**SOURCE CODE**

User Side views.py

**from** django.shortcuts **import** render, HttpResponse  
**from** django.contrib **import** messages  
**from** .forms **import** UserRegistrationForm  
**from** .models **import** UserRegistrationModel, UserSearchTweetsLocationModel, UserAlgorithmResultsModel  
**from** .TwitterClientAlgo **import** TwitterClient  
**from** .Tweetinfo **import** GetTweetLocatin  
**from** django\_pandas.io **import** read\_frame  
**from** django.core.paginator **import** Paginator, EmptyPage, PageNotAnInteger  
**from** .algorithms.UserNaiveBayes **import** UserNaiveBayesClass  
**from** .algorithms.UserSVMAlgorithm **import** UserSVMClass  
**from** .algorithms.UserDecisionTree **import** UserDecisionTreeClass  
  
*# Create your views here.***def** UserRegisterActions(request):  
 **if** request.method == **'POST'**:  
 form = UserRegistrationForm(request.POST)  
 **if** form.is\_valid():  
 print(**'Data is Valid'**)  
 form.save()  
 messages.success(request, **'You have been successfully registered'**)  
 form = UserRegistrationForm()  
 **return** render(request, **'UserRegistrations.html'**, {**'form'**: form})  
 **else**:  
 messages.success(request, **'Email or Mobile Already Existed'**)  
 print(**"Invalid form"**)  
 **else**:  
 form = UserRegistrationForm()  
 **return** render(request, **'UserRegistrations.html'**, {**'form'**: form})  
  
  
**def** UserLoginCheck(request):  
 **if** request.method == **"POST"**:  
 loginid = request.POST.get(**'loginname'**)  
 pswd = request.POST.get(**'pswd'**)  
 print(**"Login ID = "**, loginid, **' Password = '**, pswd)  
 **try**:  
 check = UserRegistrationModel.objects.get(loginid=loginid, password=pswd)  
 status = check.status  
 print(**'Status is = '**, status)  
 **if** status == **"activated"**:  
 request.session[**'id'**] = check.id  
 request.session[**'loggeduser'**] = check.name  
 request.session[**'loginid'**] = loginid  
 request.session[**'email'**] = check.email  
 print(**"User id At"**, check.id, status)  
 **return** render(request, **'users/UserHome.html'**, {})  
 **else**:  
 messages.success(request, **'Your Account Not at activated'**)  
 **return** render(request, **'UserLogin.html'**)  
 **except** Exception **as** e:  
 print(**'Exception is '**, str(e))  
 **pass** messages.success(request, **'Invalid Login id and password'**)  
 **return** render(request, **'UserLogin.html'**, {})  
  
  
**def** UserHome(request):  
 **return** render(request, **'users/UserHome.html'**, {})  
  
**def** UserGetTweetsForm(request):  
 **return** render(request,**'users/GetTweetForm.html'**,{})  
  
**def** GetTweets(request):  
 **if** request.method==**'POST'**:  
 hashtag = request.POST.get(**'tweettag'**)  
 print(**"Working"**)  
 *#api = TwitterClient()  
 #limit = 200  
 # calling function to get tweets  
 # tweets = api.get\_tweets(query=tagname, count=200)  
 #tweets = api.get\_tweets(query=hashtag, count=limit)  
 #print(type(tweets))* obj = GetTweetLocatin()  
 dataList,dataframe = obj.getLocations(hashtag)  
 dataframe = dataframe.to\_html()  
 **for** x **in** dataList:  
 tweetId = x[0]  
 username = x[1]  
 created\_at = x[2]  
 user\_screen\_name = x[3]  
 tweettext = x[4]  
 tweet\_location = x[5]  
 user\_loc = x[6]  
 *#print("Date type ",type(created\_at))* **if** len(tweet\_location)!=0:  
 *#print("Tweet Location ", tweet\_location)* lattitude,longitude,address = obj.getLatitudeLongitude(tweet\_location)  
 *#print(tweet\_location,"==",lattitude,address)* flag = 0  
 **if** user\_loc==**None**:  
 flag = 0  
 **else**:  
 flag = 1  
 UserSearchTweetsLocationModel.objects.create(tweetid=tweetId, username=username, userscreenname=user\_screen\_name,tweettext=tweettext,  
 createdat=created\_at,address=address,latitude=lattitude,longitude=longitude,userloc=flag)  
  
 **return** render(request,**"users/GetTweetsinfo.html"**,{**'data'**:dataframe})  
  
**def** UserViewDataset(request):  
 data\_list = UserSearchTweetsLocationModel.objects.all()  
 page = request.GET.get(**'page'**, 1)  
  
 paginator = Paginator(data\_list, 20)  
 **try**:  
 users = paginator.page(page)  
 **except** PageNotAnInteger:  
 users = paginator.page(1)  
 **except** EmptyPage:  
 users = paginator.page(paginator.num\_pages)  
 **return** render(request, **'users/UserViewDataSet.html'**,{**'users'**:users})  
  
