [

{

type : 'category',

data : subSeniorNameList

}

],

yAxis : [

{

type : 'value'

}

],

series : [{

itemStyle: {

normal: {

color: function(params) {

var colorList = [

'#C1232B','#B5C334','#FCCE10','#E87C25','#27727B',

'#FE8463','#9BCA63','#FAD860','#F3A43B','#60C0DD',

'#D7504B','#C6E579','#F4E001','#F0805A','#26C0C0'

];

return colorList[params.dataIndex]

},

label: {

show: true,

position: 'top',

formatter: '{b}\n{c}'

}

}

},

type:'bar',

radius : '55%',

center: ['50%', '60%'],

data:subSeniorList

}]

};

myChart3.setOption(option3);

option2 = {

title : {

text: pName+'得分情况',

x:'center'

},

tooltip : {

trigger: 'axis'

},

toolbox: {

show : true,

feature : {

mark : {show: true},

dataView : {show: true, readOnly: false},

magicType : {show: true, type: ['line', 'bar']},

restore : {show: true},

saveAsImage : {show: true}

}

},

calculable : true,

xAxis : [

{

type : 'category',

data : evaluatorList

}

],

yAxis : [

{

type : 'value'

}

],

series : [{

itemStyle: {

normal: {

color: function(params) {

var colorList = [

'#C1232B','#B5C334','#FCCE10','#E87C25','#27727B',

'#FE8463','#9BCA63','#FAD860','#F3A43B','#60C0DD',

'#D7504B','#C6E579','#F4E001','#F0805A','#26C0C0'

];

return colorList[params.dataIndex]

},

label: {

show: true,

position: 'top',

formatter: '{b}\n{c}'

}

}

},

type:'bar',

radius : '55%',

center: ['50%', '60%'],

data:evaluatorResArray

}]

};

myChart2.setOption(option2);

});

var periodSeniorDict = seniorResDict["评价阶段名称"];

var evaluatorSeniorDict = seniorResDict["评价者名称"];

var seniorNameDict = seniorResDict["一级评价指标名称"];

var seniorResNumDict = seniorResDict["一级评价指标结果"];

var subSeniorList = Array();

var subSeniorNameList = Array();

var seniorNameList = Array();

var seniorDataList = Array();

for(pN in evaluatorList){

$("#selectEvaluator").append("<option value='"+evaluatorList[pN]+"'>"+evaluatorList[pN]+"</option>")

}

var eName = $("#selectEvaluator").val();

for(i in periodSeniorDict){

var seniorData = {};

seniorData["name"] = seniorNameDict[i];

seniorData["value"] = +seniorResNumDict[i];

seniorDataList.push(seniorData);

seniorNameList.push(seniorNameDict[i]);

if(periodSeniorDict[i]+"阶段"==pName && evaluatorSeniorDict[i]==eName){

var sub = {};

sub["value"] = + seniorResNumDict[i];

sub["name"] = seniorNameDict[i];

subSeniorList.push(sub);

subSeniorNameList.push(seniorNameDict[i])

}

}

option3 = {

title : {

text: pName+eName+'得分情况',

x:'center'

},

tooltip : {

trigger: 'axis'

},

toolbox: {

show : true,

feature : {

mark : {show: true},

dataView : {show: true, readOnly: false},

magicType : {show: true, type: ['line', 'bar']},

restore : {show: true},

saveAsImage : {show: true}

}

},

calculable : true,

xAxis : [

{

type : 'category',

data : subSeniorNameList

}

],

yAxis : [

{

type : 'value'

}

],

series : [{

itemStyle: {

normal: {

color: function(params) {

var colorList = [

'#C1232B','#B5C334','#FCCE10','#E87C25','#27727B',

'#FE8463','#9BCA63','#FAD860','#F3A43B','#60C0DD',

'#D7504B','#C6E579','#F4E001','#F0805A','#26C0C0'

];

return colorList[params.dataIndex]

},

label: {

show: true,

position: 'top',

formatter: '{b}\n{c}'

}

}

},

type:'bar',

radius : '55%',

center: ['50%', '60%'],

data:subSeniorList

}]

};

myChart3.setOption(option3);

$("#selectEvaluator").change(function () {

var pName = $("#selectPeriod").val();

var eName = $("#selectEvaluator").val();

var subSeniorList = Array();

var subSeniorNameList = Array();

for(i in periodSeniorDict){

if(periodSeniorDict[i]+"阶段"==pName && evaluatorSeniorDict[i]==eName){

var sub = {};

sub["value"] = + seniorResNumDict[i];

sub["name"] = seniorNameDict[i];

subSeniorList.push(sub);

subSeniorNameList.push(seniorNameDict[i])

}

}

option3 = {

title : {

text: pName+eName+'得分情况',

x:'center'

},

tooltip : {

trigger: 'axis'

