ANNA UNIVERSITY : CHENNAI -600025

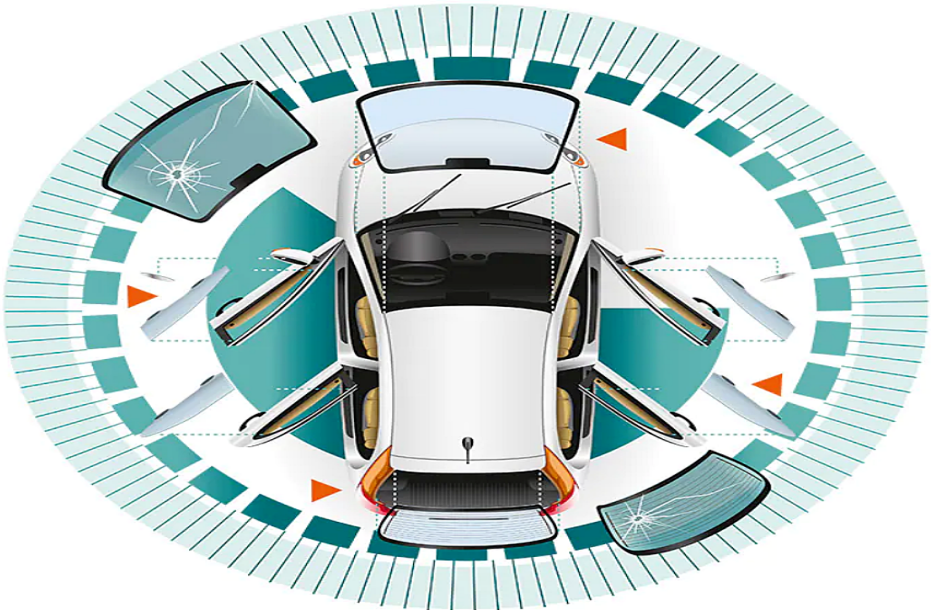
V. S. B. ENGINEERING COLLEGE

KARUR 639111

INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATOR FOR INSURANCE COMPANIES

Submitted by

K. FELOOMI	**922519104041**
B.DHARSHANA	**922519104031**
A.ASHIKA	**922519104015**
M.SHAJITHA PARVEEN	**922519104145**



Project Report Format

1. **INTRODUCTION**
    1. Project Overview
    2. Purpose
2. **LITERATURE SURVEY**
    1. Existing problem
    2. References
    3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    1. Empathy Map Canvas
    2. Ideation & Brainstorming
    3. Proposed Solution
    4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    1. Functional requirement
    2. Non-Functional requirements
5. **PROJECT DESIGN**
    1. Data Flow Diagrams
    2. Solution & Technical Architecture
    3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    1. Sprint Planning & Estimation
    2. Sprint Delivery Schedule
    3. Reports from JIRA

7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    1. Feature 1
    2. Feature 2
    3. Database Schema (if Applicable)
8. **TESTING**
    1. Test Cases
    2. User Acceptance Testing
9. **RESULTS**
    1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

Intelligent Vehicle Damage Assessment & Cost Estimator For Insurance Companies

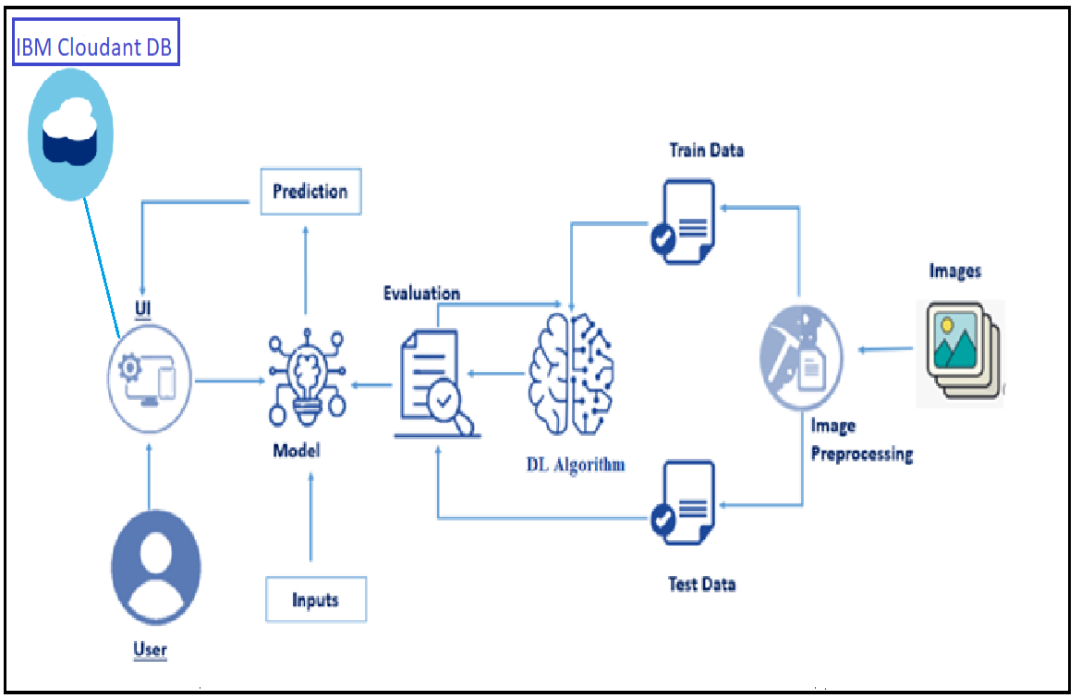
1.INTRODUCTION

1.1 Project Review

Nowadays, a lot of money is being wasted in the car insurance business due to leakage claims. Claims leakage Underwriting leakage is characterized as the discrepancy between the actual payment of claims made and the sum that should have been paid if all of the industry's leading practices were applied. Visual examination and testing have been used to may these results. However, they impose delays in the processing of claims.

The aim of this project is to build a VGG16 model that can detect the area of damage on a car. The rationale for such a model is that it can be used by insurance companies for faster processing of claims if users can upload pictures and the model can assess damage be it dent scratch from and estimates the cost of damage. This model can also be used by lenders if they are underwriting a car loan, especially for a used car.

Technical Intelligence:



1.2

Purpose

The aim of this project is **to build a VGG16 model that can detect the area of damage on a car**. The rationale for such a model is that it can be used by insurance companies for faster processing of claims if users can upload pictures and the model can assess damage(be it dent scratch from and estimates the cost of damage.This model can also be used by lenders if they are underwriting a car loan, especially for a used car.

2. LITERATURE SURVEY

2.1 Existing Problem

The claim settlement team finds it difficult to process the claim application if there is lack of proper documentation. An insurance policy is invalid if it has crossed its expiry date, claims raised with an invalid insurance policy are not settled. The insurance company might need copies of the applicant's driving license,vehicle registration certificate, etc. for verification.

2.2 References

https://www.upgrad.com/blog/basic-cnn-architecture/

https://machinelearningmastery.com/review-of-architectural-innovations-for- convolutional-neural-networks-for-image-classification/

https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp#:~:=Artificial%20intelligence%20(AI)%20refers%20to,as%20learning%20and%20problem%2Dsolving.

https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/ai-overview

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture- diagrams-2d20c9fda90d

2.3 Problem Statement Definition

Ms.Feloomi works with world-leading motor insurance companies to streamline claims and underwriting processes. Rather than asking a customer to bring their car into a garage to be assessed or manually reviewing images they've taken, Feloomi's AI technology can provide an initial assessment of the damage, allowing claims to be triaged into the appropriate channels.

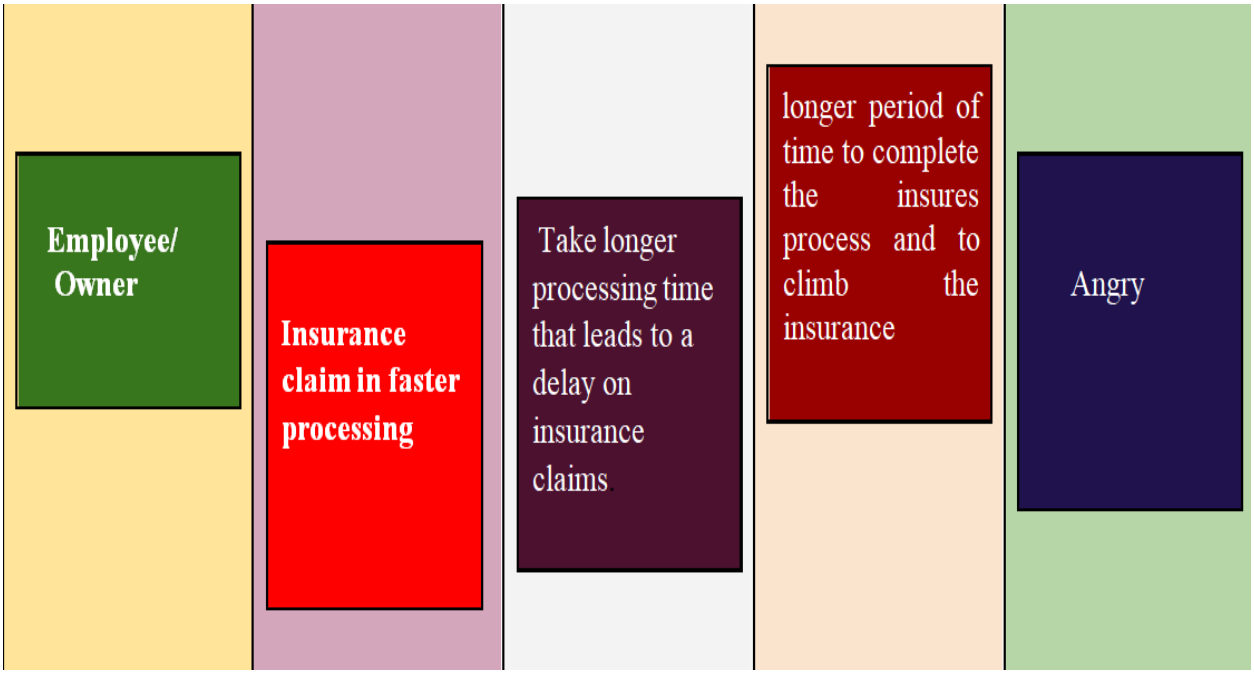
Small claims can be directly paid out or can be sent to specialist garages that can carry out PDR (paint less dent repair) or other smart repairs.

Larger claims, where longer repair times may mean a replacement vehicle has to be provided, can be handled through dedicated administrative processes. For insurance companies, this means they can process claims faster with less administrative costs, leading to lower premiums and happy customers.

Feloomi's mobile scanning technology is also being used as part of the

underwriting process. Customers can scan their vehicles to provide a baseline condition report that can be referenced when a claim is submitted. This ensures pre-existing damage is taken into account when assessing the cost of repair.

I am	Ms. Feloomi, is an employee at a motor insurance company.
I'm trying to	To make a review on the damaged vehicle andprovide the insurance claim in faster processing.
But,	Processing the review process by the physical surveyor may take longer processing time that leads to a delay on insurance claims.
Because	It takes a longer period to complete the insuresprocess and to climb the insurance.
Which makes me feel	Anxious and angry, by considering these as a main issue we applied image processing by using Artificial intelligence.