**def** UserNaiveBayes(request):  
 data\_list = UserSearchTweetsLocationModel.objects.all()  
 df = read\_frame(data\_list)  
 obj = UserNaiveBayesClass()  
 accuracy,mae,mse,rmse,r\_squared = obj.getNaiveResults(df)  
 algorithmname = **"Naive Bayes"** username = request.session[**'loginid'**]  
 UserAlgorithmResultsModel.objects.create(username=username,algorithmname=algorithmname,accuracy=accuracy,mae=mae,mse=mse,rmse=rmse,r\_squared=r\_squared)  
 **return** render(request,**'users/NaiveResults.html'**,{**"accuracy"**:accuracy,**"mae"**:mae,**"mse"**:mse,**"rmse"**:rmse,**"r\_squared"**:r\_squared})  
  
**def** UserSVM(request):  
 data\_list = UserSearchTweetsLocationModel.objects.all()  
 df = read\_frame(data\_list)  
 obj = UserSVMClass()  
 accuracy, mae, mse, rmse, r\_squared = obj.getSVM(df)  
 algorithmname = **"SVM"** username = request.session[**'loginid'**]  
 UserAlgorithmResultsModel.objects.create(username=username, algorithmname=algorithmname, accuracy=accuracy, mae=mae,mse=mse, rmse=rmse, r\_squared=r\_squared)  
 **return** render(request, **'users/SVMResults.html'**,{**"accuracy"**: accuracy, **"mae"**: mae, **"mse"**: mse, **"rmse"**: rmse, **"r\_squared"**: r\_squared})  
  
**def** UserDecisionTree(request):  
 data\_list = UserSearchTweetsLocationModel.objects.all()  
 df = read\_frame(data\_list)  
 obj = UserDecisionTreeClass()  
 accuracy, mae, mse, rmse, r\_squared = obj.getDecisionTree(df)  
 algorithmname = **"Decision Tree"** username = request.session[**'loginid'**]  
 UserAlgorithmResultsModel.objects.create(username=username, algorithmname=algorithmname, accuracy=accuracy, mae=mae,  
 mse=mse, rmse=rmse, r\_squared=r\_squared)  
 **return** render(request, **'users/DecisionTreeResults.html'**,  
 {**"accuracy"**: accuracy, **"mae"**: mae, **"mse"**: mse, **"rmse"**: rmse, **"r\_squared"**: r\_squared})

User Side Models.py:

**from** django.db **import** models  
  
*# Create your models here.***class** UserRegistrationModel(models.Model):  
 name = models.CharField(max\_length=100)  
 loginid = models.CharField(unique=**True**, max\_length=100)  
 password = models.CharField(max\_length=100)  
 mobile = models.CharField(unique=**True**, max\_length=100)  
 email = models.CharField(unique=**True**, max\_length=100)  
 locality = models.CharField(max\_length=100)  
 address = models.CharField(max\_length=1000)  
 city = models.CharField(max\_length=100)  
 state = models.CharField(max\_length=100)  
 status = models.CharField(max\_length=100)  
  
 **def** \_\_str\_\_(self):  
 **return** self.loginid  
  
 **class** Meta:  
 db\_table = **'UserRegistrations'  
  
class** UserSearchTweetsLocationModel(models.Model):  
 tweetid = models.IntegerField()  
 username = models.CharField(max\_length=250)  
 userscreenname = models.CharField(max\_length=250)  
 tweettext = models.CharField(max\_length=1500)  
 createdat = models.DateTimeField()  
 address = models.CharField(max\_length=250,null=**True**)  
 latitude = models.FloatField()  
 longitude=models.FloatField()  
 userloc = models.IntegerField()  
 **def** \_\_str\_\_(self):  
 **return** self.tweetid  
 **class** Meta:  
 db\_table = **"TweetsTable"  
  
class** UserAlgorithmResultsModel(models.Model):  
 username = models.CharField(max\_length=100)  
 algorithmname = models.CharField(max\_length=100)  
 accuracy = models.FloatField()  
 mae = models.FloatField()  
 mse = models.FloatField()  
 rmse = models.FloatField()  
 r\_squared = models.FloatField()  
 cdate = models.DateTimeField(auto\_now\_add=**True**)  
 **def** \_\_str\_\_(self):  
 **return** self.id  
 **class** Meta:  
 db\_table = **"AlgorithmResults"**

user Side Forms.py:

**from** django **import** forms  
**from** .models **import** UserRegistrationModel  
  
  
**class** UserRegistrationForm(forms.ModelForm):  
 name = forms.CharField(widget=forms.TextInput(attrs={**'pattern'**: **'[a-zA-Z]+'**}), required=**True**, max\_length=100)  
 loginid = forms.CharField(widget=forms.TextInput(attrs={**'pattern'**: **'[a-zA-Z]+'**}), required=**True**, max\_length=100)  
 password = forms.CharField(widget=forms.PasswordInput(attrs={**'pattern'**: **'(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,}'**,  
 **'title'**: **'Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters'**}),  
 required=**True**, max\_length=100)  
 mobile = forms.CharField(widget=forms.TextInput(attrs={**'pattern'**: **'[56789][0-9]{9}'**}), required=**True**,  
 max\_length=100)  
 email = forms.CharField(widget=forms.TextInput(attrs={**'pattern'**: **'[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}$'**}),  
 required=**True**, max\_length=100)  
 locality = forms.CharField(widget=forms.TextInput(), required=**True**, max\_length=100)  
 address = forms.CharField(widget=forms.Textarea(attrs={**'rows'**: 4, **'cols'**: 22}), required=**True**, max\_length=250)  
 city = forms.CharField(widget=forms.TextInput(  
 attrs={**'autocomplete'**: **'off'**, **'pattern'**: **'[A-Za-z ]+'**, **'title'**: **'Enter Characters Only '**}), required=**True**,  
 max\_length=100)  
 state = forms.CharField(widget=forms.TextInput(  
 attrs={**'autocomplete'**: **'off'**, **'pattern'**: **'[A-Za-z ]+'**, **'title'**: **'Enter Characters Only '**}), required=**True**,  
 max\_length=100)  
 status = forms.CharField(widget=forms.HiddenInput(), initial=**'waiting'**, max\_length=100)  
  
 **class** Meta():  
 model = UserRegistrationModel  
 fields = **'\_\_all\_\_'**

Getting Tweet Search

**import** os  
**import** tweepy **as** tw  
**import** pandas **as** pd  
**from** geopy **import** geocoders  
**from** geopy.geocoders **import** Nominatim  
  
  
**class** GetTweetLocatin:  
 **def** getLocations(self, tweet):  
 search\_words = **"#"** + tweet  
 date\_since = **"2018-11-16"** consumer\_key = **'kU5EnStGNVvMU2J9Ni0dsiITj'** consumer\_secret = **'26HZTTQjiGSwBIhyiVt7LzlTXzD1d5WHgetr586FPS4rhbqVaj'** access\_token = **'1089021744336244737-XdMHpwoCZKZuPHqPwqjoG9v8vDEdjI'** access\_token\_secret = **'kq2cS2Q6IrwmadqbMKFBh2cfqJKTXhNvKW0Cki86W9oc6'** auth = tw.OAuthHandler(consumer\_key, consumer\_secret)  
 auth.set\_access\_token(access\_token, access\_token\_secret)  
 api = tw.API(auth, wait\_on\_rate\_limit=**True**)  
 *# Collect tweets* tweets = tw.Cursor(api.search, q=search\_words, lang=**"en"**, since=date\_since).items(50)  
 print(**"=====>"**, tweets.\_\_dict\_\_)  
  
 users\_locs = [[tweet.id, tweet.user.name, tweet.created\_at, tweet.user.screen\_name, tweet.text,  
 tweet.user.location, tweet.coordinates] **for** tweet **in** tweets]  
 *# print(users\_locs)* dataframe = pd.DataFrame(data=users\_locs,  
 columns=[**'Tweet ID'**, **'User Name'**, **'Created at'**, **'User Screen Name'**, **"Tweets"**,  
 **'Tweet Location'**, **'User Location'**])  
 *#gn = geocoders.GeoNames()  
 #print(gn.geocode("Cleveland, OH 44106"))* **return** users\_locs, dataframe  
  
 *# return users\_locs* **def** getLatitudeLongitude(self,cityname):  
 geolocator = Nominatim(user\_agent=**"datapointprojects13@gmail.com"**)  
 location = geolocator.geocode(cityname)  
 **try**:  
 **return** location.latitude, location.longitude,location.address  
 **except** Exception **as** ex:  
 **return** 0,0,**None**

Aglorithm Code:

**from** sklearn.model\_selection **import** train\_test\_split  
**from** sklearn.naive\_bayes **import** GaussianNB  
**from** sklearn.metrics **import** accuracy\_score, mean\_absolute\_error,mean\_squared\_error,r2\_score  
**import** math  
**import** numpy **as** np  
**class** UserNaiveBayesClass:  
 **def** getNaiveResults(self,df):  
 df = df[[**'latitude'**,**'longitude'**, **'userloc'**]]  
 print(**"df="**,df.head())  
 X = df[[**'latitude'**, **'longitude'**]]  
 y = df[[**'userloc'**]]  
 X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=1 / 3, random\_state=0)  
 model = GaussianNB()  
 model.fit(X\_train, y\_train)  
 y\_pred = model.predict(X\_test)  
 accuracy = accuracy\_score(y\_test,y\_pred)  
 mae = mean\_absolute\_error(y\_pred,y\_test)  
 mse = mean\_squared\_error(y\_pred,y\_test)  
 rmse = math.sqrt(mse)  
 r\_squared =r2\_score(y\_pred,y\_test)  
 print(**"Naivebayes"**,**"Accuracy = "**,accuracy,**"\t MAE="**,mae,**"\t MSE="**,mse,**"\t RMSE="**,rmse,**"\t r\_squared = "**,r\_squared)  
 *#return round(accuracy,2),round(mae,2),round(mse,2),round(rmse,2),round(r\_squared,2)* **return** accuracy,mae,mse,rmse,r\_squared

All urls.py

*"""LocationPrediction URL Configuration  
  
The `urlpatterns` list routes URLs to views. For more information please see:  
 https://docs.djangoproject.com/en/2.0/topics/http/urls/  
Examples:  
Function views  
 1. Add an import: from my\_app import views  
 2. Add a URL to urlpatterns: path('', views.home, name='home')  
Class-based views  
 1. Add an import: from other\_app.views import Home  
 2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')  
Including another URLconf  
 1. Import the include() function: from django.urls import include, path  
 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))  
"""***from** django.contrib **import** admin  
**from** django.urls **import** path  
**from** LocationPrediction **import** views **as** mainView  
**from** users **import** views **as** usr  
**from** admins **import** views **as** admins  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 path(**""**, mainView.index, name=**"index"**),  
 path(**"index/"**, mainView.index, name=**"index"**),  
 path(**"logout/"**, mainView.logout, name=**"logout"**),  
 path(**"UserLogin/"**, mainView.UserLogin, name=**"UserLogin"**),  
 path(**"AdminLogin/"**, mainView.AdminLogin, name=**"AdminLogin"**),  
 path(**"UserRegister/"**, mainView.UserRegister, name=**"UserRegister"**),  
  
 *### User Side Views* path(**"UserRegisterActions/"**, usr.UserRegisterActions, name=**"UserRegisterActions"**),  
 path(**"UserLoginCheck/"**, usr.UserLoginCheck, name=**"UserLoginCheck"**),  
 path(**"UserHome/"**, usr.UserHome, name=**"UserHome"**),  
 path(**"UserGetTweetsForm/"**, usr.UserGetTweetsForm, name=**"UserGetTweetsForm"**),  
 path(**"GetTweets/"**, usr.GetTweets, name=**"GetTweets"**),  
 path(**"UserViewDataset/"**, usr.UserViewDataset, name=**"UserViewDataset"**),  
 path(**"UserNaiveBayes/"**, usr.UserNaiveBayes, name=**"UserNaiveBayes"**),  
 path(**"UserSVM/"**, usr.UserSVM, name=**"UserSVM"**),  
 path(**"UserDecisionTree/"**, usr.UserDecisionTree, name=**"UserDecisionTree"**),  
  
  
 *### Admin Side Views* path(**"AdminLoginCheck/"**, admins.AdminLoginCheck, name=**"AdminLoginCheck"**),  
 path(**"AdminHome/"**, admins.AdminHome, name=**"AdminHome"**),  
 path(**"ViewRegisteredUsers/"**, admins.ViewRegisteredUsers, name=**"ViewRegisteredUsers"**),  
 path(**"AdminActivaUsers/"**, admins.AdminActivaUsers, name=**"AdminActivaUsers"**),  
 path(**"AdminNaiveBayes/"**, admins.AdminNaiveBayes, name=**"AdminNaiveBayes"**),  
 path(**"AdminSVM/"**, admins.AdminSVM, name=**"AdminSVM"**),  
 path(**"AdminDecisionTree/"**, admins.AdminDecisionTree, name=**"AdminDecisionTree"**),  
  
]

Admin side Views.py

**from** django.shortcuts **import** render  
**from** django.contrib **import** messages  
**from** users.models **import** UserRegistrationModel,UserAlgorithmResultsModel  
  
*# Create your views here.***def** AdminLoginCheck(request):  
 **if** request.method == **'POST'**:  
 usrid = request.POST.get(**'loginid'**)  
 pswd = request.POST.get(**'pswd'**)  
 print(**"User ID is = "**, usrid)  
 **if** usrid == **'admin' and** pswd == **'admin'**:  
 **return** render(request, **'admins/AdminHome.html'**)  
 **elif** usrid == **'Admin' and** pswd == **'Admin'**:  
 **return** render(request, **'admins/AdminHome.html'**)  
 **else**:  
 messages.success(request, **'Please Check Your Login Details'**)  
 **return** render(request, **'AdminLogin.html'**, {})  
  
