

TEAM ID: PNT2022TMID43126

DEVELOPING A FLIGHT DELAY PREDICTION MODEL USING MACHINE LEARNING

1. INTRODUCTION

1.1 Project Overview

Flight delay is inevitable and it plays an important role in both profits and loss of the airlines. Most of the proposed methods are not accurate enough because of massive volume data, dependencies and extreme number of parameters. These delays not only cause inconveniences to the airlines but also to passengers. The reasons for these delays vary a lot going from air congestion to weather conditions, mechanical problems, difficulties while boarding passengers.

1.2 Project Purpose

As people increasingly choose to travel by air, the amount of flights that fails to take off on time increases. This growth exacerbates the crowded situation at airports and causes financial difficulties within the airline industry. Therefore, predicting flight delays can improve airline operations and passengers satisfaction, which will result in a positive impact on the economy.

2. LITERATURE SURVEY

2.1 Existing problem

Nowadays, the demand for airline transportation is increasing significantly. Analysis of flight delay, therefore, has become a popular research area. Various researchers used different techniques of machine learning and data mining to conduct the investigation. They were interested in different aspects such as airport faculty location, weather condition, and airport capacity.

2.2 References

1. Maryam Farshchian Yazdi, Seyed Reza Kamel & Maryam Kheirabadi.2020;

<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00380-z>

2. Yuemin Tang

<https://dl.acm.org/doi/fullHtml/10.1145/3497701.3497725>

3. Javier Herbas

<https://medium.com/analytics-vidhya/using-machine-learning-to-predict-flight-delays-e8a50b0bb64c>

4. Jorge de Antonio Del Pecho, Fran Jose Diego Acosta & Anthony Roux

<https://developers.amadeus.com/blog/flight-delay-prediction-machine-learning>

5. Bhuvan Bhatia

<https://scholarworks.calstate.edu/downloads/qr46r081g>

2.3 Problem Statement Definition

Most common problem that is experienced by every airline passenger is flight delay. The flight delay may occur due to the major three reasons. The foremost reason is the abnormal changes in weather, the other reasons include technical problems and successive landing of flights. In this project we are gonna develop a machine learning model to predict the flight delay beforehand. The website is developed to notify the flight delay and show the live weather condition in required area.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

**Project Design Phase-I
Proposed Solution**

Date	10 October 2022
Team ID	PNT2022TMID43126
Project Name	Developing a flight delay prediction model using machine learning
Maximum Marks	2 Marks

Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none">• Predicting the flight delay accurately• Creating high expensive websites with user friendly interface
2.	Idea / Solution description	<ul style="list-style-type: none">• The data exploration, data testing and training is done using high resulting predictive model algorithms• Simple and easily accessible interface in website
3.	Novelty / Uniqueness	<ul style="list-style-type: none">• Website with live weather updation on specified place
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none">• Predicting and updating the delay before hand helps users to be unhurried and relaxed during their journey
5.	Business Model (Revenue Model)	<ul style="list-style-type: none">• The website with user friendly interface and design .• Live graph and plots of weather and time delay
6.	Scalability of the Solution	<ul style="list-style-type: none">• The website design can be changed according to user's feedback• 100 percentage scalability

3.4 Problem Solution Fit

Project Design Phase-I Problem Solution Fit

Date	10 October 2022
Team ID	PNT2022TMID43126
Project Name	Developing a flight delay prediction model using machine learning
Maximum Marks	4 Marks

Problem solution fit:

1. CUSTOMER SEGMENT(S) <small>Define CL, BL into CL</small> CS <ul style="list-style-type: none"> ✦ Travelers ✦ Business man ✦ Friends/Families of passenger 	6. CUSTOMER LIMITATIONS <small>CL, SUBJECT, SERVICES</small> CL <ul style="list-style-type: none"> ✦ Mobile networks can't be accessed inside aircraft ✦ Some website doesn't give accurate information which makes the customer to lose trust in online websites 	5. AVAILABLE SOLUTIONS <small>PROB & CONS</small> AS <ul style="list-style-type: none"> ✦ Predicting flight delay ✦ Creating high responsive website ✦ Use friendly website
2. PROBLEMS / PAINS <small>+ CLS frequency</small> PR <ul style="list-style-type: none"> ✦ Listening to flight updation taken on and take off details. ✦ Rescheduling their work time based on the delay time. 	3. PROBLEM ROOT / CAUSE <small>PRC</small> <ul style="list-style-type: none"> ✦ Due to weather condition, landing problems occurs. ✦ The delay of previous flight to take off also result in other success flight delay. ✦ Traffic in airways. 	7. BEHAVIOR <small>+ CLS intensity</small> BS <ul style="list-style-type: none"> ✦ Flight updation by pilot or flight attendees. ✦ Checking out on other websites.
3. TRIGGERS TO ACT <small>TR</small> <ul style="list-style-type: none"> ✦ Checking out flight travel time and weather conditions before take on. 	10. YOUR SOLUTION <small>SL</small> <ul style="list-style-type: none"> ✦ Predicting the accurate flight delay time using good predictive models and data classifier algorithms. ✦ Creating a highly responsive website which is user friendly and can be accessed by everyone. 	8. CHANNELS of BEHAVIOR <small>CH</small> CH <ul style="list-style-type: none"> ✦ Online : checking out weather conditions of destination places. ✦ Offline : listening to flight updates at airport and inside aircraft.
4. EMOTIONS <small>BEFORE / AFTER</small> EM <ul style="list-style-type: none"> ✦ Before : confused about travel time, depressed, about landing time, stressed, annoyed. ✦ After : Feel in ease, relaxed. 		