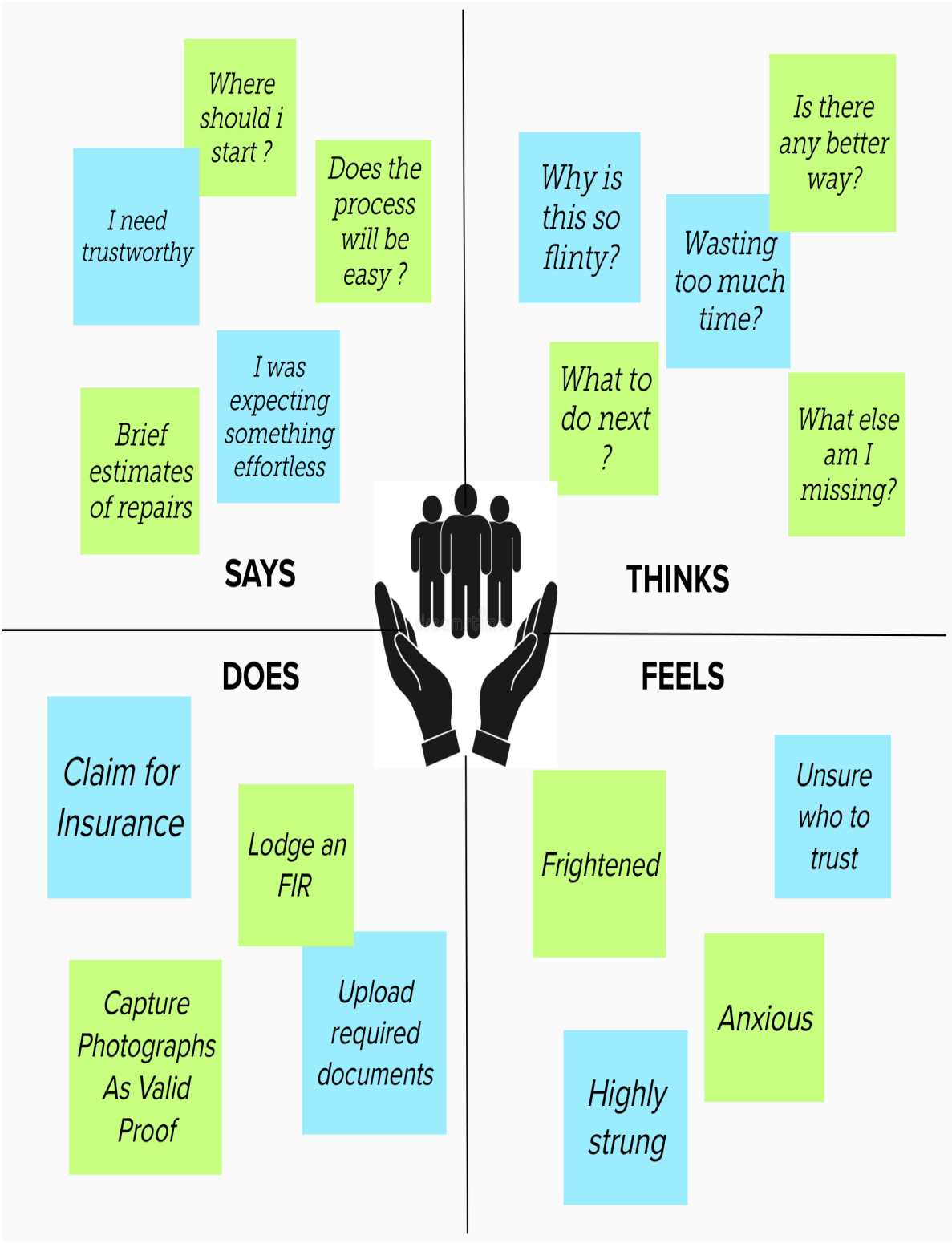


3. IDEATION AND PROPOSED SOLUTION:

3.1 Empathy Map Canvas

PROBLEM STATEMENT:

Nowadays, a lot of money is being wasted in the car insurance business due to leakage claims. Claims leakage or Underwriting leakage is characterized as the discrepancy between the actual payment of claims made and the sum that should have been paid if all of the industry's leading practices were applied. Visual examination and testing have been used to may these results. However, they impose delays in the processing of claims. The aim of this project is to build a model that can detect the area of damage on a car. The rationale for such a model is that it can be used by insurance companies for faster processing of claims if users can upload pictures and the model can assess damage that can be dent or scratch and estimates the cost of damage. This model can also be used by lenders if they are underwriting a car loan, especially for a used car.



3.2 Ideation and brainstorming:

Brainstrom

Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies

Nowadays, a lot of money is being wasted in the car insurance business due to leakage claims. Claims leakage Underwriting leakage is characterized as the discrepancy between the actual payment of claims made and the sum that should have been paid if all of the industry's leading practices were applied. Visual examination and testing have been used to may these results. However, they impose delays in the processing of claims.

The aim of this project is to build a VGG16 model that can detect the area of damage on a car. The rationale for such a model is that it can be used by insurance companies for faster processing of claims if users can upload pics and the model can assess damage(be it dent scratch from and estimates the cost of damage. This model can also be used by lenders if they are underwriting a car loan, especially for a used car.

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

1

Define your problem statement

Physical surveyor inspection of vehicle damage may cause delay in processing and it results in leakage claims. The customer may not receive an absolute claim for vehicle insurance.

PROBLEM

In today's world, it can observe that the number of vehicles we use is quickly expanding; let's agree that there isn't a single street without a car. As a result, an increase in the number of automobiles on the road may lead to an increase in the percentage of accidents occurring nearby; additionally, the number of accidents occurring nearby would be significant; the accidents would not be particularly serious, but the automobile would be damaged, prompting people to file insurance claims. The whole idea focuses on this question: how can a customer claim insurance more quickly? To keep the procedure quiet, a machine learning model is developed that utilizes image processing to categorize the photographs and calculate the percentage of damage to the car.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Defer judgment.
- Go for volume.
- Encourage wild ideas.
- Listen to others.
- If possible, be visual.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

Feloomi K

- Procedures
- Very little resource usage
- Is it trust worthy
- Where should I start?
- Effortless
- Faster processing ?

Dharshana B

- Data privacy
- FIR lodge
- Upload photos
- User friendly
- What to do next?
- Is there any better way ?

Ashika A

- Accurate estimate of money
- Stable internet
- Surveyor availability
- Tracking the process
- Upload FIR documents
- Verify the documents

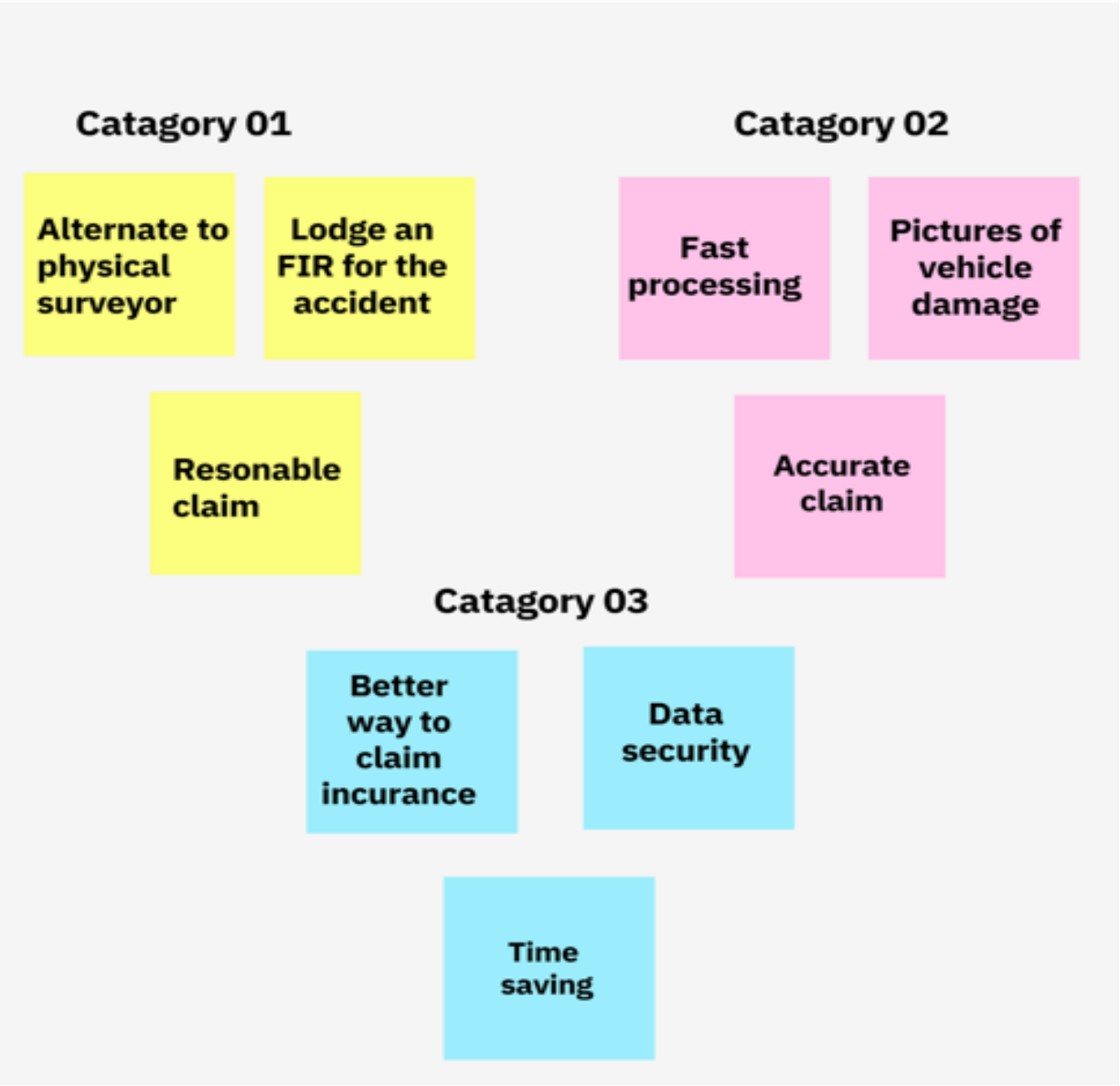
Shajitha Parveen M

- Usability
- user Understandability
- Delay in processing
- Time saving
- Alternative for surveyor
- Satisfied with the claim amount

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint

Define the components of a new idea or strategy.

Customer experience journey map

Understand customer needs, motivations, and obstacles for an experience.

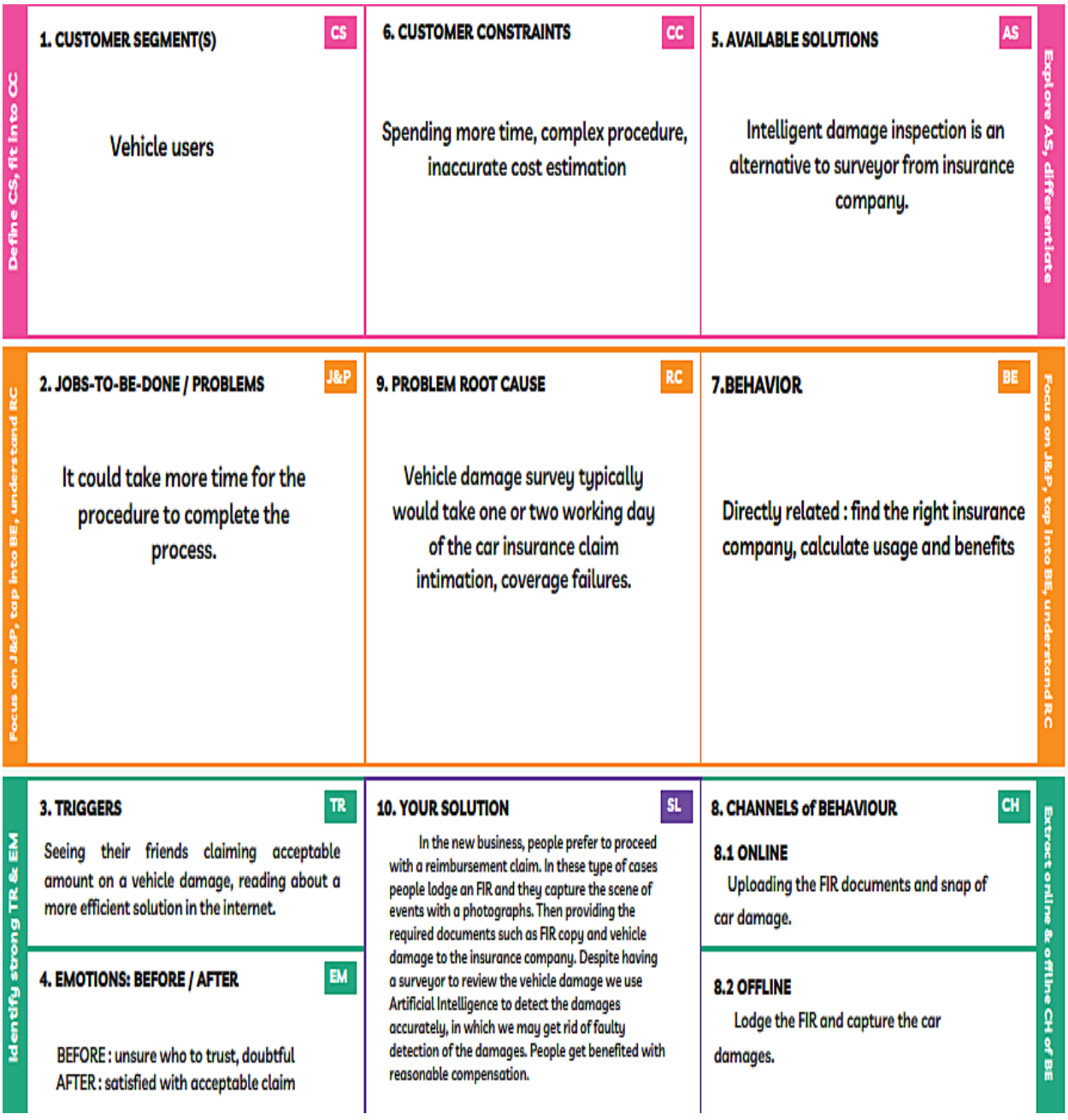
Strengths, weaknesses, opportunities & threats

Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

3.3 Proposed Solution:

S. No	Parameter	Description
1.	Problem Statement (Problem to besolved)	Physical surveyor inspection of vehicle damage may cause delay in processing and it results in leakage claims. The customer may not receive an absolute claim for vehicle insurance.
2.	Idea / Solution description	The solution benefits insurers by transforming theclaim value chain. It can speed up time-consumingclaim settlements and creates a better customer experience, while also increasing efficiency.
3.	Novelty / Uniqueness	The aim of this project is to build a VGG16 modelthat can detect the area of damage on a car.
4.	Social Impact / Customer Satisfaction	Customer is satisfied with reasonable claim and faster processing. This project estimates the claim accurately where both customer and Insurance company gets benefited.
5.	Business Model(Revenue Model)	VGG16 model has been used to detect the vehicle damages using Artificial Intelligence.
6.	Scalability of the Solution	We have developed an app to display the functionalities and connectivity of the model. The app is fully cloud-hosted, yielding scalability and on-demand service, without any requirement of the internal infrastructure or maintenance cost.

3.4 Problem Solution fit:



4. REQUIREMENT ANALYSIS:

4.1 Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Registration**	Registration through Form Registration through Gmail
FR-2	**User Confirmation**	Confirmation via Email Confirmation via OTP
FR-3	**User details**	Users are required to register their details. likename, age, date of birth, driving license, car number etc.
FR-4	**User requirements**	The user simply inputs vehicle damage images. The software will instantly generate an accurate reading of thebased on the image detection analysis in a readable format familiar to the customer. It compares the information already given and states the defect percentage and cost in that vehicle damage image.

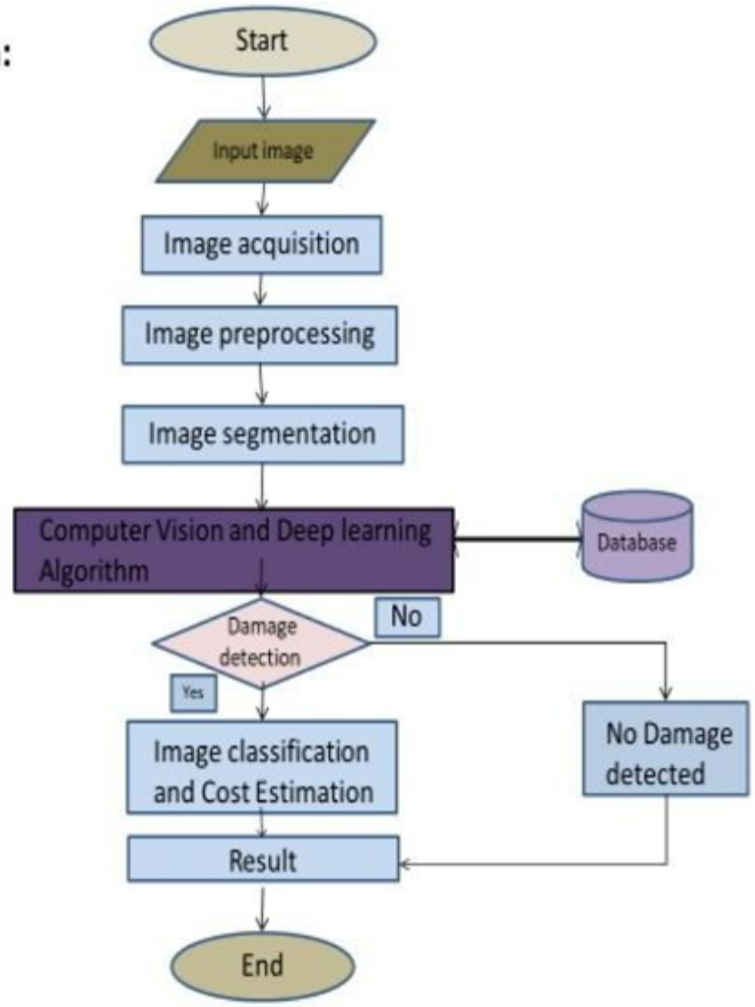
4.2 Non - Functional Requirements:

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	More efficient for the frequent users. userscan easily understand what the application does and feel satisfied with the system.
NFR-2	**Security**	1. AI powered vehicle damage assessment and cost estimator for insurance company should contain more security in which our data whichentered or maintained should be more security. 2. With the help of the username and password it provides more security in which it can access more secure and the data are private
NFR-3	**Reliability**	This application must perform without failure in 90percent of use cases during a month.it is more reliable.
NFR-4	**Performance**	This application supporting 1,050 users per hour must provide 5 seconds or less response time in a desktop browser, including the rendering of text and images, over an LTE connection. The performance of this application is effective and efficien
NFR-5	**Availability**	The web dashboard must be available to user's 99.9 percent of the time every month during business hours EST.Users can access any time and any where.
NFR-6	**Scalability**	The application must be scalable enough to support 10,000 visits at the same time while maintaining optimal performance and efficient to retrieve image in large scale thus improving scalability.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

Data Flow Diagram:



5.2 Solution and Technical Architecture

TECHNICAL ARCHITECTURE:

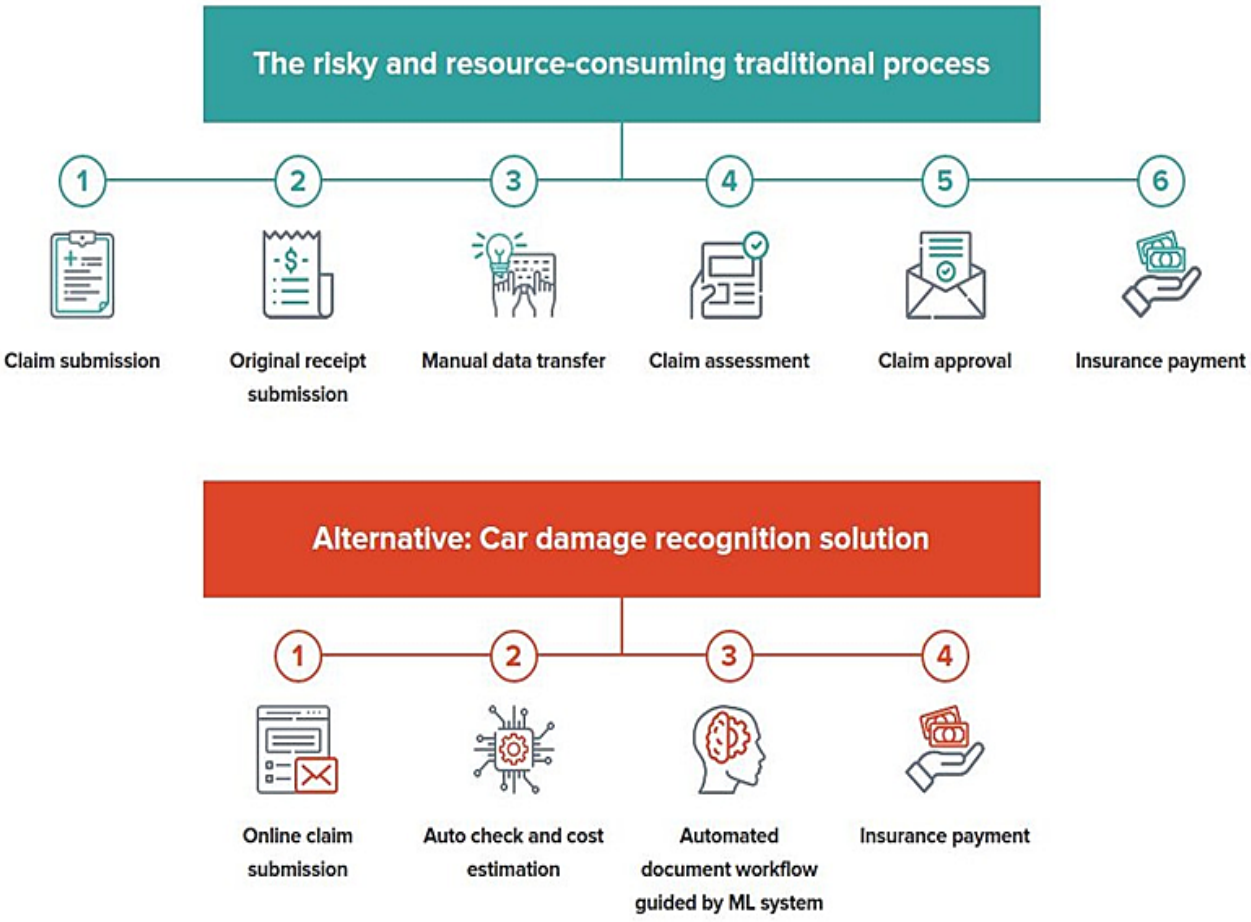


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts withapplication e.g. Web UI, Mobile App,Chatbot etc.	HTML, CSS, JavaScript /Angular Js / ReactJs etc.
1.	Application Logic-1	Logic for a process in theapplication	Python
1.	Application Logic-2	Logic for a process in theapplication	CNN, image processing
1.	Application Logic-3	Logic for a process in theapplication	Artificial Intelligence
1.	Database	Data Type, Configurationsetc.	MySQL, NoSQL, etc.
1.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
1.	File Storage	File storage requirements	IBM Block Storage or OtherStorage Service or Local Filesystem
1.	External API-1	Purpose of External API used in the application	Vehicle API, etc.
1.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
1.	Machine LearningModel	Purpose of Machine Learning Model	Object Recognition Model,etc.
1.	Infrastructure (Server/ Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud ServerConfigura tion :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworksused	Technology ofOpen source framework
1.	Security Implementatio ns	List all the security / access controlsimplemented, use of firewalls etc.	e.g. IAM Controls, OWASPetc.
1.	Scalable Architecture	Justify the scalability of architecture(3 – tier, Micro-services)	Technology used
1.	Availability	Justify the availability of application(e.g.use of load balancers, distributed servers etc.)	Technology used
1.	Performance	Design consideration for the performance of the application (number of requests per sec,use of Cache, use of CDN's) etc.	Technology used

5.3 User Stories

Ms.Feloomi works with world-leading motor insurance companies to streamline claims and underwriting processes. Rather than asking a customer to bring their car into a garage to be assessed or manually reviewing images they've taken, Feloomi's AI technology can provide an initial assessment of the damage, allowing claims to be triaged into the appropriate channels. Small claims can be directly paid out or can be sent to specialist garages that can carry out PDR (paint less dent repair) or other smart repairs. Larger claims, where longer repair times may mean a replacement vehicle has to be provided, can be handled through dedicated administrative processes.

For insurance companies, this means they're able to process claims faster with less administrative costs, leading to lower premiums and happy customers. Feloomi's mobile scanning technology is also being used as part of the underwriting process. Customers can scan their vehicles to provide a baseline condition report that can be referenced when a claim is submitted. This ensures pre-existing damage is taken into account when assessing the cost of repair.