  
**def** AdminHome(request):  
 **return** render(request, **'admins/AdminHome.html'**)  
  
  
**def** ViewRegisteredUsers(request):  
 data = UserRegistrationModel.objects.all()  
 **return** render(request, **'admins/RegisteredUsers.html'**, {**'data'**: data})  
  
  
**def** AdminActivaUsers(request):  
 **if** request.method == **'GET'**:  
 id = request.GET.get(**'uid'**)  
 status = **'activated'** print(**"PID = "**, id, status)  
 UserRegistrationModel.objects.filter(id=id).update(status=status)  
 data = UserRegistrationModel.objects.all()  
 **return** render(request, **'admins/RegisteredUsers.html'**, {**'data'**: data})  
  
**def** AdminNaiveBayes(request):  
 data = UserAlgorithmResultsModel.objects.filter(algorithmname=**"Naive Bayes"**)  
 **return** render(request,**'admins/AdminNaiveBayes.html'**,{**'data'**:data})  
  
**def** AdminSVM(request):  
 data = UserAlgorithmResultsModel.objects.filter(algorithmname=**"SVM"**)  
 **return** render(request,**'admins/AdminSVM.html'**,{**'data'**:data})  
  
**def** AdminDecisionTree(request):  
 data = UserAlgorithmResultsModel.objects.filter(algorithmname=**"Decision Tree"**)  
 **return** render(request, **'admins/AdminDecisionTree.html'**, {**'data'**: data})

base.html

<!doctype **html**>  
{%load static%}  
<**html class="no-js" lang="zxx"**>  
<**head**>  
 <**meta charset="utf-8"**>  
 <**meta http-equiv="x-ua-compatible" content="ie=edge"**>  
 <**title**>Tweets of locations </**title**>  
 <**meta name="description" content=""**>  
 <**meta name="viewport" content="width=device-width, initial-scale=1"**>  
  
 <**link rel="shortcut icon" type="image/x-icon" href="{%static 'img/favicon.ico'%}"**>  
  
 *<!-- CSS here -->* <**link rel="stylesheet" href="{%static 'css/bootstrap.min.css'%}"**>  
 <**link rel="stylesheet" href="{%static 'css/owl.carousel.min.css'%}"**>  
 <**link rel="stylesheet" href="{%static 'css/slicknav.css'%}"**>  
 <**link rel="stylesheet" href="{%static 'css/flaticon.css'%}"**>  
 <**link rel="stylesheet" href="{%static 'css/animate.min.css'%}"**>  
 <**link rel="stylesheet" href="{%static 'css/magnific-popup.css'%}"**>  
 <**link rel="stylesheet" href="{%static 'css/fontawesome-all.min.css'%}"**>  
 <**link rel="stylesheet" href="{%static 'css/themify-icons.css'%}"**>  
 <**link rel="stylesheet" href="{%static 'css/slick.css'%}"**>  
 <**link rel="stylesheet" href="{%static 'css/nice-select.css'%}"**>  
 <**link rel="stylesheet" href="{%static 'css/style.css'%}"**>  
</**head**>  
  
<**body class="body-bg"**>  
*<!--? Preloader Start -->*<**div id="preloader-active"**>  
 <**div class="preloader d-flex align-items-center justify-content-center"**>  
 <**div class="preloader-inner position-relative"**>  
 <**div class="preloader-circle"**></**div**>  
 <**div class="preloader-img pere-text"**>  
 <**img src="{%static 'img/logo/myloader.png'%}" alt=""**>  
 </**div**>  
 </**div**>  
 </**div**>  
</**div**>  
<**header**>  
 *<!-- Header Start -->* <**div class="header-area"**>  
 <**div class="main-header "**>  
 <**div class="header-top d-none d-lg-block"**>  
  