4. REQUIRMENT ANALYSIS

4.1 Functional Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Visiting website	Visiting website for weather updates Visiting website for flight departure and arrival timings
FR-2	Login	Login using your name Create a password
FR-3	Searching	Searching for specified flight times
FR-4	Notifying	Notifying users incase of flight delay

4.2 Non-Functional Requirement

Non-functional Requirements:

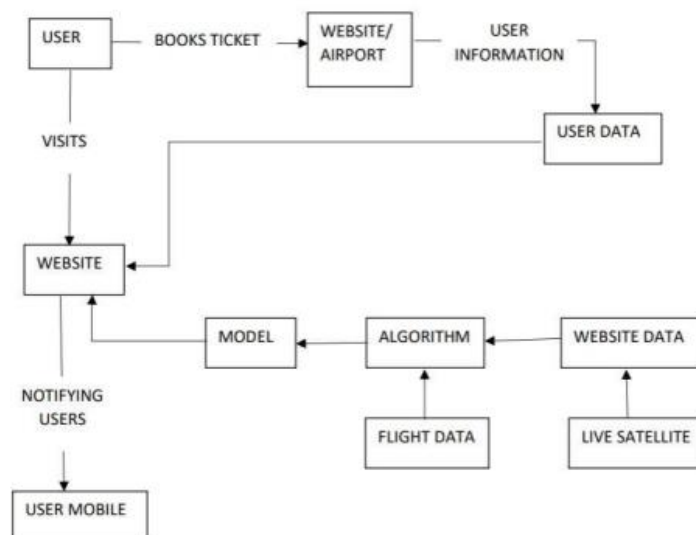
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The UI should be simple enough for everyone to understand
NFR-2	Security	The website must be secure enough to trust by the users.
NFR-3	Reliability	The UI should be able to withstand any errors in the data.
NFR-4	Performance	The live plot for the weather conditions is presented in the dashboard.
NFR-5	Availability	The UI should respond to the users within 2 seconds
NFR-6	Scalability	Different flight datas can be captured and shown.

5. PROJECT DESIGN

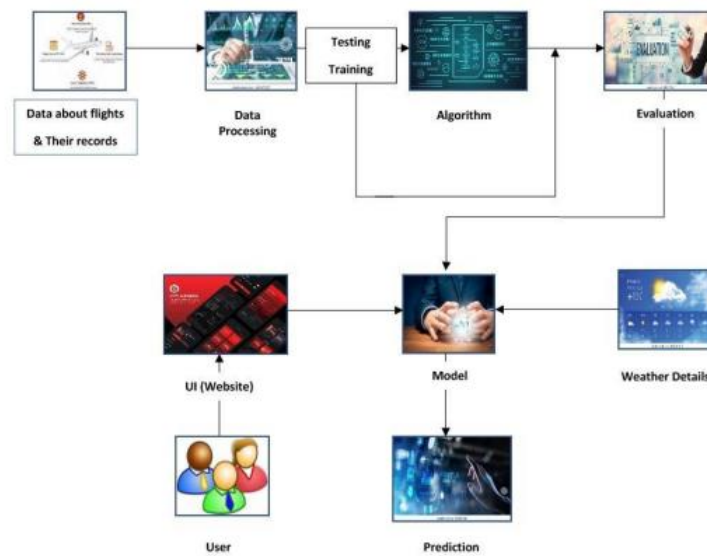
5.1 Data Flow Diagrams

Data Flow Diagrams:



5.2 Solution & Technical Architecture

Solution Architecture:



5.3 User Stories

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Travellers	Booking tickets	USN-1	As a traveller, I can book tickets.	I can use my ticket for traveling.	High	Sprint-1
	Login	USN-2	As a traveller, I can login to the website	I can register for the website	High	Sprint-1
	Visiting website	USN-3	As a traveller I can check for weather updates and flight delay time	I can access the website	High	Sprint-2
Airport manager	Visiting website	USN-4	As a user, I can log into the application by entering email & password	I can access the website	High	Sprint-1
		USN-5	As a user, I can check for weather updates and flight delay time	I can access the website	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

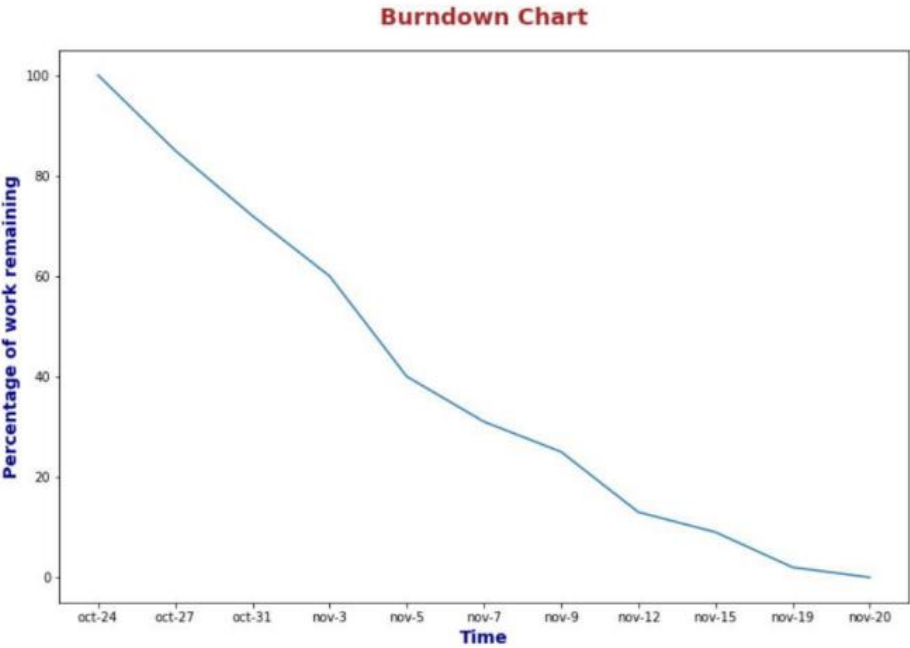
Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dashboard	USN-1	As a user, I can select the airport city to view the weather	2	High	Selvabhuvaneswari Raj Kumar
Sprint-1	Dashboard	USN-2	As a user, I can select the flight by entering flight number to view flight usual timings	1	High	Sri Brintha Sabareeswaran
Sprint-2	Registration	USN-3	As a user, I can register in the website to get notified in case of flight delay	2	Low	Lavanya Sri Brintha
Sprint-3	Dashboard	USN-4	As a user, I can check for flight delay	2	Medium	Selvabhuvaneswari Sabareeswaran
Sprint-4	Notification	USN-5	As a user, I can get notified in case of flight delay	1	High	Raj Kumar Lavanya