},

toolbox: {

show : true,

feature : {

mark : {show: true},

dataView : {show: true, readOnly: false},

magicType : {show: true, type: ['line', 'bar']},

restore : {show: true},

saveAsImage : {show: true}

}

},

calculable : true,

xAxis : [

{

type : 'category',

data : subSeniorNameList

}

],

yAxis : [

{

type : 'value'

}

],

series : [{

itemStyle: {

normal: {

color: function(params) {

var colorList = [

'#C1232B','#B5C334','#FCCE10','#E87C25','#27727B',

'#FE8463','#9BCA63','#FAD860','#F3A43B','#60C0DD',

'#D7504B','#C6E579','#F4E001','#F0805A','#26C0C0'

];

return colorList[params.dataIndex]

},

label: {

show: true,

position: 'top',

formatter: '{b}\n{c}'

}

}

},

type:'bar',

radius : '55%',

center: ['50%', '60%'],

data:subSeniorList

}]

};

myChart3.setOption(option3);

});

option4 = {

title : {

text: '一级评价指标得分情况',

x:'center'

},

tooltip : {

trigger: 'axis'

},

toolbox: {

show : true,

feature : {

mark : {show: true},

dataView : {show: true, readOnly: false},

magicType : {show: true, type: ['line', 'bar']},

restore : {show: true},

saveAsImage : {show: true}

}

},

calculable : true,

xAxis : [

{

type : 'category',

data : seniorNameList

}

],

yAxis : [

{

type : 'value'

}

],

series : [{

itemStyle: {

normal: {

color: function(params) {

var colorList = [

'#C1232B','#B5C334','#FCCE10','#E87C25','#27727B',

'#FE8463','#9BCA63','#FAD860','#F3A43B','#60C0DD',

'#D7504B','#C6E579','#F4E001','#F0805A','#26C0C0'

];

return colorList[params.dataIndex]

},

label: {

show: true,

position: 'top',

formatter: '{b}\n{c}'

}

}

},

type:'bar',

radius : '55%',

center: ['50%', '60%'],

data:seniorDataList

}]

};

myChart4.setOption(option4);

     var downPdf = document.getElementById("download-paper");

     downPdf.onclick = function() {

         html2canvas(document.body, {

 onrendered:function(canvas) {

     var contentWidth = canvas.width;

     var contentHeight = canvas.height;

     var pageHeight = contentWidth / 592.28 \* 841.89;

     var leftHeight = contentHeight;

     var position = 0;

     var imgWidth = 595.28;

     var imgHeight = 592.28/contentWidth \* contentHeight;

     var pageData = canvas.toDataURL('image/jpeg', 1.0);

     var pdf = new jsPDF('', 'pt', 'a4');

     if (leftHeight < pageHeight) {

     pdf.addImage(pageData, 'JPEG', 0, 0, imgWidth, imgHeight );

     } else {

         while(leftHeight > 0) {

             pdf.addImage(pageData, 'JPEG', 0, position, imgWidth, imgHeight)

             leftHeight -= pageHeight;

             position -= 841.89;

             if(leftHeight > 0) {

               pdf.addPage();

             }

         }

     }

     pdf.save('{{ val\_res\_id }}.pdf');

 }

})

     }

</script>

</body>

</html>

# sucregister.html

<!DOCTYPE html>

<html>

<head>

<title>登录成功</title>

<meta charset="UTF-8" />

<META HTTP-EQUIV="pragma" CONTENT="no-cache">

<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache, must-revalidate">

<META HTTP-EQUIV="expires" CONTENT="0">

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/static/css/bootstrapValidator.css" />

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap-select.min.css" />

</head>

<body>

<div class="container" style="margin: 100px">

<div class="row">

<div class="col-lg-3 col-lg-offset-4">

<div class="panel panel-default">

<div class="panel-heading">

<h1 class="panel-title">恭喜您注册成功</h1>

</div>

<p>

<a href="/evasys/login/" class="btn btn-primary" style="margin: 50px 90px">确定</a>

</p>

</div>

</div>

</div>

</div>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/bootstrapValidator.js"></script>

<script type="text/javascript" src="/static/js/bootstrap.min.js"></script>

</body>

</html>

# user\_feedback.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>企业信息化绩效评价系统</title>

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap-table.css" />

<link href="/static/css/defindex.css" rel="stylesheet" />

<link href="/static/themes/icon.css" rel="stylesheet" />

<link href="/static/css/bootstrap-editable.css" rel="stylesheet" type="text/css"/>

<link href="/static/css/usercenter.css" rel="stylesheet" />

<style>

.hidden{

display: none;

}

</style>

</head>

<body>

<section>

<div class="ydc-content-slide ydc-body" id="body">

<div class="ydc-flex">

<div class="ydc-column ydc-column-8" >

<div class="ydc-release-content" >

<div class="ydc-tabPanel ydc-tabPanel-release">

<div class="ydc-release-tab-head">

<ul>

<li class="hit">用户反馈</li>

</ul>

</div>

<div style="margin-top: 60px">

<textarea class="form-control txt" rows="4" PLACEHOLDER="输入您的意见或建议" id="evaluation"></textarea>