 </**div**>  
 <**div class="header-bottom header-sticky"**>  
 <**div class="container"**>  
 <**div class="row align-items-center"**>  
 *<!-- Logo -->* <**div class="col-xl-2 col-lg-2"**>  
 <**div class="logo"**>  
 <**a href="about.html" class="btn post-btn"**>Location Prediction on Twitter</**a**>  
 </**div**>  
 </**div**>  
 <**div class="col-xl-10 col-lg-10"**>  
 <**div class="menu-wrapper d-flex align-items-center justify-content-end"**>  
 *<!-- Main-menu -->* <**div class="main-menu d-none d-lg-block"**>  
 <**nav**>  
 <**ul id="navigation"**>  
 <**li**><**a href="{%url 'index'%}"**>Home</**a**></**li**>  
 <**li**><**a href="{%url 'UserLogin'%}"**>User</**a**></**li**>  
 <**li**><**a href="{%url 'AdminLogin'%}"**>Admin</**a**></**li**>  
 <**li**><**a href="{%url 'UserRegister'%}"**>Registrations</**a**></**li**>  
 </**ul**>  
 </**nav**>  
 </**div**>  
 </**div**>  
 </**div**>  
 *<!-- Mobile Menu -->* <**div class="col-12"**>  
 <**div class="mobile\_menu d-block d-lg-none"**></**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 *<!-- Header End -->*</**header**>  
<**main**>  
 {%block contents%}  
  
 {%endblock%}  
</**main**>  
<**footer**>  
 *<!--? Footer Start-->* <**div class="footer-area section-bg" data-background="{%static 'img/gallery/footer\_bg.jpg'%}"**>  
 <**div class="container"**>  
 <**div class="footer-top footer-padding"**>  
  
 </**div**>  
 <**div class="footer-bottom"**>  
 <**div class="row d-flex justify-content-between align-items-center"**>  
 <**div class="col-xl-9 col-lg-8"**>  
 <**div class="footer-copy-right"**>  
 <**p**>*<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->* Copyright **&copy;**<**script**>document.write(new Date().getFullYear());</**script**> All rights reserved | This template is made with <**i class="fa fa-heart" aria-hidden="true"**></**i**> by Alex Corporation  
 *<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->*</**p**>  
 </**div**>  
 </**div**>  
  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 *<!-- Footer End-->*</**footer**>  
  
 *<!-- JS here -->* <**script src="{%static 'js/vendor/modernizr-3.5.0.min.js'%}"**></**script**>  
 *<!-- Jquery, Popper, Bootstrap -->* <**script src="{%static 'js/vendor/jquery-1.12.4.min.js'%}"**></**script**>  
 <**script src="{%static 'js/popper.min.js'%}"**></**script**>  
 <**script src="{%static 'js/bootstrap.min.js'%}"**></**script**>  
 *<!-- Jquery Mobile Menu -->* <**script src="{%static 'js/jquery.slicknav.min.js'%}"**></**script**>  
  
 *<!-- Jquery Slick , Owl-Carousel Plugins -->* <**script src="{%static 'js/owl.carousel.min.js'%}"**></**script**>  
 <**script src="{%static 'js/slick.min.js'%}"**></**script**>  
 *<!-- One Page, Animated-HeadLin -->* <**script src="{%static 'js/wow.min.js'%}"**></**script**>  
 <**script src="{%static 'js/animated.headline.js'%}"**></**script**>  
 <**script src="{%static 'js/jquery.magnific-popup.js'%}"**></**script**>  
  
 *<!-- Nice-select, sticky -->* <**script src="{%static 'js/jquery.nice-select.min.js'%}"**></**script**>  
 <**script src="{%static 'js/jquery.sticky.js'%}"**></**script**>  
  
 *<!-- counter , waypoint -->* <**script src="http://cdnjs.cloudflare.com/ajax/libs/waypoints/2.0.3/waypoints.min.js"**></**script**>  
 <**script src="{%static 'js/jquery.counterup.min.js'%}"**></**script**>  
  
 *<!-- contact js -->* <**script src="{%static 'js/contact.js'%}"**></**script**>  
 <**script src="{%static 'js/jquery.form.js'%}"**></**script**>  
 <**script src="{%static 'js/jquery.validate.min.js'%}"**></**script**>  
 <**script src="{%static 'js/mail-script.js'%}"**></**script**>  
 <**script src="{%static 'js/jquery.ajaxchimp.min.js'%}"**></**script**>  
  
 *<!-- Jquery Plugins, main Jquery -->* <**script src="{%static 'js/plugins.js'%}"**></**script**>  
 <**script src="{%static 'js/main.js'%}"**></**script**>  
  