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

6.3 Report From JIRA



7. CODING & SOLUTIONING (Explain the features Added in the project along with code)

7.1 Feature 1

Weather.css

```
*{  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
  font-family: "Poppins" , sans-serif;  
}  
  
body{  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  min-height: 100vh;  
  background- image:url (weather.jpg);  
  background-repeat: no-repeat;  
}  
  
.wrapper {  
  width: 400px;  
  background: #fff;  
  border-radius: 7px;  
}
```

```
.wrapper header {  
    color: #8d8b8b;  
    font-size: 21px;  
    font-weight: 500;  
    padding: 16px 15px;  
    display: flex;  
    align-items: center;  
    /* justify-content: center; */  
    border-bottom: 1px solid #bfbfbf;  
}
```

```
header i{ /*arrow icon*/  
    cursor: pointer;  
    font-size: 0em;  
    margin-right: 8px;  
}
```

```
.wrapper.active header i{  
    font-size: 30px;  
    margin-left: 5px;  
}
```

```
.wrapper .input-part {  
    margin: 20px 25px 30px; /*top left bottom*/  
}
```

```
.wrapper.active .input-part{  
display: none;  
}
```

```
.input-part .info-txt {  
    display: none;  
    font-size: 17px;  
    text-align: center;  
    padding: 12px 10px;  
    border-radius: 7px;  
    margin-bottom: 15px;  
}
```

```
.info-txt.error{  
    display: block;  
    color: #721c24;  
    background: #f8d7da ;  
    border: 1px solid #f5c6cb;  
}
```

```
.info-txt.pending{  
    display: block;  
    color: #0c5460;  
    background: #d1ecf1;  
    border: 1px solid #bee5eb;
```

```
}
```

```
.input-part :where(input,button) {  
width: 100%;  
height: 55px;  
border: none;  
outline: none;  
font-size: 18px;  
border-radius: 7px;  
}
```

```
.input-part input {  
width: 75% !important;  
text-align: center;  
border: 1px solid #bfbfbf;  
}
```

```
.input-part input:is(:focus, :valid ){  
border: 1px solid #8d8b8b;  
}
```

```
.input-part input::placeholder{  
color:#bfbfbf;  
}
```

```
.searchArea{  
display: flex;  
justify-content: space-between;  
align-items: center;  
cursor: pointer;  
}
```

```
.searchIcon{  
border:1px solid #bfbfbf;  
border-radius: 10px;  
background-color:#8d8b8b;  
color: white;  
width: 25%;  
height: 55px;  
outline: none;  
font-size: 20px;  
display: flex;  
justify-content: center;  
align-items: center;  
text-align: center;  
padding-top: 14px;  
margin-left: 8px;  
transition: 0.3s ease;  
}
```

```
.searchIcon:hover {
```

```
    background: #6b6a6a;
}
```

```
.input-part .separator {
    height: 1px;
    width: 100%;
    margin: 25px 0;
    background-color: #ccc;
    display: flex;
    align-items: center;
    justify-content: center;
}
```

```
.separator::before{
    content: "or";
    color: #ccc;
    padding: 0 15px;
    font-size: 19px;
    background-color: #fff;
}
```

```
.input-part button {
    color: #fff;
    cursor: pointer;
    background-color: #8d8b8b;
    transition: 0.3s ease;
}
```



```
.input-part button:hover{  
    background: #6b6a6a;  
}
```

```
.wrapper .weather-part{  
    display:none;  
    align-items: center;  
    justify-content: center;  
    flex-direction: column;  
}
```

```
.wrapper.active .weather-part{  
    display: flex;  
}
```

```
.weather-part img {  
    width: 170px;  
}
```

```
.weather-part .temp {  
    display: flex;  
    font-size: 70px;  
}
```

```
.weather-part .temp .numb{  
font-weight: 100;  
}
```

```
.weather-part .temp .deg {  
    font-size: 40px;  
    margin: 10px 5px 0 0 ;  
    display: block;  
}
```

```
.weather-part .weather{  
    font-size: 17px;  
    text-align: center;  
    margin: -5px 20px 15px;  
}
```

```
.weather-part .location{  
    display: flex;  
    font-size: 19px;  
    margin-bottom: 30px;  
    text-align: center;  
    align-items: flex-start;  
    padding: 0 20px;  
  
}
```

```
.location i{  
font-size: 21px;  
margin: 0 5px 0 0 ;  
}
```

```
.weather-part .bottom-details{  
width: 100%;  
display: flex;  
justify-content: space-between;  
border-top: 1px solid #bfbfbf;  
}
```

```
.bottom-details .column {  
width: 100%;  
display: flex;  
padding: 15px 0;  
align-items: center;  
justify-content: center;  
}
```

```
.column i{  
color: #8d8b8b;  
font-size: 40px;  
}
```

```
.column.humidity {
```

```
border-left:1px solid #bfbfbf;  
}
```

```
.column .details{  
    margin-left: 3px;  
}
```

```
.details .temp, .humidity span {  
    font-size: 18px;  
    font-weight: 500;  
    margin-top: -3px;  
}
```

```
.details .temp .deg{  
    margin:0;  
    font-size: 17px;  
    padding: 0 2px 0 1px;  
}
```

```
.column .details p{  
    font-size: 14px;  
    margin-top: -6px;  
}
```

```
.humidity i{  
    font-size: 37px;
```

```
}
```

```
@media screen and (max-width: 770px){
```

```
body {
```

```
    display: flex;
```

```
    justify-content: center;
```

```
    align-items: center;
```

```
}
```

```
.wrapper {
```

```
    max-width: 400px;
```

```
    max-height: 500px;
```

```
    font-size: 20px;
```

```
    margin: 0 20px 0 20px;
```

```
}
```

```
.wrapper header {
```

```
    font-weight: 400;
```

```
}
```

```
img {
```

```
    max-width: 170px;
```

```
}
```

```
};
```