I am	Ms. Feloomi, is an employee at a motor insurancecompany.
I'm trying to	To make a review on the damaged vehicle andprovide the insurance claim in faster processing.
But,	Processing the review process by the physical surveyor may take longer processing time that leads to a delay on insurance claims.
Because	It takes a longer period to complete the insuresprocess and to climb the insurance.
Which makes me feel	Anxious and angry, by considering these as a main issue we applied image processing by using Artificial intelligence.


Employee/ Owner
Insurance claim in faster processing
Take longer processing time that leads to a delay on insurance claims.
longer period of time to complete the insures process and to climb the insurance
Angry

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning, Estimation and Sprint Delivery schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As an owner of a particular vehicle, I can log into the application by entering email & password.	2	High	K.Feloomi
Sprint-1	User Confirmation	USN-2	As an owner of a particular vehicle, I will receive confirmation email once I have registered for the application.	1	Medium	M.Shajitha Parveen
Sprint-1	Login	USN-3	As an owner of a particular vehicle, I can log into the application by entering email & password.	2	High	A.Ashika

Sprint-2	Data Collection	USN-1	Download the dataset used in intelligent vehicle damage assessment & cost estimator for insurance companies.	2	High	B.Dharshana
Sprint-2	Image Pre Processing	USN-1	Improve the image data that suppresses unwilling distortions or enhances some image features important for further processing, although performing somegeometric transformations of images like rotation, scaling, etc.	2	High	K.Feloomi
Sprint-3	Model Building	USN-1	Define the model architecture and adding CNN layer and testingsaving the model.	2	High	B.Dharshana

Sprint-3	Cloud DB	USN-1	Below are steps that need to follow for creating and using cloudant service. ● Register & login to IBMcloud ● Create service instance ● Creating credentials ● Launch clouda DB ● Create database	2	High	K.Feloomi
Sprint-4	Application Building	USN-1	Building a web application that is integrated into the model we built. A UI is provided to the user where he has uploaded the image Based on the saved model, the uploaded image will be analysed and prediction is showcased on the UI.	2	High	A.Ashika
Sprint-4	Train The Model On IBM	USN-1	Build Deep learning model and computer vision Using the IBM cloud.	2	High	K.Feloomi, M.Shajitha Parveen

6.3 Reports from JIRA

BACKLOG

ROADMAP

7.CODING & SOLUTIONING

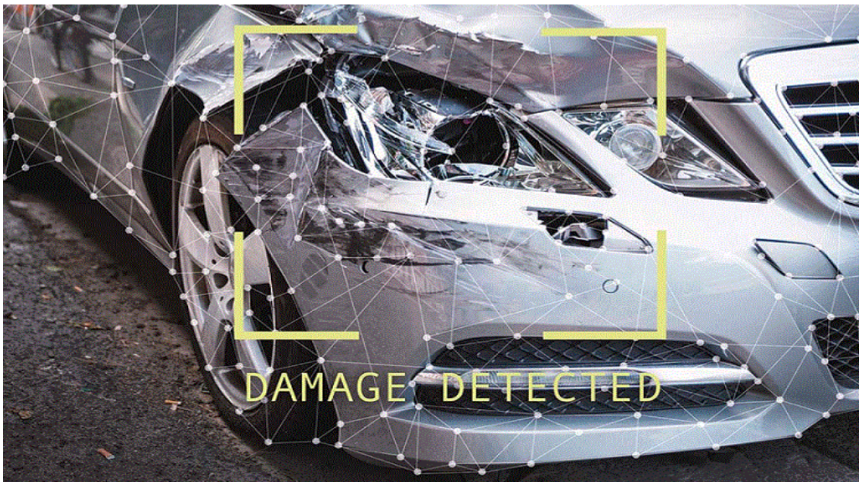
7.1 Feature 1:

Intelligent damage determination system can be used to determine the appearance damage of vehicles in small cases. The system completes the whole process of survey and damage determination through four functions. They are:
(1) ***Accident investigation***: Photographs of target vehicles and multiple trio vehicles were taken and uploaded, intelligent recognition, information input, intelligent recognition and event finalization are completed in accident investigation.
(2) ***Intelligent image damage assessment***: image damage assessment is achieved by intelligent component recognition and intelligent damage recognition.
(3) ***Damage result output***: Damage results including maintenance scheme

recommendation and maintenance price recommendation are automatically given according to damage recognition results.

(4) ***Vehicle insurance anti-fraud***: In the process of fixing the damage, the anti-fraud screening of vehicle insurance is completed by means of image fraud recognition and logical detection.

Intelligent damage assessment system can assist the damage locator in the front-end damage detection process. The operator only needs to take several photos to upload according to the requirements, and the system can automatically identify the damage degree of the damaged parts and components. The system in the back-end nuclear damage link can provide auxiliary nuclear damage and anti-fraud services. It can identify the cases of fixed-loss errors through the logical recognition of vehicle parts, image fraud recognition, fixed-loss logic recognition, etc. At the same time, it can also meet the demands of anti-fraud and leakage prevention. At present, the intelligent damage assessment system can realize the appearance damage of passenger cars, including CAR, SUV, MPV and VAN. The applicable damage range covers all types of damage of vehicle exterior parts; the applicable environment range covers rain and snow environment, dark environment (vehicle can be seen by human eyes), strong light environment and other scenarios.



7.2 Feature 2:

The Realization Path of Intelligent Damage Assessment System

Intelligent damage determination system has four functional modules: accident investigation, image damage, result output and anti-fraud of automobile insurance. The implementation path is described in detail below.

- ***Accident Investigation:*** Accident investigation module includes the photography of certificates and vehicle photos, the intelligent recognition of certificate photos and the intelligent stereotyping work based on the basic information data of vehicle accessories.
- ***Take Photos:*** The photographs taken in the accident investigation of intelligent damage determination system include driving license (front and side pages), driving license (front and side pages), person-car photograph, vehicle corner photograph and vehicle damage photograph. In order to apply the photograph of vehicle damage to the image damage based on artificial intelligence image recognition algorithm, some shooting requirements are put forward

    1. Using smartphones to shoot pictures with no less than 2 million pixels
    2. For the photography of vehicle damage, it is necessary to shoot the vehicle damage head-on so that the damage location is as far as possible in the center of the picture. The shooting distance is about 1 meter, and it is suitable to shoot clearly.
    3. Multiple damage or cross-component vehicle appearance damage, if the damage distance is relatively close, then a photo can be taken, if the damage distance is relatively far, can not take a photo, then need to be taken separately. In addition, the intellectualization of photography is also reflected in the following aspects: When taking photographs, it automatically identifies whether it is a document photo, a person-car photo, etc. If the photograph does not meet the requirements is not approved, it needs to be re-taken. At the same time, it is not mandatory to satisfy what angle of shooting can be taken, which is easy to operate and makes it easier for the damage fixer or other users to use.

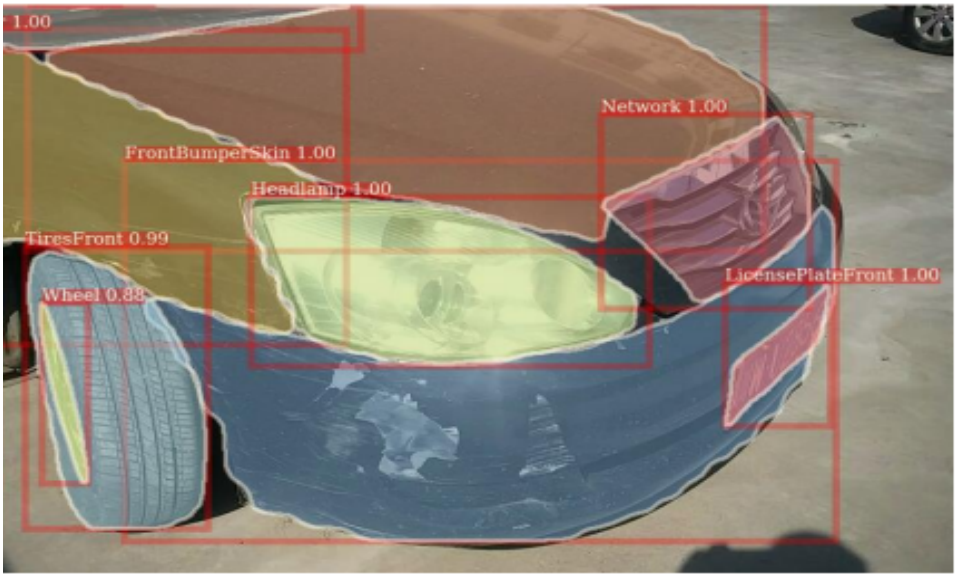
- ***Intelligent ID Recognition:*** For the photos of the uploaded driving license (front and side pages), driving license (front and side pages) and other documents, the intelligent damage determination system embedded OCR recognition technology. The VIN code, license plate number, engine number, driver's name and other information of the uploaded driving license and driver's license can be intelligently recognized and filled in. At present, the embedded OCR technology can recognize Chinese characters, English upper and lower case letters, numbers and other information, and the recognition accuracy is 98.5%. Aiming at the problem of manual input for most fixed-loss products at mobile terminals, the embedding of OCR technology can not only save the time for the invalidation personnel to input the certificates without basic information, such as the three vehicles, but also effectively avoid the problems of input errors, which greatly improves the work efficiency.

- ***Intelligent Stereotyping and Fixing:*** The advantages of intelligent loss determination system are also reflected in its abundant basic information data. Through VIN code, the basic information database of vehicles and accessories can be automatically linked to realize the output of specific vehicle information such as brand, vehicle system, vehicle type, and OE code of parts corresponding to vehicle type, so as to realize one-to-one correspondence between vehicles and accessories.

- ***Intelligent Image Damage Assessment:*** The core of intelligent damage fixing products is to determine which kind of damage happened to the exterior parts of the vehicle by image. The system has been experimented many times in the development of intelligent image damage algorithm. Finally, it divides the problem into three parts: the recognition of appearance parts by image, the recognition of damage parts by image, and the determination of damage parts by relative position relationship.

- ***Vehicle Appearance Component Recognition Algorithms***: According to the statistics of vulnerable parts in vehicle accidents, thirty-one vehicle exterior parts have been identified in this product. Each part is divided into front and back parts, regardless of left and right parts. Aiming at the recognition of 31 vehicle

appearance parts (regardless of left or right), the recognition algorithm for panoramic or local vehicles is realized, in the complex environment of rain and snow, too strong light or dark, by using the self-built data set of vehicle appearance parts and the depth learning target detection algorithm. The component recognition algorithm AP50 is 88.7%.



- ***Damage Recognition Algorithms for Vehicle Appearance Components***. This product is aimed at six types of vehicle appearance damage, and also applies the deep learning target detection method. Through the self-built damage data set, it can recognize high-light pictures, low contrast pictures and multi-category mixed damage. The damage recognition algorithm AP50 is 87.6%.

- ***An Algorithm for Locating Components and Damages:*** By calculating the intersection relationship between the polygon identified by the algorithm of vehicle appearance components and the rectangular position identified by the algorithm of appearance damage, the appearance parts where the damage occurs are finally determined. At present, based on vehicle appearance component recognition algorithm, vehicle appearance damage recognition algorithm and image position determination algorithm, the comprehensive accuracy of image damage determination algorithm reaches 87.3%.

- ***Output of Loss Assessment Result:*** The output of fixed-loss results can not be separated from maintenance rules and repair logic. Among them, the maintenance rules are based on the experience of fixing damage and testing the appearance of components in the specific material damage needs to be maintained. The repair logic needs to formulate the damage inclusion relation logic, for example, if there are two damages in the same component, the maintenance scheme should adopt the scheme with higher maintenance level. Based on the maintenance rules and the repair logic, the repair knowledge base is established, the output of the maintenance plan is finally realized.

- ***Vehicle Insurance Anti-Fraud****:* The automobile insurance fraud of intelligent damage determination system is mainly embodied in the following three aspects. It realizes the fraud recognition in the whole process of damage determination and can effectively control the cost expenditure of insurance companies.