 </**body**>  
</**html**>

User Registrationform.html

{%extends 'base.html'%}  
{%load static%}  
{%block contents%}  
 <**div class="testimonial-area testimonial-padding" data-background="{%static 'img/gallery/section\_bg04.jpg'%}"**>  
 <**div class="container "**>  
 <**div class="row d-flex justify-content-center"**>  
 <**div class="col-xl-10 col-lg-10 col-md-9"**>  
 <**div class="h1-testimonial-active"**>  
 *<!-- Single Testimonial -->* <**div class="single-testimonial text-center"**>  
 *<!-- Testimonial Content -->* <**div class="testimonial-caption "**>  
 <**div class="testimonial-top-cap"**>  
 <**center**>  
 <**form action="{%url 'UserRegisterActions'%}" method="POST" class="text-primary" style="**width:100%**"**>  
 <**h1**><**font color="WHITE"**>User Registration form </**font**></**h1**>  
 {% csrf\_token %}  
 <**table**>  
 <**tr**><**td**></**td**>  
 <**td class="text-primary"**>Customer Name</**td**>  
 <**td**>{{form.name}}</**td**>  
 </**tr**>  
 <**tr**><**td**></**td**>  
 <**td**>Login ID</**td**>  
 <**td**>{{form.loginid}}</**td**>  
 </**tr**>  
 <**tr**><**td**></**td**>  
 <**td**>Password</**td**>  
 <**td**>{{form.password}}</**td**>  
 </**tr**>  
 <**tr**><**td**></**td**>  
 <**td**>Mobile</**td**>  
 <**td**>{{form.mobile}}</**td**>  
 </**tr**>  
 <**tr**><**td**></**td**>  
 <**td**>email</**td**>  
 <**td**>{{form.email}}</**td**>  
 </**tr**>  
 <**tr**><**td**></**td**>  
 <**td**>Locality</**td**>  
 <**td**>{{form.locality}}</**td**>  
 </**tr**>  
 <**tr**><**td**></**td**>  
 <**td**>Address</**td**>  
 <**td**>{{form.address}}</**td**>  
 </**tr**>  
 <**tr**><**td**></**td**>  
 <**td**>City</**td**>  
 <**td**>{{form.city}}</**td**>  
 </**tr**>  
 <**tr**><**td**></**td**>  
 <**td**>State</**td**>  
 <**td**>{{form.state}}</**td**>  
 </**tr**>  
 <**tr**><**td**></**td**>  
 <**td**></**td**>  
 <**td**>{{form.status}}</**td**>  
 </**tr**>  
 <**tr**><**td**></**td**>  
 <**td**></**td**>  
 <**td**><**button type="submit" value="Register" class="button button-contactForm boxed-btn"**>Register</**button**></**td**>  
 </**tr**>  
 <**tr**>  
 <**td**>  
 <**div class="form-group mt-3"**>  
 <**span** >**&nbsp;**</**span**>  
 </**div**>  
 </**td**>  
 </**tr**>  
  
 {% if messages %}  
 {% for message in messages %}  
 <**font color='GREEN'**> {{ message }}</**font**>  
 {% endfor %}  
 {% endif %}  
  
 </**table**>  
  
 </**form**></**center**>  
  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
  
{%endblock%}

User View Dataset:

{%extends 'users/userbase.html'%}  
{%load static%}  
{%block contents%}  
 <**div class="testimonial-area testimonial-padding" data-background="{%static 'img/gallery/section\_bg04.jpg'%}"**>  
 <**div class="container "**>  
 <**div class="row d-flex justify-content-center"**>  
  
 <**font color="WHITE"**>  
 <**h2**>Dataset Views</**h2**>  
 <**table class="table table-bordered bg-light text-dark" id="example" cellspacing="0" width="100%"**>  
 <**thead**>  
 <**tr**>  
 <**th**>S.No</**th**>  
 <**th**>Tweet ID</**th**>  
 <**th**>Name</**th**>  
 <**th**>Screen Name</**th**>  
 <**th**>Tweet</**th**>  
 <**th**>Created At</**th**>  
 <**th**>Location</**th**>  
 <**th**>Latitude</**th**>  
 <**th**>Longitude</**th**>  
 <**th**>User loc</**th**>  
  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 {% for i in users %}  
 <**tr style="**color: Black**"**>  
 <**td**>{{forloop.counter}}</**td**>  
 <**td**>{{i.tweetid}}</**td**>  
 <**td**>{{i.username}}</**td**>  
 <**td**>{{i.userscreenname}}</**td**>  
 <**td**>{{i.tweettext}}</**td**>  
 <**td**>{{i.createdat}}</**td**>  
 <**td**>{{i.address}}</**td**>  
 <**td**>{{i.latitude}}</**td**>  
 <**td**>{{i.longitude}}</**td**>  
 <**td**>{{i.userloc}}</**td**>  
  