Weather.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

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

```
<!-- CSS -->
```

```
<link rel="stylesheet" href="weather.css" />
```

```
<!-- Icon -->
```

```
<link
```

```
href="https://unpkg.com/boxicons@2.1.1/css/boxicons.min.css"
```

```
rel="stylesheet"
```

```
/>
```

```
<title>Weather</title>
```

```
</head>
```

```
<body>
```

```
<div class="wrapper">
```

```
<header><i class="bx bx-left-arrow-alt"></i>Weather </header>
```

```
<!-- Middle area (input ,search, button)-->
```

```
<section class="input-part">
```

```
<p class="info-txt"></p>
<div class="searchArea">
  <input
    type="text"
    placeholder="Enter city name"
    spellcheck="false"
    required
  />

  <i class="bx bx-search-alt searchIcon"></i>
</div>
<div class="separator"></div>
<button>Get Device Location</button>
</section>
<!-- Middle area End -->

<!-- Weather active area start -->
<section class="weather-part">
  
  <div class="temp">
    <span class="numb">_</span>
    <span class="deg">&#176C</span>
  </div>

  <div class="weather">_</div>
  <div class="location">
```

<i class="bx bx-map"></i>

_

</div>

<div class="bottom-details">

<div class="column feels">

<i class="bx bxs-thermometer"></i>

<div class="details">

<div class="temp">

_

°C

</div>

<p>Feels like</p>

</div>

</div>

<div class="column humidity">

<i class="bx bxs-droplet-half"></i>

<div class="details">

_

<p>Humidity</p>

</div>

</div>

</div>

</section>

<!--Weather active area end -->


```
</div> <!-- Wrapper div end -->
```

```
</div>
```

```
<script src="weather.js"></script>
```

```
</body>
```

```
</html>
```

Weather.js

```
const wrapper = document.querySelector(".wrapper"); //container div
```

```
inputPart = wrapper.querySelector(".input-part"); // input section
```

```
infoTxt = inputPart.querySelector(".info-txt"); // please enter... kismi
```

```
inputField = inputPart.querySelector("input"); // input alani
```

```
locationBtn = inputPart.querySelector("button");
```

```
wlcon = document.querySelector("img");
```

```
arrowBack = wrapper.querySelector("header i");
```

```
searchIcon = document.querySelector(".searchIcon");
```

```
let api;
```

```
inputField.onkeydown = function(e){
```

```
    if(e.keyCode == 13 && inputField.value != ""){
```

```
        requestApi(inputField.value);
```

```
    }
```

```
};
```

```
searchIcon.addEventListener("click", e => {  
  if (inputField.value === "" || inputField.value === " "){  
    alert("Please enter data in this field")  
  } else {  
    requestApi(inputField.value);  
  }  
});
```

```
locationBtn.addEventListener("click", () => {  
  //eger tarayici lokasyonu destekliyorsa  
  if(navigator.geolocation){  
    navigator.geolocation.getCurrentPosition(onSuccess, onError);  
  }else{  
    alert("Your browser not support geolocation api");  
  }  
});
```

```
function requestApi(city){  
  //api alip json object dönüştürme  
  api =  
  `https://api.openweathermap.org/data/2.5/weather?q=${city}&units  
  =metric&appid=8641075dda9ea5d5c961c48c00929bec`;   
  fetchData();  
};
```

```
function onSuccess(position){  
    //enlem boylam. Coords objesinden getiriyoruz burada.  
    const {latitude, longitude} = position.coords;  
    api =  
    `https://api.openweathermap.org/data/2.5/weather?lat=${latitude}  
    &lon=${longitude}&units=metric&appid=8641075dda9ea5d5c961c4  
    8c00929bec`;   
    fetchData();  
};
```

```
function onError(error){  
    infoTxt.innerText = error.message; //tarayicinin verecegi error  
    ekrana basar.  
    infoTxt.classList.add("error");  
};
```

```
function fetchData(){  
    infoTxt.innerText = "Getting weather details...";  
    infoTxt.classList.add("pending");  
    fetch(api).then(response => response.json()).then(result =>  
weatherDetails(result));  
    //then func. api sonucu ile weatherDetails çağırır.  
};
```

```
function weatherDetails(info){
  if(info.cod == "404"){
    infoTxt.classList.replace("pending", "error");
    infoTxt.innerText = `${inputField.value} isn't a valid city name`;
  }else {
    //console'dan alinan bilgiler
    const city = info.name;
    const country = info.sys.country;
    const {description, id} = info.weather[0];
    const {feels_like, humidity, temp} = info.main;

    if(id == 800){
      wlcon.src="clear.svg"
    } else if (id >= 200 && id <=232){
      wlcon.src="strom.svg"
    } else if (id >= 600 && id <= 622){
      wlcon.src="snowy.svg"
    } else if (id >= 701 && id <=781){
      wlcon.src="haze.svg"
    } else if (id >= 801 && id <=804){
      wlcon.src="cloudy.svg"
    } else if ((id >= 300 && id <= 321) || (id >= 500 && id <= 531)){
      wlcon.src="rainy.svg"
    }
  }
}
```

```

//html'e basmamiz için
wrapper.querySelector(".temp .numb").innerText =
Math.floor(temp);

wrapper.querySelector(".weather").innerText =
description.toUpperCase();

wrapper.querySelector(".location span").innerText = `
${city},${country}`;

wrapper.querySelector(".temp .numb-2").innerText =
Math.floor(feels_like);

wrapper.querySelector(".humidity
span").innerText=`${humidity}%`;


infoTxt.classList.remove("pending", "error");
infoTxt.innerText = "";
inputField.value = "";
wrapper.classList.add("active"); // hava durumunu gösterir.
}
};

arrowBack.addEventListener("click", () => {
    wrapper.classList.remove("active");
});