APPLICATION CODE:

```
from flask import Flask, render_template,request, redirect , session
from cloudant.client import Cloudant
import re
import numpy as np
from tensorflow.keras import models
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.python.ops.gen_array_ops import concat
from tensorflow.keras.applications.inception_v3 import preprocess_input
import requests


import os
import tensorflow
app=Flask(__name__)

client = Cloudant.iam('6e0ea55b-5a46-4999-868c-ac161198beaf-
bluemix','wdmCb0gFYsyVbuInTstD2M-
Hy_RI82jxU_BCqaKNloYR',connect=True)
myDatabase = client.create_database('myproject')
print(client)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/login')
def login():
    return render_template('login.html')
```

```
@app.route('/register')
def register():
    return render_template('Register.html')

@app.route('/afterreg', methods=['POST'])
def afterreg():
    x=[x for x in request.form.values()]
    print(x)
    data={'_id':x[1],'name':x[0],'psw':x[2]}
    print(data)
    query={'_id':{'$eq':data['_id']}}
    docs = myDatabase.get_query_result(query)
    print(docs)
    print(len(docs.all()))
    #return "<p> Sucessful Register</p>"
    if len(docs.all())==0:
        url = myDatabase.create_document(data)
        return render_template('Register.html',pred="Register Suecess , Need
to login")
    else:
        return render_template('Register.html',pred="You are already member,
Need to login")

@app.route('/afterlogin',methods=['POST'])
def afterlogin():
    user = request.form['_id']
    passw = request.form['psw']
    print(user,passw)

    query ={'_id':{'$eq':user}}
    docs=myDatabase.get_query_result(query)
    print(docs)
```

```
    print(len(docs.all()))
    print(docs[0])
    print(docs.all())
    print("Query ",query)


    if(len(docs.all())==0):
        return render_template('login.html',pred="User Name is Not Found.")
    else:
        if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
            return render_template('prediction.html')
        else:
            print("invalid user")

@app.route('/logout')
def logout():
    return render_template('logout.html')

@app.route('/prediction')
def prediction():
    return render_template('prediction.html')

@app.route('/result', methods=["GET","POST"])
def res():
    if request.method=="POST":
        f=request.files['image']
        basepath=os.path.dirname(__file__)
        filepath=os.path.join(basepath,'uploads',f.filename)
        f.save(filepath)

@app.route("/prediction", methods=['GET', 'POST'])
def prediction():
```

```
    if request.method == 'POST':


        file = request.files['fileupload']
        file.save('static/Out/Test.jpg')

        import warnings
        warnings.filterwarnings('ignore')

        import tensorflow as tf
        classifierLoad = tf.keras.models.load_model('body.h5')

        import numpy as np
        from keras.preprocessing import image

        test_image = image.load_img('static/Out/Test.jpg', target_size=(200,
200))
        img1 = cv2.imread('static/Out/Test.jpg')
        # test_image = image.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis=0)
        result = classifierLoad.predict(test_image)

        result1 = ''

        if result[0][0] == 1:

            result1 = "front"

        elif result[0][1] == 1:

            result1 = "rear"

        elif result[0][2] == 1:
```

```
        result1 = "side"

    file = request.files['fileupload1']
    file.save('static/Out/Test1.jpg')

    import warnings
    warnings.filterwarnings('ignore')

    import tensorflow as tf
    classifierLoad = tf.keras.models.load_model('level.h5')

    import numpy as np
    from keras.preprocessing import image

    test_image = image.load_img('static/Out/Test1.jpg', target_size=(200,
200))
    img1 = cv2.imread('static/Out/Test1.jpg')
    # test_image = image.img_to_array(test_image)
    test_image = np.expand_dims(test_image, axis=0)
    result = classifierLoad.predict(test_image)

    result2 = ''

    if result[0][0] == 1:

        result2 = "minor"

    elif result[0][1] == 1:

        result2 = "moderate"

    elif result[0][2] == 1:
        result2 = "severe
```

```
        if (result1 == "front" and result2 == "minor"):
            value = "3000 - 5000 INR"
        elif (result1 == "front" and result2 == "moderate"):
            value = "6000 8000 INR"
        elif (result1 == "front" and result2 == "severe"):
            value = "9000 11000 INR"

        elif (result1 == "rear" and result2 == "minor"):
            value = "4000 - 6000 INR"

        elif (result1 == "rear" and result2 == "moderate"):
            value = "7000 9000 INR"

        elif (result1 == "rear" and result2 == "severe"):
            value = "11000 - 13000 INR"

        elif (result1 == "side" and result2 == "minor"):
            value = "6000 - 8000 INR"

        elif (result1 == "side" and result2 == "moderate"):
            value = "9000 - 11000 INR"

        elif (result1 == "side" and result2 == "severe"):
            value = "12000 - 15000 INR"

        else:
            value = "16000 - 50000 INR"

        return render_template('userhome.html', prediction=value)

if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)
```

MODEL CODE:

```
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.models import model_from_json
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
batch_size = 32



from tensorflow.keras.preprocessing.image import ImageDataGenerator



# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(rescale=1/255)


# Flow training images in batches of 128 using train_datagen generator
train_generator = train_datagen.flow_from_directory(
    'body',  # This is the source directory for training images
    target_size=(200, 200),  # All images will be resized to 200 x 200
    batch_size=batch_size,
    # Specify the classes explicitly
    classes = ['00-front','01-rear','02-side'],
    # Since we use categorical_crossentropy loss, we need categorical labels
    class_mode='categorical')
```

```
import tensorflow as tf
#cnn Model
model = tf.keras.models.Sequential([
   # Note the input shape is the desired size of the image 200x 200 with 3 bytes
color
   # The first convolution
   tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(200, 200, 3)),
   tf.keras.layers.MaxPooling2D(2, 2),
   # The second convolution
   tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
   tf.keras.layers.MaxPooling2D(2,2),
   # The third convolution
   tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
   tf.keras.layers.MaxPooling2D(2,2),
   # The fourth convolution
   tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
   tf.keras.layers.MaxPooling2D(2,2),
   # The fifth convolution
   tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
   tf.keras.layers.MaxPooling2D(2,2),
   # Flatten the results to feed into a dense layer
   tf.keras.layers.Flatten(),
   # 128 neuron in the fully-connected layer
   tf.keras.layers.Dense(128, activation='relu'),
   # 5 output neurons for 5 classes with the softmax activation
   tf.keras.layers.Dense(3, activation='softmax')
])


model.summary()

from tensorflow.keras.optimizers import RMSprop
early = tf.keras.callbacks.EarlyStopping(monitor='val_loss',patience=5)
```

```
model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(lr=0.001),
              metrics=['accuracy'])

total_sample=train_generator.n

n_epochs = 20

history = model.fit_generator(
      train_generator,
      steps_per_epoch=int(total_sample/batch_size),
      epochs=n_epochs,
      verbose=1)



model.save('body.h5')


acc = history.history['accuracy']

loss = history.history['loss']

epochs = range(1, len(acc) + 1)

# Train and validation accuracy
plt.plot(epochs, acc, 'b', label=' accurarcy')

plt.title('  accurarcy')
plt.legend()

plt.figure()

# Train and validation loss
plt.plot(epochs, loss, 'b', label=' loss')
plt.title('  loss')
plt.legend()
plt.show()
```

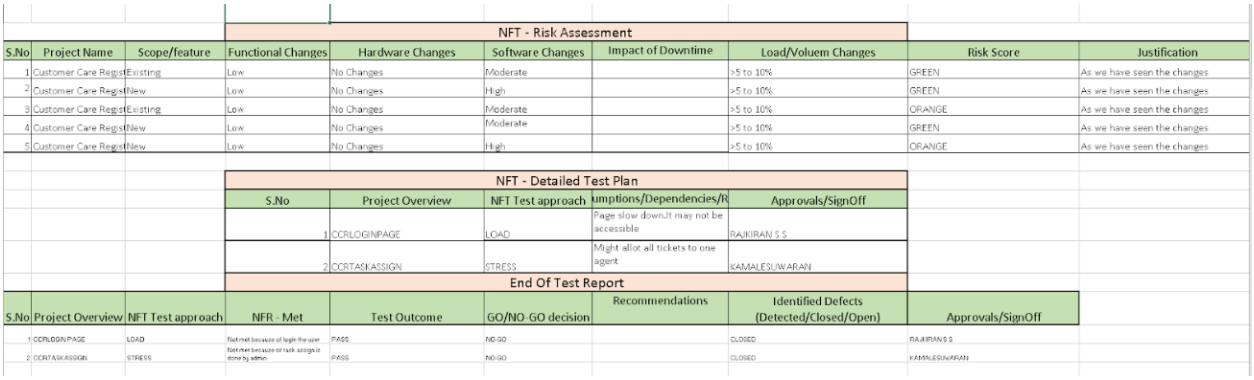
8. RESULTS

8.1 Performance Testing

The sources of this claim leakage vary, inaccurate estimation of vehicle damage after an accident is one of them. Therefore, it becomes crucial to improve the accuracy of payout calculations in order to reduce this waste.

However, the process of assessing damage is both effort- and time-consuming with multiple parties involved. While an insurer is obviously responsible for covering the repair cost, the estimation itself is usually done by a body shop. The whole procedure involves experts from both sides, which often results in weeks or months spent on each case.

Apart from being a time sink, manual inspection is also highly prone to human error, which can lead to inaccurate or unfair payouts.



NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justification
1	Customer Care Regist	Existing	Low	No Changes	Moderate		>5 to 10%	GREEN	As we have seen the changes
2	Customer Care Regist	New	Low	No Changes	High		>5 to 10%	GREEN	As we have seen the changes
3	Customer Care Regist	Existing	Low	No Changes	Moderate		>5 to 10%	ORANGE	As we have seen the changes
4	Customer Care Regist	New	Low	No Changes	Moderate		>5 to 10%	GREEN	As we have seen the changes
5	Customer Care Regist	New	Low	No Changes	High		>5 to 10%	ORANGE	As we have seen the changes

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	sumptions/Dependencies/R	Approvals/SignOff
1	CCRLOGINPAGE	LOAD	Page slow downIt may not be accessible	RAJKIRAN S S
2	CCRTASKASSIGN	STRESS	Might allot all tickets to one agent	KAMALESUWARAN

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	CCRLOGINPAGE	LOAD	Not met because of login the user	PASS	NO-GO		CLOSED	RAJKIRAN S S
2	CCRTASKASSIGN	STRESS	Not met because of task assign is done by admin	PASS	NO-GO		CLOSED	KAMALESUWARAN

Insurer Concerns about Premium Leakage

Source: *2016 Verisk Auto Insurance Premium Leakage Survey*

70% monitor leakage today

26% intend to within the next five years[2]

The solution speeds up data processing saving the company's spendings on human resources, defending form fraud (in 80% and more) and boosting the process of image data analytics in times. The system is used on sight and guides a user on actions to meet photo requirements. Deploying Car Damage Recognition, businesses replace a human-operated, time-consuming process of claims proceeding and approval with Artificial Intelligence and analytic systems.

While Artificial Intelligence helps to automate car damage assessment, the time and resource investment is significant. Without proper infrastructure and equipment in place, your project may not meet its goals. So, what are the recommendations and lessons learned?

- Building up extensive and diversified data sets requires **collaboration** of multiple parties: drivers submitting quality photos of their cars, insurance companies sharing the available data (with a consent of the insured), and integration with body shops involved in the repair process.
- For accurate image recognition, the data set should feature **high-resolution photos**, which are not blurry, light-struck, or taken from an inappropriate angle. The associated recommendation here is to provide drivers with tips and hints via a mobile app on how to take a picture of a sufficient quality. If the quality of photos is inappropriate, you could deliver **notifications** to users urging them to take another photo.
- Furthermore, each car part may look different when shot at **various angles**. This way, you need to encourage users to take as many

photos as possible to get the full picture.

- As mentioned earlier, proper **infrastructure** is key to success

This way, it is possible to create 3D models capable of generating realistic photos and using them to train classificators to achieve almost 100% precision. After that, such a solution can be integrated into the day-to-day routine of insurance companies, car rental services, and body shops. Through automation of manual processes, these organizations can significantly reduce time and effort spent on human inspection, cutting operational costs on preliminary damage assessment.

1. Finding a proper data set

Training Artificial Intelligence models requires a sufficient data set of relevant images. The more varied the images are, the better the model will be able to classify images appropriately.