 </**tr**>  
 {% endfor %}  
  
 </**tbody**>  
  
 </**table**>  
{% if users.has\_other\_pages %}  
 <**ul class="pagination"**>  
 {% if users.has\_previous %}  
 <**li**><**a href="?page={{ users.previous\_page\_number }}"**>**&laquo;**</**a**></**li**>  
 {% else %}  
 <**li class="disabled"**><**span**>**&laquo;**</**span**></**li**>  
 {% endif %}  
 {% for i in users.paginator.page\_range %}  
 {% if users.number == i %}  
 <**li class="active"**><**span**>{{ i }} <**span class="sr-only"**>(current)</**span**></**span**></**li**>  
 {% else %}  
 <**li**><**a href="?page={{ i }}"**>{{ i }}</**a**></**li**>  
 {% endif %}  
 {% endfor %}  
 {% if users.has\_next %}  
 <**li**><**a href="?page={{ users.next\_page\_number }}"**>**&raquo;**</**a**></**li**>  
 {% else %}  
 <**li class="disabled"**><**span**>**&raquo;**</**span**></**li**>  
 {% endif %}  
 </**ul**>  
{% endif %}  
  
  
  
  
  
 </**font**>  
  
  
 </**div**>  
 </**div**>  
 </**div**>  
  
{%endblock%}

Activate Registerd users

{%extends 'admins/adminbase.html'%}  
{%load static%}  
{%block contents%}  
 <**div class="testimonial-area testimonial-padding" data-background="{%static 'img/gallery/section\_bg04.jpg'%}"**>  
 <**div class="container "**>  
 <**div class="row d-flex justify-content-center"**>  
 <**div class="col-xl-10 col-lg-10 col-md-9"**>  
 <**div class="h1-testimonial-active"**>  
 *<!-- Single Testimonial -->* <**div class="single-testimonial text-center"**>  
 *<!-- Testimonial Content -->* <**h1**><**font color="WHITE"**> Activate Registered users</**font**></**h1**>  
  
 <**table border="2px" class="table table-hover table-dark"**>  
 <**thead**>  
 <**tr class="table-danger"**>  
 <**th scope="col"**>S.No</**th**>  
 <**th scope="col"**>Name</**th**>  
 <**th scope="col"**>Login ID</**th**>  
 <**th scope="col"**>Mobile</**th**>  
 <**th scope="col"**>Email</**th**>  
 <**th scope="col"**>Locality</**th**>  
 <**th scope="col"**>Status</**th**>  
 <**th scope="col"**>Activate</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 {% for i in data %}  
 <**tr scope="row" style="**color: BLUE; background-color:#FFFCBB**"**>  
 <**td**>{{forloop.counter}}</**td**>  
 <**td**>{{i.name}}</**td**>  
 <**td**>{{i.loginid}}</**td**>  
 <**td**>{{i.mobile}}</**td**>  
 <**td**>{{i.email}}</**td**>  
 <**td**>{{i.locality}}</**td**>  
 <**td**>{{i.status}}</**td**>  
  
 {% if i.status == 'waiting' %}  
 <**td**><**a class="btn-link" href="/AdminActivaUsers/?uid={{ i.id }}"  
 style="**color:DARKBLUE**"**>Activate</**a**></**td**>  
 {% else %}  
 <**td**> Activated</**td**>  
 {% endif %}  
 </**tr**>  
 {% endfor %}  
  
  
 </**tbody**>  
 </**table**>  
  
  
  
  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
  
{%endblock%}

Admin View Algorithm Results

{%extends 'admins/adminbase.html'%}  
{%load static%}  
{%block contents%}  
 <**div class="testimonial-area testimonial-padding" data-background="{%static 'img/gallery/section\_bg04.jpg'%}"**>  
 <**div class="container "**>  
 <**div class="row d-flex justify-content-center"**>  
  
  
  
  
 <**h1**><**font color="WHITE"**>Admin View Naive Bayes Results</**font**></**h1**>  
  
 <**table border="2px" class="table table-hover table-dark"**>  
 <**thead**>  
 <**tr class="table-danger"**>  
 <**th scope="col"**>S.No</**th**>  
 <**th scope="col"**>User Name</**th**>  
 <**th scope="col"**>Accuracy</**th**>  
 <**th scope="col"**>MAE</**th**>  
 <**th scope="col"**>MSE</**th**>  
 <**th scope="col"**>RMSE</**th**>  
 <**th scope="col"**>R\_Squared</**th**>  
 <**th scope="col"**>Date</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 {% for i in data %}  
 <**tr scope="row" style="**color: BLUE; background-color:#FFFCBB**"**>  
 <**td**>{{forloop.counter}}</**td**>  
 <**td**>{{i.username}}</**td**>  
 <**td**>{{i.accuracy}}</**td**>  
 <**td**>{{i.mae}}</**td**>  
 <**td**>{{i.mse}}</**td**>  
 <**td**>{{i.rmse}}</**td**>  
 <**td**>{{i.r\_squared}}</**td**>  
 <**td**>{{i.cdate}}</**td**>  
  
 </**tr**>  
 {% endfor %}  
 </**tbody**>  
 </**table**>  
 </**div**>  
 </**div**>  
 </**div**>  
  
{%endblock%}