```

7.2 Feature 2

Flight Timings.css

```
.section {  
    position: relative;  
    height: 100vh;  
    background:transparent;  
}
```

```
.section .section-center {  
    position: absolute;  
    top: 50%;  
    left: 0;  
    right: 0;  
    -webkit-transform: translateY(-50%);  
    transform: translateY(-50%);  
}
```

```
.booking {  
    font-family:sans-serif;  
    background-image: url('sunset.jpg');  
    background-size: cover;  
    background-position: center;  
    color: #191a1e;  
}
```

```
.booking-form {  
    position: center;  
    background:transparent;
```

```
max-width:700px;
width: 100%;
margin: auto;
padding: 45px 25px 25px;
border-radius: 4px;
-webkit-box-shadow: 0px 0px 10px -5px rgba(0, 0, 0, 0.4);
box-shadow: 0px 0px 10px -5px rgba(0, 0, 0, 0.4);
}
```

```
.booking-form .form-group {
    position: relative;
    height:70px;
    margin-bottom:25px;
}
```

```
.booking-form .form-control {
    Background-color: #fff;
    height: 55px;
    padding: 0px 15px;
    padding-top: 24px;
    color: #000000;
    border: 2px solid #A9A9A9;
    font-size: 16px;
    font-weight: 700;
    -webkit-box-shadow: none;
```

```
    box-shadow: none;
    border-radius: 4px;
    -webkit-transition: 0.2s all;
    transition: 0.2s all;
}

.booking-form .form-control::-webkit-input-placeholder {
    color: #A9A9A9;
}

.booking-form .form-control:-ms-input-placeholder {
    color: #A9A9A9;
}

.booking-form .form-control::placeholder {
    color: #A9A9A9;
}

.booking-form .form-control:focus {
    background: #f9fafb;
}

.booking-form input[type="date"].form-control:invalid {
    color: #dfe5e9;
}
```



```
.booking-form select.form-control {  
    -webkit-appearance: none;  
    -moz-appearance: none;  
    appearance: none;  
}
```

```
.booking-form select.form-control+.select-arrow {  
    position: absolute;  
    right: 6px;  
    bottom: 6px;  
    width: 32px;  
    line-height: 32px;  
    height: 32px;  
    text-align: center;  
    pointer-events: none;  
    color: #dfe5e9;  
    font-size: 14px;  
}
```

```
.booking-form select.form-control+.select-arrow:after {  
    content: '\279C';  
    display: block;  
    -webkit-transform: rotate(90deg);  
    transform: rotate(90deg);  
}
```

```
.booking-form .form-label {  
    position: absolute;  
    top: 6px;  
    left: 20px;  
    font-weight: 700;  
    text-transform: uppercase;  
    line-height: 24px;  
    height: 24px;  
    font-size: 12px;  
    color: #4B0082;  
}
```

```
.booking-form .form-checkbox input {  
    position: absolute !important;  
    margin-left: -9999px !important;  
    visibility: hidden !important;  
}
```

```
.booking-form .form-checkbox label {  
    position: relative;  
    padding-top: 4px;  
    padding-left: 30px;  
    font-weight: 700;  
    color: #191a1e;  
}
```

```
.booking-form .form-checkbox label+label {  
    margin-left: 15px;  
}
```

```
.booking-form .form-checkbox input+span {  
    position: absolute;  
    left: 2px;  
    top: 4px;  
    width: 20px;  
    height: 20px;  
    background: #fff;  
    border: 2px solid #dfe5e9;  
    border-radius: 50%;  
}
```

```
.booking-form .form-checkbox input+span:after {  
    content: "";  
    position: absolute;  
    padding: -15px;  
    top: 50%;  
    left: 50%;  
    width: 0px;  
    height: 0px;  
    border-radius: 50%;  
    background-color: #4fa3e3;  
    -webkit-transform: translate(-50%, -50%);
```

```
transform: translate(-50%, -50%);
-webkit-transition: 0.2s all;
transition: 0.2s all;
}

.booking-form .form-checkbox input:not(:checked)+span:after {
    opacity: 0;
}

.booking-form .form-checkbox input:checked+span:after {
    opacity: 1;
    width: 10px;
    height: 10px;
}

.booking-form .submit-btn {
    color: #fff;
    background-color: #4fa3e3;
    font-weight: 400;
    height: 65px;
    font-size: 18px;
    border: none;
    width: 100%;
    border-radius: 4px;
    text-transform: uppercase
}
```

```
.booking-cta {  
    margin-top: 45px;  
}
```

```
.booking-cta h1 {  
    font-size: 52px;  
    text-transform: uppercase;  
    color: #4fa3e3;  
    font-weight: 400;  
}
```

```
.booking-cta p {  
    font-size: 22px;  
    color: #191a1e;  
}
```

Flight Timings.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="utf-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
    <!-- The above 3 meta tags *must* come first in the head; any other head  
content must come *after* these tags -->
```

```
<title>Flight Timings</title>
```

```
<!-- Google font -->

<link href="https://fonts.googleapis.com/css?family=Lato:400,700"
rel="stylesheet">

<!-- Bootstrap -->

<link type="text/css" rel="stylesheet" href="{{ url_for('static',
filename='css/bootstrap.min.css')}}" />

<!-- Custom stylesheet -->

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

<style>

    #booking
    {
font-family: 'Lato', sans-serif;

        background: url(sunset.jpg);

        background-size: cover;

        background-position: center;

        color: #191a1e;

    }

</style>

<!---->

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and
media queries -->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script
src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

        <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
```

```
<![endif]-->
```

```
</head>
```

```
<body>
```

```
<div id="booking" class="section">
```

```
<div class="section-center">
```

```
<div class="container">
```

```
<div class="row">
```

```
<div class="col-md-4">
```

```
<div class="booking-cta">
```

```
<h1>Flight Delay Prediction</h1>
```

```
<p></p>
```

```
</div>
```

```
</div>
```

```
<div class="col-md-7 col-md-offset-1">
```

```
<div class="booking-form">
```

```
<form action={{ url_for("predict") }} method="post">
```

```
<div class="row">
```

```
<div class="col-md-4">
```

```
<div class="form-group">
```

```
<span class="form-label">Year</span>
```

```
<input type="text" class="form-control"
name="year" placeholder="Enter year" required="true">
```

```
<span class="select-arrow"></span>
</div>
</div>
<!Year>
```

```
<div class="col-md-4">
  <div class="form-group">
    <span class="form-label">Month</span>
    <input type="text" class="form-control"
name="month" placeholder="Enter month" required="true">
    <span class="select-arrow"></span>
  </div>
</div>
<!Month>
```

```
<div class="col-md-4">
  <div class="form-group">
    <span class="form-label">Date</span>
    <input type="text" class="form-control"
name="day" placeholder="Enter date" required="true">
    <span class="select-arrow"></span>
  </div>
</div>
<! Date>
```

```
</div>
<!--Year,Month,Date end-->
```



```

<div class="row">

    <div class="col-md-6">
        <div class="form-group">
            <span class="form-label">Select an Airline</span>
            <!-- <input class="form-control" type="date"
required> -->

            <select class="form-control" name="carrier">
                <option value="UA">United Air Lines
Inc.(UA)</option>

                <option value="AA">American Airlines
Inc.(AA)</option>

                <option value="US">US Airways Inc.(US)</option>
                <option value="F9">Frontier Airlines
Inc.(F9)</option>

                <option value="B6">JetBlue Airways(B6)</option>
                <option value="OO">Skywest Airlines
Inc.(OO)</option>

                <option value="AS">Alaska Airlines
Inc.(AS)</option>

                <option value="WN">Southwest Airlines
Co.(WN)</option>

                <option value="DL">Delta Air Lines
Inc.(DL)</option>

                <option value="EV">Atlantic Southeast
Airlines(EV)</option>

                <option value="HA">Hawaiian Airlines
Inc.(HA)</option>

```

```

Inc.(MQ)</option>
    <option value="MQ">American Eagle Airlines
    <option value="VX">Virgin America(VX)</option>
    <option value="9E">Endeavor Air(9E)</option>
    <option value="FL">AirTran Airways(FL)</option>
    <option value="YV">Mesa Airlines(YV)</option>
</select>
</div>
</div>
<!Airline>

</div>
<!--Airline end-->

<div class="row">

    <div class="col-md-6">
        <div class="form-group">
            <span class="form-label">Flying from</span>
            <!--<input class="form-control" type="text"
placeholder="City or airport">-->
            <select class="form-control" name="origin">
                <option value="EWR">Newark Liberty International
Airport(EWR)</option>
                <option value="JFK">John F. Kennedy International
Airport(New York International Airport)(JFK)</option>
                <option value="LGA">LaGuardia Airport(Marine Air
Terminal)(LGA)</option>

```

```
</select>
</div>
</div>
<!Flying from>
```

```
<div class="col-md-6">
  <div class="form-group">
    <span class="form-label">Flying to</span>
    <!-- <input class="form-control" type="text"
placeholder="City or airport"> -->
    <select class="form-control" name="dest">
      <option value="ATL">Hartsfield-Jackson Atlanta
International Airport(ATL)</option>
      <option value="ORD">Chicago O'Hare
International Airport(ORD)</option>
      <option value="LAX">Los Angeles International
Airport(LAX)</option>
      <option value="BOS">Gen. Edward Lawrence
Logan International Airport(BOS)</option>
      <option value="MCO">Orlando International
Airport(MCO)</option>
      <option value="CLT">Charlotte Douglas
International Airport(CLT)</option>
      <option value="SFO">San Francisco International
Airport(SFO)</option>
      <option value="FLL">Fort Lauderdale-Hollywood
International Airport(FLL)</option>
      <option value="MIA">Miami International
Airport(MIA)</option>
```

`<option value="DCA">Ronald Reagan Washington
National Airport(DCA)</option>`

`<option value="DTW">Detroit Metropolitan
Airport(DTW)</option>`

`<option value="DFW">Dallas/Fort Worth
International Airport(DFW)</option>`

`<option value="RDU">Raleigh-Durham
International Airport(RDU)</option>`

`<option value="TPA">Tampa International
Airport(TPA)</option>`

`<option value="DEN">Denver International
Airport(DEN)</option>`

`<option value="IAH">George Bush
Intercontinental Airport(IAH)</option>`

`<option value="MSP">Minneapolis-Saint Paul
International Airport(MSP)</option>`

`<option value="PBI">Palm Beach International
Airport(PBI)</option>`

`<option value="BNA">Nashville International
Airport(BNA)</option>`

`<option value="LAS">McCarran International
Airport(LAS)</option>`

`<option value="SJU">Luis Mu±oz Mar±n
International Airport(SJU)</option>`

`<option value="IAD">Washington Dulles
International Airport(IAD)</option>`

`<option value="PHX">Phoenix Sky Harbor
International Airport(PHX)</option>`

`<option value="BUF">Buffalo Niagara
International Airport(BUF)</option>`

`<option value="CLE">Cleveland Hopkins
International Airport(CLE)</option>`

`<option value="STL">St. Louis International
Airport at Lambert Field(STL)</option>`

`<option value="MDW">Chicago Midway
International Airport(MDW)</option>`

`<option value="SEA">Seattle-Tacoma
International Airport(SEA)</option>`

`<option value="CVG">Cincinnati/Northern
Kentucky International Airport(CVG)</option>`

`<option value="MSY">Louis Armstrong New
Orleans International Airport(MSY)</option>`

`<option value="RSW">Southwest Florida
International Airport(RSW)</option>`

`<option value="CMH">Port Columbus
International Airport(CMH)</option>`

`<option value="CHS">Charleston International
Airport/Charleston AFB(CHS)</option>`

`<option value="PIT">Pittsburgh International
Airport(PIT)</option>`

`<option value="SAN">San Diego International
Airport (Lindbergh Field)(SAN)</option>`

`<option value="MKE">General Mitchell
International Airport(MKE)</option>`

`<option value="JAX">Jacksonville International
Airport(JAX)</option>`

`<option value="BTV">Burlington International
Airport(BTV)</option>`

`<option value="SLC">Salt Lake City International
Airport(SLC)</option>`

`<option value="AUS">Austin-Bergstrom
International Airport(AUS)</option>`

`<option value="ROC">Greater Rochester
International Airport(ROC)</option>`

`<option value="RIC">Richmond International
Airport(RIC)</option>`

`<option value="PWM">Portland International
Jetport(PWM)</option>`

`<option value="HOU">William P. Hobby
Airport(HOU)</option>`

`<option value="IND">Indianapolis International
Airport(IND)</option>`

`<option value="MCI">Kansas City International
Airport(MCI)</option>`

`<option value="SYR">Syracuse Hancock
International Airport(SYR)</option>`

`<option value="BWI">Baltimore-Washington
International Airport(BWI)</option>`

`<option value="MEM">Memphis International
Airport(MEM)</option>`

`<option value="PHL">Philadelphia International
Airport(PHL)</option>`

`<option value="GSO">Piedmont Triad
International Airport(GSO)</option>`

`<option value="ORF">Norfolk International
Airport(ORF)</option>`

`<option value="DAY">James M. Cox Dayton
International Airport(DAY)</option>`

`<option value="PDX">Portland International
Airport(PDX)</option>`

`<option value="SRQ">Sarasota-Bradenton
International Airport(SRQ)</option>`

`<option value="SDF">Louisville International
Airport (Standiford Field)(SDF)</option>`

`<option value="XNA">Northwest Arkansas
Regional Airport(XNA)</option>`

`<option value="MHT">Manchester-Boston
Regional Airport(MHT)</option>`

`<option value="BQN">Rafael Hern  ndez
Airport(BQN)</option>`

`<option value="CAK">Akron-Canton Regional
Airport(CAK)</option>`

`<option value="OMA">Eppley
Airfield(OMA)</option>`

`<option value="SNA">John Wayne Airport (  
Orange County Airport)(SNA)</option>`

`<option value="GSP">Greenville-Spartanburg
International Airport(GSP)</option>`

`<option value="SAV">Savannah/Hilton Head
International Airport(SAV)</option>`

`<option value="GRR">Gerald R. Ford International
Airport(GRR)</option>`

`<option value="HNL">Honolulu International
Airport(HNL)</option>`

`<option value="LGB">Long Beach Airport (  
Daugherty Field)(LGB)</option>`

`<option value="SAT">San Antonio International
Airport(SAT)</option>`

`<option value="TYS">McGhee Tyson
Airport(TYS)</option>`

Airport(MSN)</option>	<option value="MSN">Dane County Regional
Airport(DSM)</option>	<option value="DSM">Des Moines International
Airport(STT)</option>	<option value="STT">Cyril E. King
Airport(ALB)</option>	<option value="ALB">Albany International
Airport(BDL)</option>	<option value="BDL">Bradley International
(Hollywood Burbank Airport)(BUR)</option>	<option value="BUR">Bob Hope Airport
State Airport(PVD)</option>	<option value="PVD">Theodore Francis Green
Airport(BGR)</option>	<option value="BGR">Bangor International
Airport(PSE)</option>	<option value="PSE">Mercedita
José Martí International Airport(SJC)</option>	<option value="SJC">Norman Y. Mineta San
Airport(OKC)</option>	<option value="OKC">Will Rogers World
Airport(OAK)</option>	<option value="AOK">Oakland International
Airport(TUL)</option>	<option value="TUL">Tulsa International
Airport(SMF)</option>	<option value="SMF">Sacramento International
International Airport(BHM)</option>	<option value="BMH">Birmingham-Shuttlesworth

Airport(ACK)</option>	<option value="ACK">Nantucket Memorial
Airport(AVL)</option>	<option value="AVL">Asheville Regional
Sunport(ABQ)</option>	<option value="ABQ">Albuquerque International
Airport(MVY)</option>	<option value="MVY">Martha's Vineyard
	<option value="EGE">Eagle County Regional Airport(EGE)</option>
	<option value="CRW">YeagerAirport(CRW)</option>
Airport(ILM)</option>	<option value="ILM">Wilmington International
Airport(CAE)</option>	<option value="CAE">Columbia Metropolitan
Airport(TVC)</option>	<option value="TVC">Cherry Capital
Airport(MYR)</option>	<option value="MYR">Myrtle Beach International
Airport(CHO)</option>	<option value="CHO">Charlottesville-Albemarle
International AirportÂ (Gallatin Field Airport)(BZN)</option>	<option value="BZN">Bozeman Yellowstone
Airport(JAC)</option>	<option value="JAC">Jackson Hole
Airport(PSP)</option>	<option value="PSP">Palm Springs International
Airport(EYW)</option>	<option value="EYW">Key West International
(Yampa Valley Regional)(HDN)</option>	<option value="HDN">Yampa Valley AirportÂ

```

        <option value="MTJ">Montrose Regional
Airport(MTJ)</option>

        <option value="SBN">South Bend International
AirportÂ (South Bend Regional)(SBN)</option>