In the context of car damage assessment, obtaining a substantial amount of images is a challenge, since there is no public database for images depicting damaged vehicles. While it may be possible to come up with a raw data set through web scraping, working with car insurance companies—which already have numerous images of broken car parts—may also be a feasible option. So, a company needs to evaluate the most optimal way in terms of ROI or time by deciding whether to buy the data set, get it from an industry partner, or to build/collect it from scratch.

Even having obtained a collection of images, one should ensure the

pictures satisfy the demands for size, quality, etc.

2. Preprocessing

Preprocessing image data sets is a crucial step in speeding up and obtaining better training results for models. This activity may span a variety of tasks: applying filters, removing noise, enhancing contrast, downsampling images, etc. With proper preprocessing, photos that are too blurry, too dark, or too bright can also be utilized. This way, photos where a car is not initially detected or looks ambiguous can be adjusted to work.

For instance, this can be done with OpenCV, one of the most used machine learning libraries for image preprocessing. It has over 2,500 optimized algorithms for identifying objects and putting pictures together to create a high-resolution image of an entire scene.

3. Building a model

After you have a quality data set at hand, there are still some considerations when building a machine learning model.

- Creating and training a model takes time. The main goals here can be associated with a) detecting a vehicle and b) distinguishing between its exterior parts in the photo. Aiming for descent accuracy, numerous corrections may be needed: e.g., more input data, improved algorithms, etc. Eventually, the process may take weeks or even more,

taking into account all the necessary adjustments.

- As a result, computing powers for the model to process images at an acceptable rate may become costly—thousands of dollars worth.
- Finally, finding means to unbiasedly estimate vehicle damage can also be tricky.

For estimating the damage extent, our team initially applied binary classification and then ran the data set through machine learning algorithms built upon the MobileNetV2 neural network. At the end of the day, the solution achieved 90% of precision in both detecting the damaged car parts and assessing the loss incurred.

4. Optimizing performance and costs

As insurance companies have to deal with damage assessment on a daily basis, the working solution also needs to demonstrate resonating performance, so it won't take weeks to seal the deal. Ideally, a car driver should be able to get an approximate damage estimate right away.

To maximize performance, an engineer can rely on the capacities of GPU or parallel computing to the fullest. The importance of hardware characteristics cannot be neglected, as well.

5. Privacy

When processing photos for damage estimation and sharing between parties, it is critical to ensure that the privacy of car owners remains intact. In most cases, it is possible to encounter images containing vehicle license plates, which may be used to identify individual car owners. This may cause

privacy concerns, especially in European regions where the General Data Protection Regulation (GDPR) is enforced.

To avoid complications, it is worth following certain guidelines.

- Vehicle owners need to be aware about data collection, the reasons for data processing, and how long their data is being retained.
- Data cannot be kept indefinitely and should be deleted upon the owner's request.

If your solution involves integration with third parties, such as medical institutions, insurance companies, etc., data should be shared in a secure way—requiring consent, as well. Not to mention the creation of a safe data storage, encryption, or even using blockchain, if needed.

10.ADVANTAGES & DISADVANTAGES

The insurance sector is embracing new technologies to address some of its major customer pain points. Leveraging artificial intelligence (AI) in insurance, NLP technology, Machine Learning (ML) helps to tackle customer experience in health insurance and combat insurance fraud.Indeed, **87% of surveyed insurers are already seeing their companies invest $5 million or more in AI technology each year.**

Furthermore, corporate interest in insurance startups has led to some impressive fundraising.

The benefits of implementing AI in insurance seem clear to stakeholders in this ecosystem. Indeed, 84% of surveyed French investors believe AI will revolutionize the insurance sector. Moreover, 66% of insurers feel AI can help improve workforce productivity.

- By implementing AI into their processes, insurers can save time, reduce costs, improve customer experience and increase profitability.
- AI can also transform typically tedious and time-consuming processes i.e. underwriting, claims management, fraud detection, customer service.
- With AI insurers can also reduce human errors. These errors can be common due to factors like changing the regulation of information to analyze for fraud prevention.

11.CONCLUSION

Conversational AI abilities allow it to understand the context of an entire conversation and adapt the response accordingly. The highly intelligent and personalized approach increases the customer's confidence in the overall service, while improving the customer experience in the insurance industry.

Insurers can rely on AI chatbots to deploy on insurance processes by taking on the always-on customer service channel. Conversational AI enabled virtual agents can familiarize customers with various insurance policies and rank them based on customer needs.

Intelligent bots can ask the right questions and understand the customer's needs and expectations. With Natural Language Understanding (NLP), it is easy to figure out the customer's real intent based on their responses. Bots can also recommend the best policies based on the customer's profile, risk assessment, and the type of cover needed.

The automotive industry is at the cusp of a phenomenal transformation. Automakers worldwide have started incorporating advanced technologies in their products and operations to ensure that they stay a step ahead of competitors in the market.

AI for motor insurance companies witnessed working its magic in an autonomous car using machine learning and also automating motor insurance underwriting with AI.

Here is an example of the largest general insurance company in Australia, IAG deploys AI to assess vehicle damage and reduce claim times.

The company uses an AI-based system to assess damage from motor vehicle accidents and speed up customers' insurance claims. The support team can predict whether a car is a total loss following an accident, and by mitigating the need to tow the vehicle to a garage for

assessment, can reduce claims processing times from weeks to days.

Key Use Cases for AI Auto Insurance

- **Predictive Analytics for Manufacturing:** The use of AI in vehicle manufacturing helps automakers reduce manufacturing costs and provides a safer and more efficient factory floor. It also helps in predicting malfunctions in automotive parts
- **Vehicle Maintenance Recommendations:** AI and ML algorithms for motor insurance companies help in providing recommendations to drivers about vehicle maintenance.
- **Driver Behavior Analytics:** AI auto insurance based automotive applications offer a plethora of valuable in-car analytics. Cameras and IR sensors can detect the driver's behavior accurately and provide warning signals to avoid accidents

AI-based technology is being applied to many different industries now, including the insurance industry, with residential and commercial property insurance. It has the potential to transform how property insurance works.

Artificial Intelligence helps to improve the claims process, making it

faster and more accurate than ever before. Its ability to automate many tasks is making everything more efficient, which translates to better customer service and outcomes.

Key Use Cases: AI for Property Insurance Companies

- **Risk Analysis & Underwriting:** Using AI to analyze data can result in a more accurate assessment of potential perils, a determination of the best policy and coverage for your needs, and ways to reduce your risk of future claims.
- **Fraud Detection:** AI chatbots help to automate many processes that are often subject to fraud and take action without needing human support. It can also flag new policies, helping to identify people who are more likely to commit insurance fraud.

- **Automated Claims Processing:** AI can help customers report a claim, record damage or loss, verify policy and limits, communicate with customers, and alert customers and insurers about the status of claims.

12.FUTURE SCOPE

Welcome to the future of insurance, as seen through the eyes of Scott, a customer in the year 2030. His digital personal assistant orders him a a vehicle with self-driving capabilities for a meeting across town.

Upon hopping into the arriving car, Scott decides he wants to drive today and moves the car into “active” mode. Scott’s personal assistant maps out a potential route and shares it with his mobility insurer, which immediately responds with an alternate route that has a much lower likelihood o accidents and auto damage as well as the calculated adjustment to his monthly premium. Scott’s assistant notifies him that his mobility insurance premium will increase by 4 to 8 percent based on the route he selects and the volume and distribution of other cars on the road. It also alerts him that his life insurance policy, which is now priced on a “pay-as-you-live” basis, will increase by 2 percent for this quarter. The additional amounts are automatically debited from his bank account.

When Scott pulls into his destination’s parking lot, his car bumps into one of several parking signs. As soon as the car stops moving, its internal diagnostics determine the extent of the damage. His personal assistant instructs him to take three pictures of the front right bumper area and two of the surroundings. By the time Scott gets back to the driver’s seat, the screen on the dash informs him of the damage, confirms the claim has been approved, and reports that a mobile response drone has been dispatched to the lot for inspection. If the vehicle is drivable, it may be directed to the nearest in-network garage for repair after a replacement vehicle arrives.

While this scenario may seem beyond the horizon, such integrated user stories will emerge across all lines of insurance with increasing frequency over the next decade. In fact, all the technologies required above already exist, and many are available to consumers. With the new wave of deep learning techniques, such as convolutional neural networks,1 artificial intelligence (AI) has the potential to live up to its promise of mimicking the perception, reasoning, learning, and problem solving of the human mind (Exhibit 1). In this evolution, insurance will shift from its current state of "detect and repair" to "predict and prevent," transforming every aspect of the industry in the process. The pace of change will also accelerate as brokers, consumers, financial intermediaries, insurers, and suppliers become more adept at using advanced technologies to enhance decision making and productivity, lower costs, and optimize the customer experience

AI for Life Insurance Claims

The technology world is transforming rapidly, which is why the insurance sector is shifting with it, driven by customer expectations. To be competitive, insurance companies require more customer insights and the capability to turn these insights into action.

AI in life insurance offers the chance to increase revenue, improve efficiency, and reduce risk. AI has the ability to improve mortality, optimize decision-making to help build long lasting profitable customer relationships.

Utilizing AI in life insurance underwriting can be determined along with individualized pricing, disease severity prediction, submission prioritization, and rapid product development.

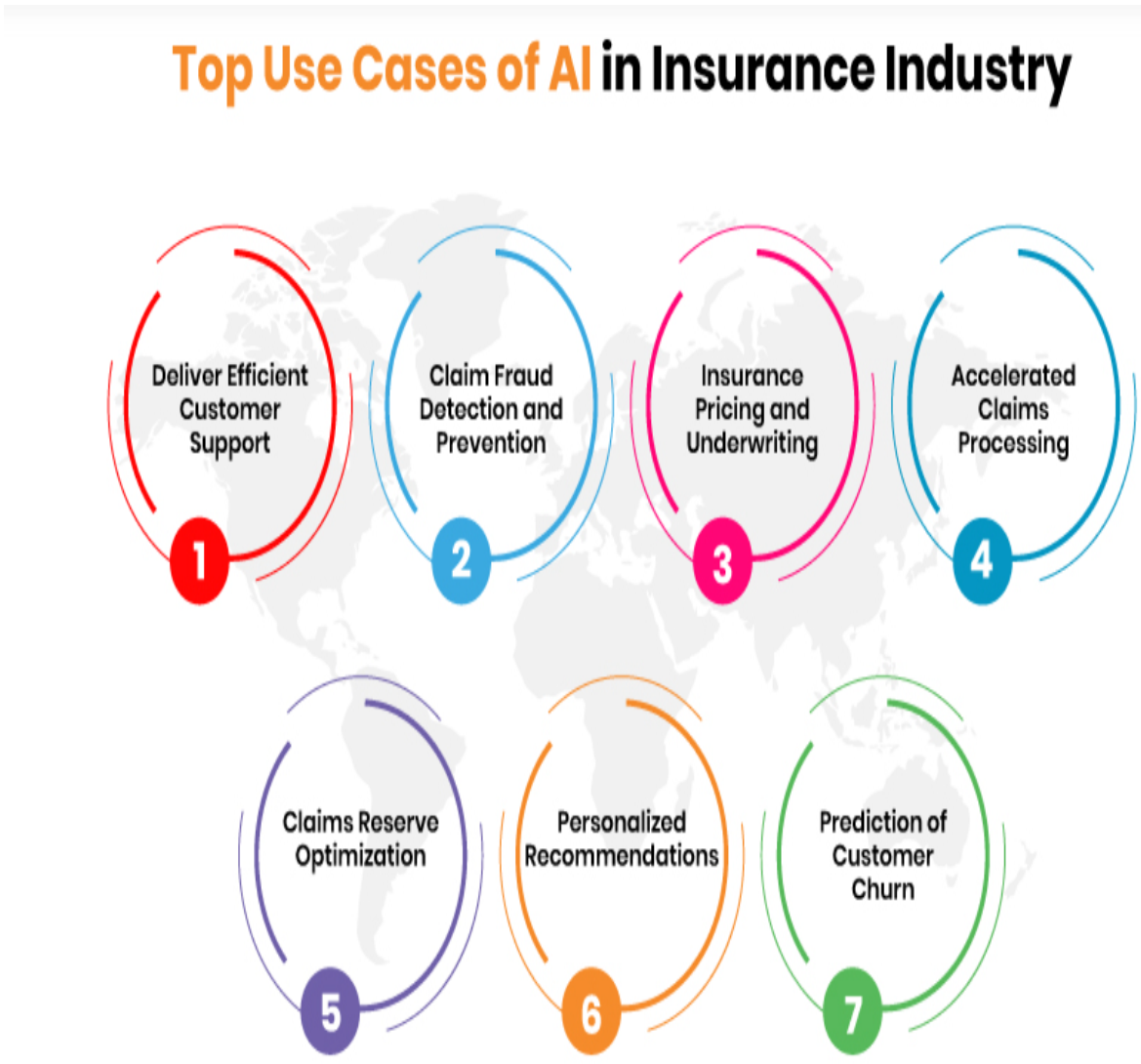
AI for Commercial Insurance

Artificial Intelligence has the ability that allows programmers to create software that can identify patterns, extrapolate predictions and manage information more quickly and efficiently.