<div style="float: right;width: 30%">

<button data-toggle="modal" data-target= "#txt" class="btn btn-primary btn-large" style="float: left" >提交</button>

</div>

</div>

<div class="modal fade" id="txt" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>

<h4 class="modal-title" id="myModalLabel">提示</h4>

</div>

<div class="modal-body">确认提交么？</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">取消</button>

<button class="btn btn-primary" data-dismiss="modal" value="确认提交" onclick="submit()">确认提交</button>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

<script>

function submit() {

var evaluation=$('#evaluation').val();

if (evaluation==="") {

alert("内容不能为空")

}

else{

$.ajax({

url: "/evasys/user\_feedback/",

type: 'POST',

data: {'evaluation':evaluation},

headers:{"X-CSRFtoken": $.cookie("csrftoken")},

success:function (msg) {

console.log(msg);

alert(msg);

location.reload()

}

})

}

}

</script>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/bootstrap.min.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-zh-CN.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-export.js"></script>

<script type="text/javascript" src="/static/js/tableExport.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-toolbar.js"></script>

<script type="text/javascript" src="/static/js/jquery.cookie.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-editable.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-editable.js"></script>

</body>

</html>

# Usercenter.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>企业信息化绩效评价系统</title>

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/static/css/bootstrap-table.css" />

<link href="/static/css/defindex.css" rel="stylesheet" />

<link href="/static/themes/icon.css" rel="stylesheet" />

<link href="/static/css/bootstrap-editable.css" rel="stylesheet" type="text/css"/>

<link href="/static/css/usercenter.css" rel="stylesheet" />

<style>

.hidden{

display: none;

}

</style>

</head>

<body>

<section>

<div class="ydc-content-slide ydc-body" id="body">

<div class="ydc-flex">

<div class="ydc-column ydc-column-8" >

<div class="ydc-release-content" >

<div class="ydc-tabPanel ydc-tabPanel-release" id="d1" name="dv" >

<div class="ydc-release-tab-head">

<ul>

<li class="hit">用户信息</li>

</ul>

</div>

<div>

<div class="ydc-pane ydc-pane-clear" style="display:block;">

<div class="ydc-reg-form-group clearfix">

<label>用户名</label>

<div class="ydc-reg-form-input" >

<label id="username">{{ user\_name }}</label>

</div>

</div>

<div class="ydc-reg-form-group clearfix">

<label>邮箱</label>

<div class="ydc-reg-form-input">

<label>{{ email }}</label>

</div>

</div>

<div class="ydc-reg-form-group clearfix">

<label>创建时间</label>

<div class="ydc-reg-form-input">

<label>{{ creat\_time }}</label>

</div>

</div>

<div class="ydc-reg-form-group clearfix">

<label>上次修改</label>

<div class="ydc-reg-form-input">

<label>{{ uptime }}</label>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/bootstrap.min.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-zh-CN.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-export.js"></script>

<script type="text/javascript" src="/static/js/tableExport.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-toolbar.js"></script>

<script type="text/javascript" src="/static/js/jquery.cookie.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-editable.js"></script>

<script type="text/javascript" src="/static/js/bootstrap-table-editable.js"></script>

<script type="text/javascript" src="/static/js/test2.js" ></script>

</body>

</html>

# Weightalgorithm.html

<!DOCTYPE html>

<html >

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title> Extension Superiority Evaluation Method</title>

<link href="/static/css/moddescribe.css" rel="stylesheet" />

</head>

<body>

<div>

<div class="top1">

<p>权重算法描述</p>

<div class="top2">

<span>当前位置：</span>

<a href="">首页</a>

<span >></span>

<a href="">权重算法描述</a>

</div>

</div>

<div class="middle1">

<p>评价指标权重算法模块的主要功能是提供不同的权重算法，以供用户选择，包括可拓优度法、熵权法、层次分析法和组合法等四种方法。其具体功能结构图如图3.9所示：</p>

<img src="/static/img/w-1.png"><br>

<span align="center">图3.9 评价指标权重算法选择功能结构图</span><br><br>

<p>假设某因素Z受n个因子X={x<sub>1</sub>,...,x<sub>n</sub>}影响，我们以X={x<sub>1</sub>,...,x<sub>n</sub>}这一维通用指标为例，分别介绍四种方法下求解指标权重的具体过程。</p><br>

</div>

</div>

<script type="text/javascript" src="/static/js/jquery.js"></script>

<script type="text/javascript" src="/static/js/jquery.easyui.min.js"></script>

<script type="text/javascript" src="/static/js/easyui-lang-zh\_CN.js" ></script>

</body>

</html>