        <option value="ANC">Ted Stevens Anchorage
International Airport(ANC)</option>

        <option value="LEX">Blue Grass
Airport(LEX)</option>

    </select>

</div>

</div>

<!--Flying to,from end-->

<div class="form-btn">

    <button class="submit-btn">Predict</button>

</div>

<!--Button>

</form>

<!--Form end-->

</div>

<!--Booking form>

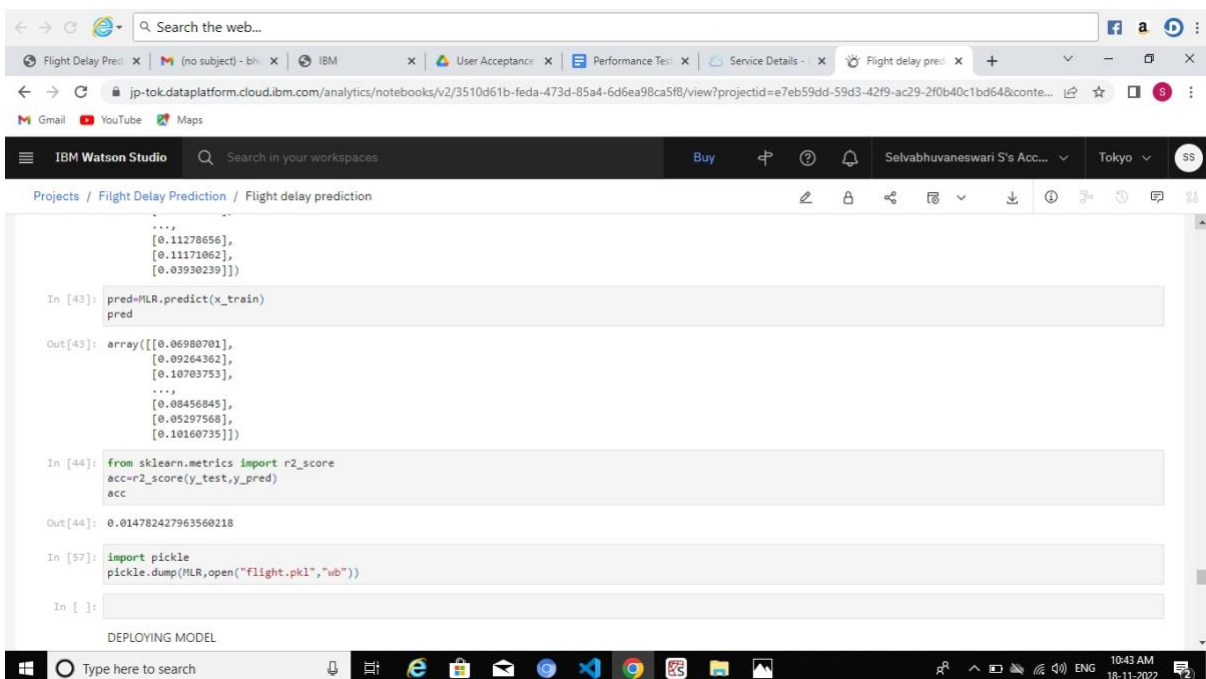
</div>
```

</div>
</div>
</div>
</div>
</body>

</html>

8. Testing

8.1 Test cases:



The screenshot displays the IBM Watson Studio web interface. The browser address bar shows the URL: `jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/3510d61b-feda-473d-85a4-6d6ea98ca5f8/view?projectid=e7eb59dd-59d3-42f9-ac29-2f0b40c1bd64&conte...`. The interface includes a top navigation bar with the IBM Watson Studio logo and a search bar. Below this, a breadcrumb trail indicates the current project: `Projects / Flight Delay Prediction / Flight delay prediction`. The main workspace contains a Jupyter Notebook with the following code cells:

```
...  
[0.11278656],  
[0.11171062],  
[0.03930239]]]  
  
In [43]: pred=MLR.predict(x_train)  
pred  
  
Out[43]: array([[0.06980701],  
[0.09264362],  
[0.10703753],  
...  
[0.08456845],  
[0.05297568],  
[0.10160735]])  
  
In [44]: from sklearn.metrics import r2_score  
acc=r2_score(y_test,y_pred)  
acc  
  
Out[44]: 0.014782427963560218  
  
In [57]: import pickle  
pickle.dump(MLR,open("flight.pkl","wb"))  
  
In [ ]:
```

At the bottom of the notebook, a status bar indicates `DEPLOYING MODEL`. The Windows taskbar at the very bottom shows the time as 10:43 AM on 10-11-2022.

8.2 User Acceptance testing:

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	3	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	78

9.Results:

9.1 Performance metrics:

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51

--

Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

10. Advantages and disadvantages:

10.1 Advantages

- Users might not reach their destined places on time after flight delay.
- The family/friends waiting to pick up the passenger ought to wait long.
- The successive flights may also get delayed in arrival airport
- The flights takeoff and takeon time gets affected.
- The airport get crowded.

10.2 Disadvantages

- Flight delay can be updated to passengers beforehand
- Users can plan their schedule
- Flight delays can be reported to concern airports.
- Flight delay updation on website
- Increase passengers convenience

11. Conclusion

In this project, we use flight data, weather, and demand data to predict flight departure delay. Our result shows that the Linear Recursion method yields the best performance compared to the SVM mode. In the end, our model correctly predicts 91% of the non-delayed flights. However, the delayed flights are only correctly predicted 41% of time. As a result, there can be additional features related to the causes of flight delay that are not yet discovered using our existing data sources.

12. Future Scope

The future work of this project includes incorporating a larger dataset. There are many different ways to pre-process a larger dataset like running a Spark cluster over a server or using a cloud-based services like AWS and Azure to process the data. With the new advancement in the field of deep learning, we can use Neural

Networks algorithm on the flight and weather data. Neural Network works on the pattern matching methodology.

Also, the scope of this project is very much confined to flight and weather data of United States, but we can include more countries like China, India, and Russia. Expanding the scope of this project, we can also add the flight data from international flights and not just restrict our self to the domestic flights.

13. Appendix

13.1 GitHub Account Link

<https://github.com/IBM-EPBL/IBM-Project-37997-1660367180.git>

13.2 Project Demo Link

https://drive.google.com/file/d/15EzV0G36_nXOGPmM4QeMbYlXicbShb_G/view?usp=sharing