AI can easily spot patterns and flag risks in commercial insurance. Both new and established insurance companies are embracing artificial intelligence for these abilities.

Here is how AI in Commercial Insurance is Vital

- **Better Data-Driven Decision-Making:** AI in the commercial insurance industry, makes it easier to manage the massive quantities of data generated by both insurers and their customers.
- **Faster Process:** The AI algorithms control the amount of time committed to underwriting tasks decreases significantly. All the data is easily processed, and it takes less time than the traditional processes.
- **Risk Evaluation:** Applying AI to underwriting questions can be done by providing a more accurate view of underlying risk by accounting for factors that may not otherwise reach an underwriter's attention.


Top Use Cases of AI in Insurance Industry
Deliver Efficient Customer Support
1
Claim Fraud Detection and Prevention
2
Insurance Pricing and Underwriting
3
Accelerated Claims Processing
4
Claims Reserve Optimization
5
Personalized Recommendations
6
Prediction of Customer Churn
7

13.APPENDIX

Source Code

GitHub & Project Demo Link


```
{
 "nbformat": 4,
 "nbformat_minor": 0,
 "metadata": {
  "colab": {
   "provenance": [],
   "collapsed_sections": []
  },
  "kernelspec": {
   "name": "python3",
   "display_name": "Python 3"
  },
  "language_info": {
   "name": "python"
  }
 },
 "cells": [
  {
   "cell_type": "markdown",
   "source": [
    "# Basic Python"
   ],
   "metadata": {
    "id": "McSxJAwcOdZ1"
   }
  },
  {
   "cell_type": "markdown",
   "source": [
    "## 1. Split this string"
```

```
    ],
"metadata": {
      "id": "CU48hgo4Owz5"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "s = \"Hi there Sam!\""
    ],
    "metadata": {
     "id": "s07c7JK7Oqt-"
    },
    "execution_count": 1,
    "outputs": []
   },
   {
    "cell_type": "code",
    "source": [
     "s"
    ],
    "metadata": {
     "id": "6mGVa3SQYLkb",
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 35
     },
     "outputId": "1d6bb41c-a6ff-4821-e6f9-4fbd4f9d5e71"
    },
    "execution_count": 2,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "'Hi there Sam!'"
       ],
```

```
          "application/vnd.google.colaboratory.intrinsic+json": {
"type": "string"
          }
        },
        "metadata": {},
        "execution_count": 2
      }
    ]
  },
  {
    "cell_type": "markdown",
    "source": [
      "## 2. Use .format() to print the following string. \n",
      "\n",
      "### Output should be: The diameter of Earth is 12742 kilometers."
    ],
    "metadata": {
      "id": "GH1QBn8HP375"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "planet = \"Earth\"\n",
      "diameter = 12742"
    ],
    "metadata": {
      "id": "_ZHoml3kPqic"
    },
    "execution_count": 3,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "print(\"The diameter of {} is {} kilometer\".format(planet,diameter\n",
      "                                          ))"
```

```
    ],
  "metadata": {
      "id": "HyRyJv6CYPb4",
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "outputId": "f1be3e4d-a5fa-47a2-c4c1-f01702b350e7"
    },
    "execution_count": 5,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "The diameter of Earth is 12742 kilometer\n"
        ]
      }
    ]
  },
  {
    "cell_type": "markdown",
    "source": [
      "## 3. In this nest dictionary grab the word \"hello\""
    ],
    "metadata": {
      "id": "KE74ZEwkRExZ"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
    ],
    "metadata": {
      "id": "fcVwbCc1QrQI"
    },
    "execution_count": 6,
```

```
      "outputs": []
  },
    {
     "cell_type": "code",
     "source": [
      "d['k1'][3]['tricky'][3]['target'][3]"
     ],
     "metadata": {
      "id": "MvbkMZpXYRaw",
      "colab": {
       "base_uri": "https://localhost:8080/",
       "height": 35
      },
      "outputId": "562192a8-ea84-44f7-c7ad-6bd08e511512"
     },
     "execution_count": 9,
     "outputs": [
      {
       "output_type": "execute_result",
       "data": {
        "text/plain": [
         "'hello'"
        ],
        "application/vnd.google.colaboratory.intrinsic+json": {
         "type": "string"
        }
       },
       "metadata": {},
       "execution_count": 9
      }
     ]
    },
    {
     "cell_type": "markdown",
     "source": [
      "# Numpy"
     ],
```

```
  "metadata": {
    "id": "bw0vVp-9ddjv"
  }
 },
 {
  "cell_type": "code",
  "source": [
    "import numpy as np"
  ],
  "metadata": {
    "id": "LLiE_TYrhA1O"
  },
  "execution_count": 10,
  "outputs": []
 },
 {
  "cell_type": "markdown",
  "source": [
    "## 4.1 Create an array of 10 zeros? \n",
    "## 4.2 Create an array of 10 fives?"
  ],
  "metadata": {
    "id": "wOg8hinbgx30"
  }
 },
 {
  "cell_type": "code",
  "source": [
    "a=np.zeros(10)\n",
    "a"
  ],
  "metadata": {
    "id": "NHrirmgCYXvU",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "8376576f-0921-4780-d856-f25f33ff720a"
```

```
    },

"execution_count": 12,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "array([0., 0., 0., 0., 0., 0., 0., 0., 0.])"
       ]
      },
      "metadata": {},
      "execution_count": 12
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "b=np.ones(10)*5\n",
     "b"
    ],
    "metadata": {
     "id": "e4005lsTYXxx",
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "outputId": "0131e0ec-100a-42f5-9867-ad4aab2013ec"
    },
    "execution_count": 13,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])"
       ]
```

```
      },
      "metadata": {},
      "execution_count": 13
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "## 5. Create an array of all the even integers from 20 to 35"
  ],
  "metadata": {
    "id": "gZHHDUBvrMX4"
  }
},
{
  "cell_type": "code",
  "source": [
    "s=np.arange(20,50,2)\n",
    "s"
  ],
  "metadata": {
    "id": "oAI2tbU2Yag-",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "09b9e929-d046-4017-95d9-15f41f514437"
  },
  "execution_count": 15,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48])"
        ]
      },
```

```
        "metadata": {},
        "execution_count": 15
      }
     ]
    },
    {
   "cell_type": "markdown",
     "source": [
      "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
     ],
     "metadata": {
      "id": "NaOM308NsRpZ"
     }
    },
    {
     "cell_type": "code",
     "source": [
      "b=np.arange(0,9).reshape(3,3)\n",
      "b"
     ],
     "metadata": {
      "id": "tOlEVH7BYceE",
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "outputId": "60c7326a-9633-4425-bb39-c062e828d15d"
     },
     "execution_count": 17,
     "outputs": [
      {
       "output_type": "execute_result",
       "data": {
        "text/plain": [
         "array([[0, 1, 2],\n",
         "       [3, 4, 5],\n",
         "       [6, 7, 8]])"
        ]
```

```
      },
      "metadata": {},
      "execution_count": 17
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "## 7. Concatenate a and b \n",
    "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
  ],
  "metadata": {
    "id": "hQ0dnhAQuU_p"
  }
},
{
  "cell_type": "code",
  "source": [
    "a=np.array([1,2,3])\n",
    "b=np.array([4,5,6])\n",
    "np.concatenate((a,b))"
  ],
  "metadata": {
    "id": "rAPSw97aYfE0",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "7a0cce13-2d3f-4a8c-b9a4-1c9aa60b1575"
  },
  "execution_count": 18,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([1, 2, 3, 4, 5, 6])"
```

```
        ]
       },
       "metadata": {},
       "execution_count": 18
     }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
     "# Pandas"
   ],
   "metadata": {
     "id": "dlPEY9DRwZga"
   }
  },
  {
   "cell_type": "markdown",
   "source": [
     "## 8. Create a dataframe with 3 rows and 2 columns"
   ],
   "metadata": {
     "id": "ijoYW51zwr87"
   }
  },
  {
   "cell_type": "code",
   "source": [
     "import pandas as pd\n"
   ],
   "metadata": {
     "id": "T5OxJRZ8uvR7"
   },
   "execution_count": 22,
   "outputs": []
  },
  {
```

```
"cell_type": "code",
"source": [
  "d = {\"names\":[\"aaa\",\"bbb\",\"ccc\",],\"age\":[21,22,20]}\n",
  "df = pd.DataFrame(d)\n",
  "df"
],
"metadata": {
  "id": "xNpI_XXoYhs0",
  "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 143
  },
  "outputId": "92862b6c-029d-4dff-9879-8732bed4335b"
},
"execution_count": 24,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "  names  age\n",
        "0   aaa   21\n",
        "1   bbb   22\n",
        "2   ccc   20"
      ],
      "text/html": [
        "\n",
        "  <div id=\"df-b5679877-3840-42e3-9a1b-a99a23e039fd\">\n",
        "    <div class=\"colab-df-container\">\n",
        "      <div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "        vertical-align: middle;\n",
        "    }\n",
        "\n",
        "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
```

```
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>names</th>\n",
"      <th>age</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>aaa</td>\n",
"      <td>21</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>bbb</td>\n",
"      <td>22</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>ccc</td>\n",
"      <td>20</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-
b5679877-3840-42e3-9a1b-a99a23e039fd')\"\n",
"              title=\"Convert this dataframe to an interactive table.\"\n",
"              style=\"display:none;\">\n",
```

```
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0
24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"    <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-
2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94
2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-
1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78
2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-
.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"     </button>\n",
"     \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64,
67, 0.15);\n",
```

```
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-b5679877-3840-42e3-9a1b-a99a23e039fd
button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-b5679877-3840-42e3-
9a1b-a99a23e039fd');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"            '<a target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data table
notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
```

```
          "         dataTable['output_type'] = 'display_data';\n",
          "         await google.colab.output.renderOutput(dataTable, element);\n",
          "         const docLink = document.createElement('div');\n",
          "         docLink.innerHTML = docLinkHtml;\n",
          "         element.appendChild(docLink);\n",
          "       }\n",
          "     </script>\n",
          "   </div>\n",
          "  </div>\n",
          "  "
        ]
      },
      "metadata": {},
      "execution_count": 24
    }
  ]
 },
 {
  "cell_type": "markdown",
  "source": [
   "## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
  ],
  "metadata": {
   "id": "UXSmdNclyJQD"
  }
 },
 {
  "cell_type": "code",
  "source": [
   "m= pd.date_range(start='1-01-2023',end='10-02-2023')\n",
   "for i in m:\n",
   "  print(i)"
  ],
  "metadata": {
   "id": "dgyC0JhVYl4F",
   "colab": {
    "base_uri": "https://localhost:8080/"
```

```
  },
  "outputId": "148bfe6d-e4ea-4796-fa1e-79ad4b21f7d9"
},
"execution_count": 25,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "2023-01-01 00:00:00\n",
      "2023-01-02 00:00:00\n",
      "2023-01-03 00:00:00\n",
      "2023-01-04 00:00:00\n",
      "2023-01-05 00:00:00\n",
      "2023-01-06 00:00:00\n",
      "2023-01-07 00:00:00\n",
      "2023-01-08 00:00:00\n",
      "2023-01-09 00:00:00\n",
      "2023-01-10 00:00:00\n",
      "2023-01-11 00:00:00\n",
      "2023-01-12 00:00:00\n",
      "2023-01-13 00:00:00\n",
      "2023-01-14 00:00:00\n",
      "2023-01-15 00:00:00\n",
      "2023-01-16 00:00:00\n",
      "2023-01-17 00:00:00\n",
      "2023-01-18 00:00:00\n",
      "2023-01-19 00:00:00\n",
      "2023-01-20 00:00:00\n",
      "2023-01-21 00:00:00\n",
      "2023-01-22 00:00:00\n",
      "2023-01-23 00:00:00\n",
      "2023-01-24 00:00:00\n",
      "2023-01-25 00:00:00\n",
      "2023-01-26 00:00:00\n",
      "2023-01-27 00:00:00\n",
      "2023-01-28 00:00:00\n",
```

