**SORCE CODE:**

**Userside views:**

from ast import alias

from turtle import distance, speed

from django.shortcuts import render

# Create your views here.

from django.shortcuts import render, HttpResponse

from django.contrib import messages

from users.utility.process\_ml import Algorithms

import fuelconsumption

from .forms import UserRegistrationForm

from .models import UserRegistrationModel

from users.utility.process\_ml import Algorithms

algo = Algorithms()

# 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('loginid')

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/UserHomePage.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/UserHomePage.html', {})

def prediction(request):

if request.method == "POST":

import pandas as pd

from django.conf import settings

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

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

change\_in\_kineticenergy = request.POST.get("change\_in\_kineticenergy")

change\_in\_potentialenergy = request.POST.get(

"change\_in\_potentialenergy")

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

path = settings.MEDIA\_ROOT + "\\" + "fuel\_supervised\_csv.csv"

data = pd.read\_csv(path)

x = data.iloc[:, 1:]

y = data.iloc[:, 0]

x = pd.get\_dummies(x)

x = x.fillna(x.mean())

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(

x, y, test\_size=0.3, random\_state=1)

# from sklearn.preprocessing import StandardScaler

# sc = StandardScaler()

# x\_train = sc.fit\_transform(x\_train)

# x\_test = sc.fit\_transform(x\_test)

x\_train = pd.DataFrame(x\_train)

import numpy as np

from sklearn.tree import DecisionTreeRegressor

dt = DecisionTreeRegressor()

test\_set = [distance, speed, change\_in\_kineticenergy,

change\_in\_potentialenergy, weight]

print(test\_set)

#test\_set = data.drop(['Yield'], axis=1)

dt.fit(x\_train, y\_train)

print('x train:', x\_train)

print('y train:', y\_train)

#test\_set = list(np.float\_(test\_set))

y\_pred = dt.predict([test\_set])

print('y pred:', y\_pred)

return render(request, 'users/prediction.html', {'y\_pred': y\_pred})

return render(request, 'users/prediction.html')

def random\_forest(request):

mae, mse, r2 = algo.random\_forest()

print("mea:", mae)

print("mse:", mse)

print("r2:", r2)

return render(request, "users/RF.html", {'mae': mae, 'mse': mse, 'r2': r2})

def Gradiant\_Boosting(request):

mae, mse, r2 = algo.Gradiant\_Boosting()

print("mea:", mae)

print("mse:", mse)

print("r2:", r2)

return render(request, "users/GF.html", {'mae': mae, 'mse': mse, 'r2': r2})

def svm(request):

mae, mse, r2 = algo.svm()

return render(request, "users/svm.html", {'mae': mae, 'mse': mse, 'r2': r2})

def cnn(request):

history = algo.DeepLearning()

print('-'\*100)

mae = history.history['mae']

mse = history.history['mse']

print(history.history['mae'])

return render(request, "users/cnn.html", {'mae': mae[-1], 'mse': mse[-1]})

**Base.html:**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>A Machine Learning Model for Average Fuel Consumption in Heavy Vehicles</title>

<meta content="width=device-width, initial-scale=1.0" name="viewport">

<meta content="Free Website Template" name="keywords">

<meta content="Free Website Template" name="description">

<link href="{% static 'img/favicon.ico' %}" rel="icon">

<link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;400;600;700;800&display=swap" rel="stylesheet">

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" rel="stylesheet">

<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">

<link href="{% static 'lib/animate/animate.min.css' %}" rel="stylesheet">

<link href="{% static 'lib/owlcarousel/owl.carousel.min.css' %}" rel="stylesheet">

<link href="{% static 'lib/lightbox/css/lightbox.min.css' %}" rel="stylesheet">

<link href="{% static 'css/style.css' %}" rel="stylesheet">

</head>

<body>

<div class="top-bar d-none d-md-block">

<div class="container-fluid">

<div class="row">

<div class="col-md-6">

<div class="top-bar-left">

</div>

</div>

<div class="col-md-6">

<div class="top-bar-right">

</div>

</div>

</div>

</div>

</div>

<div class="navbar navbar-expand-lg bg-dark navbar-dark">

<div class="container-fluid">

<a href="index.html" class="navbar-brand"> <span style="color:Black;">Fuel Consumption </span></a>

<button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse justify-content-between" id="navbarCollapse">

<div class="navbar-nav ml-auto">

<a href="{% url 'index' %}" class="nav-item nav-link" style="color:Black;">Home</a>

<a href="{% url 'UserLogin' %}" class="nav-item nav-link" style="color:Black;">User</a>

<a href="{% url 'AdminLogin' %}" class="nav-item nav-link" style="color:Black;">Admin</a>

<a href="{% url 'UserRegister' %}" class="nav-item nav-link" style="color:Black;">Registration</a>

</div>

</div>

</div>

</div>

{%block contents%}

{%endblock%}

<div class="container copyright">

<div class="row">

<div class="col-md-6">

<p></p>

</div>

<div class="col-md-6">

</div>

<div class="col-xl-7 col-lg-7 col-md-7 co-sm-l2">

</div>

</div>

<!-- Footer End -->

<a href="#" class="back-to-top"><i class="fa fa-chevron-up"></i></a>

<!-- JavaScript Libraries -->

<script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.bundle.min.js"></script>

<script src="{% static 'lib/easing/easing.min.js' %}"></script>

<script src="{% static 'lib/owlcarousel/owl.carousel.min.js' %}"></script>

<script src="{% static 'lib/isotope/isotope.pkgd.min.js' %}"></script>

<script src="{% static 'lib/lightbox/js/lightbox.min.js' %}"></script>

<!-- Contact Javascript File -->

<script src="{% static 'mail/jqBootstrapValidation.min.js' %}"></script>

<script src="{% static 'mail/contact.js' %}"></script>

<!-- Template Javascript -->

<script src="{% static 'js/main.js' %}"></script>

</body>

</html>

**Urls.py:**

"""project8 URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

https://docs.djangoproject.com/en/2.2/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 fuelconsumption import views as mainView

from admins import views as admins

from users import views as usr

urlpatterns = [

path('admin/', admin.site.urls),

path("", mainView.index, name="index"),

path("index/", mainView.index, name="index"),

path("AdminLogin/", mainView.AdminLogin, name="AdminLogin"),

path("UserLogin/", mainView.UserLogin, name="UserLogin"),

path("UserRegister/", mainView.UserRegister, name="UserRegister"),

# Admin views

path("AdminHome/", admins.AdminHome, name="AdminHome"),

path("AdminLoginCheck/", admins.AdminLoginCheck, name="AdminLoginCheck"),

path('RegisterUsersView/', admins.RegisterUsersView, name='RegisterUsersView'),

path('ActivaUsers/', admins.ActivaUsers, name='ActivaUsers'),

# User Views

path("UserRegisterActions/", usr.UserRegisterActions, name="UserRegisterActions"),

path("UserLoginCheck/", usr.UserLoginCheck, name="UserLoginCheck"),

path("UserHome/", usr.UserHome, name="UserHome"),

path("prediction/", usr.prediction, name="prediction"),

path("random\_forest/", usr.random\_forest, name="random\_forest"),

path("Gradiant\_Boosting/",usr.Gradiant\_Boosting,name="Gradiant\_Boosting"),

path("svm/",usr.svm,name="svm"),

path("cnn", usr.cnn, name="cnn")

]