"2023-01-29 00:00:00\n",
"2023-01-30 00:00:00\n",
"2023-01-31 00:00:00\n",
"2023-02-01 00:00:00\n",
"2023-02-02 00:00:00\n",
"2023-02-03 00:00:00\n",
"2023-02-04 00:00:00\n",
"2023-02-05 00:00:00\n",
"2023-02-06 00:00:00\n",
"2023-02-07 00:00:00\n",
"2023-02-08 00:00:00\n",
"2023-02-09 00:00:00\n",
"2023-02-10 00:00:00\n",
"2023-02-11 00:00:00\n",
"2023-02-12 00:00:00\n",
"2023-02-13 00:00:00\n",
"2023-02-14 00:00:00\n",
"2023-02-15 00:00:00\n",
"2023-02-16 00:00:00\n",
"2023-02-17 00:00:00\n",
"2023-02-18 00:00:00\n",
"2023-02-19 00:00:00\n",
"2023-02-20 00:00:00\n",
"2023-02-21 00:00:00\n",
"2023-02-22 00:00:00\n",
"2023-02-23 00:00:00\n",
"2023-02-24 00:00:00\n",
"2023-02-25 00:00:00\n",
"2023-02-26 00:00:00\n",
"2023-02-27 00:00:00\n",
"2023-02-28 00:00:00\n",
"2023-03-01 00:00:00\n",
"2023-03-02 00:00:00\n",
"2023-03-03 00:00:00\n",
"2023-03-04 00:00:00\n",
"2023-03-05 00:00:00\n",
"2023-03-06 00:00:00\n",

"2023-03-07 00:00:00\n",
"2023-03-08 00:00:00\n",
"2023-03-09 00:00:00\n",
"2023-03-10 00:00:00\n",
"2023-03-11 00:00:00\n",
"2023-03-12 00:00:00\n",
"2023-03-13 00:00:00\n",
"2023-03-14 00:00:00\n",
"2023-03-15 00:00:00\n",
"2023-03-16 00:00:00\n",
"2023-03-17 00:00:00\n",
"2023-03-18 00:00:00\n",
"2023-03-19 00:00:00\n",
"2023-03-20 00:00:00\n",
"2023-03-21 00:00:00\n",
"2023-03-22 00:00:00\n",
"2023-03-23 00:00:00\n",
"2023-03-24 00:00:00\n",
"2023-03-25 00:00:00\n",
"2023-03-26 00:00:00\n",
"2023-03-27 00:00:00\n",
"2023-03-28 00:00:00\n",
"2023-03-29 00:00:00\n",
"2023-03-30 00:00:00\n",
"2023-03-31 00:00:00\n",
"2023-04-01 00:00:00\n",
"2023-04-02 00:00:00\n",
"2023-04-03 00:00:00\n",
"2023-04-04 00:00:00\n",
"2023-04-05 00:00:00\n",
"2023-04-06 00:00:00\n",
"2023-04-07 00:00:00\n",
"2023-04-08 00:00:00\n",
"2023-04-09 00:00:00\n",
"2023-04-10 00:00:00\n",
"2023-04-11 00:00:00\n",
"2023-04-12 00:00:00\n",

"2023-04-13 00:00:00\n",
"2023-04-14 00:00:00\n",
"2023-04-15 00:00:00\n",
"2023-04-16 00:00:00\n",
"2023-04-17 00:00:00\n",
"2023-04-18 00:00:00\n",
"2023-04-19 00:00:00\n",
"2023-04-20 00:00:00\n",
"2023-04-21 00:00:00\n",
"2023-04-22 00:00:00\n",
"2023-04-23 00:00:00\n",
"2023-04-24 00:00:00\n",
"2023-04-25 00:00:00\n",
"2023-04-26 00:00:00\n",
"2023-04-27 00:00:00\n",
"2023-04-28 00:00:00\n",
"2023-04-29 00:00:00\n",
"2023-04-30 00:00:00\n",
"2023-05-01 00:00:00\n",
"2023-05-02 00:00:00\n",
"2023-05-03 00:00:00\n",
"2023-05-04 00:00:00\n",
"2023-05-05 00:00:00\n",
"2023-05-06 00:00:00\n",
"2023-05-07 00:00:00\n",
"2023-05-08 00:00:00\n",
"2023-05-09 00:00:00\n",
"2023-05-10 00:00:00\n",
"2023-05-11 00:00:00\n",
"2023-05-12 00:00:00\n",
"2023-05-13 00:00:00\n",
"2023-05-14 00:00:00\n",
"2023-05-15 00:00:00\n",
"2023-05-16 00:00:00\n",
"2023-05-17 00:00:00\n",
"2023-05-18 00:00:00\n",
"2023-05-19 00:00:00\n",

"2023-05-20 00:00:00\n",
"2023-05-21 00:00:00\n",
"2023-05-22 00:00:00\n",
"2023-05-23 00:00:00\n",
"2023-05-24 00:00:00\n",
"2023-05-25 00:00:00\n",
"2023-05-26 00:00:00\n",
"2023-05-27 00:00:00\n",
"2023-05-28 00:00:00\n",
"2023-05-29 00:00:00\n",
"2023-05-30 00:00:00\n",
"2023-05-31 00:00:00\n",
"2023-06-01 00:00:00\n",
"2023-06-02 00:00:00\n",
"2023-06-03 00:00:00\n",
"2023-06-04 00:00:00\n",
"2023-06-05 00:00:00\n",
"2023-06-06 00:00:00\n",
"2023-06-07 00:00:00\n",
"2023-06-08 00:00:00\n",
"2023-06-09 00:00:00\n",
"2023-06-10 00:00:00\n",
"2023-06-11 00:00:00\n",
"2023-06-12 00:00:00\n",
"2023-06-13 00:00:00\n",
"2023-06-14 00:00:00\n",
"2023-06-15 00:00:00\n",
"2023-06-16 00:00:00\n",
"2023-06-17 00:00:00\n",
"2023-06-18 00:00:00\n",
"2023-06-19 00:00:00\n",
"2023-06-20 00:00:00\n",
"2023-06-21 00:00:00\n",
"2023-06-22 00:00:00\n",
"2023-06-23 00:00:00\n",
"2023-06-24 00:00:00\n",
"2023-06-25 00:00:00\n",

"2023-06-26 00:00:00\n",
"2023-06-27 00:00:00\n",
"2023-06-28 00:00:00\n",
"2023-06-29 00:00:00\n",
"2023-06-30 00:00:00\n",
"2023-07-01 00:00:00\n",
"2023-07-02 00:00:00\n",
"2023-07-03 00:00:00\n",
"2023-07-04 00:00:00\n",
"2023-07-05 00:00:00\n",
"2023-07-06 00:00:00\n",
"2023-07-07 00:00:00\n",
"2023-07-08 00:00:00\n",
"2023-07-09 00:00:00\n",
"2023-07-10 00:00:00\n",
"2023-07-11 00:00:00\n",
"2023-07-12 00:00:00\n",
"2023-07-13 00:00:00\n",
"2023-07-14 00:00:00\n",
"2023-07-15 00:00:00\n",
"2023-07-16 00:00:00\n",
"2023-07-17 00:00:00\n",
"2023-07-18 00:00:00\n",
"2023-07-19 00:00:00\n",
"2023-07-20 00:00:00\n",
"2023-07-21 00:00:00\n",
"2023-07-22 00:00:00\n",
"2023-07-23 00:00:00\n",
"2023-07-24 00:00:00\n",
"2023-07-25 00:00:00\n",
"2023-07-26 00:00:00\n",
"2023-07-27 00:00:00\n",
"2023-07-28 00:00:00\n",
"2023-07-29 00:00:00\n",
"2023-07-30 00:00:00\n",
"2023-07-31 00:00:00\n",
"2023-08-01 00:00:00\n",

"2023-08-02 00:00:00\n",
"2023-08-03 00:00:00\n",
"2023-08-04 00:00:00\n",
"2023-08-05 00:00:00\n",
"2023-08-06 00:00:00\n",
"2023-08-07 00:00:00\n",
"2023-08-08 00:00:00\n",
"2023-08-09 00:00:00\n",
"2023-08-10 00:00:00\n",
"2023-08-11 00:00:00\n",
"2023-08-12 00:00:00\n",
"2023-08-13 00:00:00\n",
"2023-08-14 00:00:00\n",
"2023-08-15 00:00:00\n",
"2023-08-16 00:00:00\n",
"2023-08-17 00:00:00\n",
"2023-08-18 00:00:00\n",
"2023-08-19 00:00:00\n",
"2023-08-20 00:00:00\n",
"2023-08-21 00:00:00\n",
"2023-08-22 00:00:00\n",
"2023-08-23 00:00:00\n",
"2023-08-24 00:00:00\n",
"2023-08-25 00:00:00\n",
"2023-08-26 00:00:00\n",
"2023-08-27 00:00:00\n",
"2023-08-28 00:00:00\n",
"2023-08-29 00:00:00\n",
"2023-08-30 00:00:00\n",
"2023-08-31 00:00:00\n",
"2023-09-01 00:00:00\n",
"2023-09-02 00:00:00\n",
"2023-09-03 00:00:00\n",
"2023-09-04 00:00:00\n",
"2023-09-05 00:00:00\n",
"2023-09-06 00:00:00\n",
"2023-09-07 00:00:00\n",

```
        "2023-09-08 00:00:00\n",
        "2023-09-09 00:00:00\n",
        "2023-09-10 00:00:00\n",
        "2023-09-11 00:00:00\n",
        "2023-09-12 00:00:00\n",
        "2023-09-13 00:00:00\n",
        "2023-09-14 00:00:00\n",
        "2023-09-15 00:00:00\n",
        "2023-09-16 00:00:00\n",
        "2023-09-17 00:00:00\n",
        "2023-09-18 00:00:00\n",
        "2023-09-19 00:00:00\n",
        "2023-09-20 00:00:00\n",
        "2023-09-21 00:00:00\n",
        "2023-09-22 00:00:00\n",
        "2023-09-23 00:00:00\n",
        "2023-09-24 00:00:00\n",
        "2023-09-25 00:00:00\n",
        "2023-09-26 00:00:00\n",
        "2023-09-27 00:00:00\n",
        "2023-09-28 00:00:00\n",
        "2023-09-29 00:00:00\n",
        "2023-09-30 00:00:00\n",
        "2023-10-01 00:00:00\n",
        "2023-10-02 00:00:00\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "## 10. Create 2D list to DataFrame\n",
    "\n",
    "lists = [[1, 'aaa', 22],\n",
    ""        [2, 'bbb', 25],\n",
    ""        [3, 'ccc', 24]]"
```

```
  ],
  "metadata": {
    "id": "ZizSetD-y5az"
  }
 },
 {
  "cell_type": "code",
  "source": [
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
  ],
  "metadata": {
    "id": "_XMC8aEt0llB"
  },
  "execution_count": 23,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
    "print(pd.DataFrame(lists))"
  ],
  "metadata": {
    "id": "knH76sDKYsVX",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "b7cf97e6-eaa3-4390-ec48-29f2d6aa92b7"
  },
  "execution_count": 28,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "   0    1   2\n",
        "0  1  aaa  22\n",
        "1  2  bbb  25\n",
```

```
        "2  3  ccc  24\n"
       ]
      }
     ]
    }
   ]
}
```


Github link :
https://github.com/IBM-EPBL/IBM-Project-39520-1660454186

Demo Video Link: