Simplifying Insurance Claims through Automation Using Intelligent Agents

Mohammad Shehnaj
x23305762
MSCAI1
x23305762@student.ncirl.ie

Introduction

Insurance firms process thousands of claims monthly (health, motor, travel)

Manual claim handling is error-prone, slow and repetitive.

This project automates the eligibility check using Python, UiPath and machine learning.

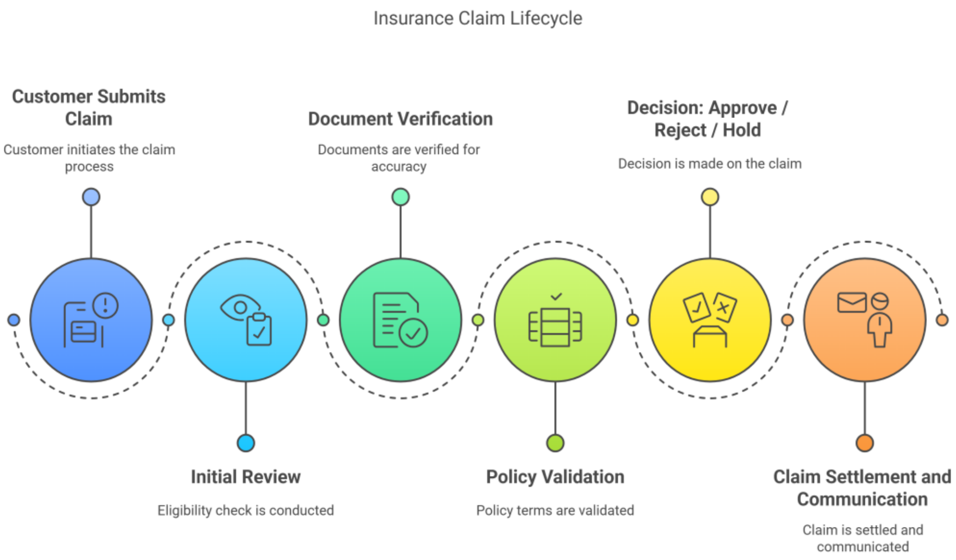
Goal: Improve accuracy, reduce delays and improve operational scalability.

Business Process Identification

Process: Insurance Claim Handling

Challenges: Repetitive, error-prone steps

Automation Scope: Document verification, eligibility checks, email updates



Insurance Claim Lifecycle

Customer Submits Claim

Customer initiates the claim process

Initial Review

Eligibility check is conducted

Document Verification

Documents are verified for accuracy

Policy Validation

Policy terms are validated

Decision: Approve / Reject / Hold

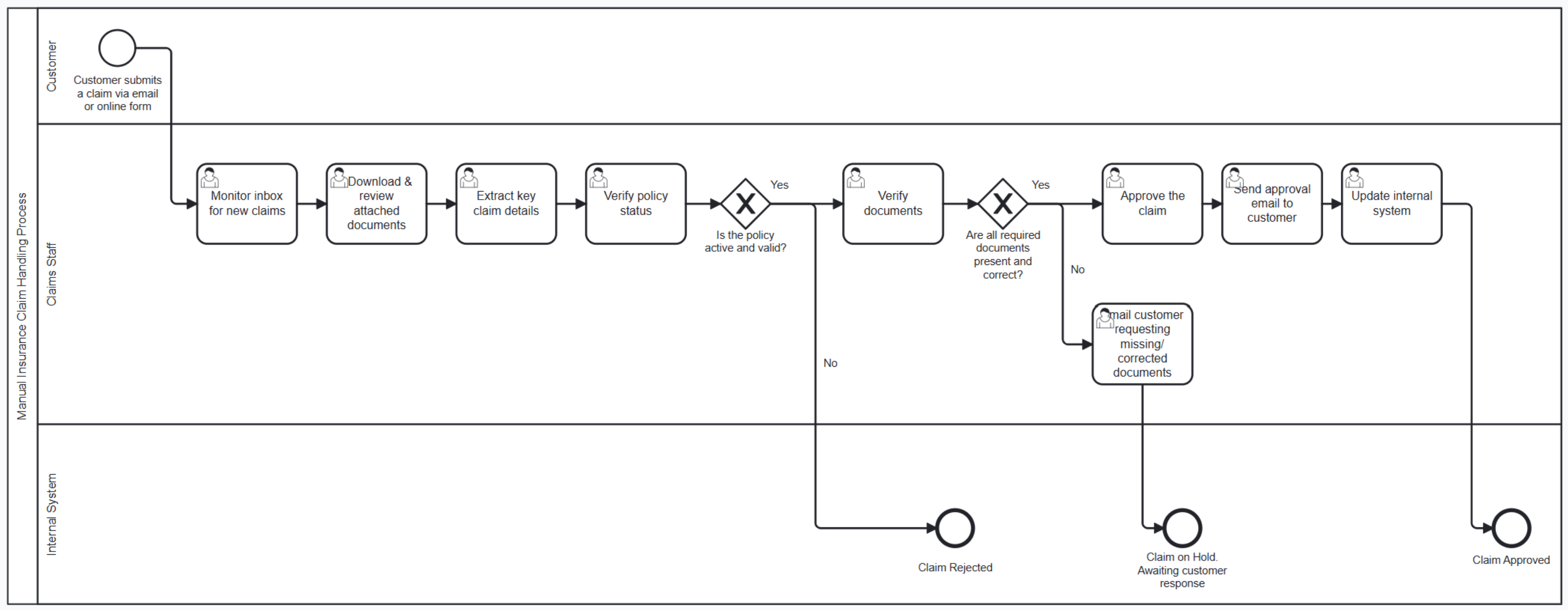
Decision is made on the claim

Claim Settlement and Communication

Claim is settled and communicated

As-Is Process Model (Manual Process)

- Manual tasks: check emails, download & verify attachments, validate policy.
- Decisions and emails handled manually.
- Issues Identified:
  - ✓ High workload, risk of error, delays
  - ✓ Low productivity during claim peak periods


Manual Insurance Claim Handling Process
Customer
Claims Staff
Internal System
Customer submits a claim via email or online form
Monitor inbox for new claims
Download & review attached documents
Extract key claim details
Verify policy status
Is the policy active and valid?
Yes
No
Verify documents
Are all required documents present and correct?
Yes
No
Approve the claim
Send approval email to customer
Update internal system
Email customer requesting missing/ corrected documents
Claim Rejected
Claim on Hold. Awaiting customer response
Claim Approved

As-Is Process Flow for Manual Insurance Claim Handling

Current Process Analysis

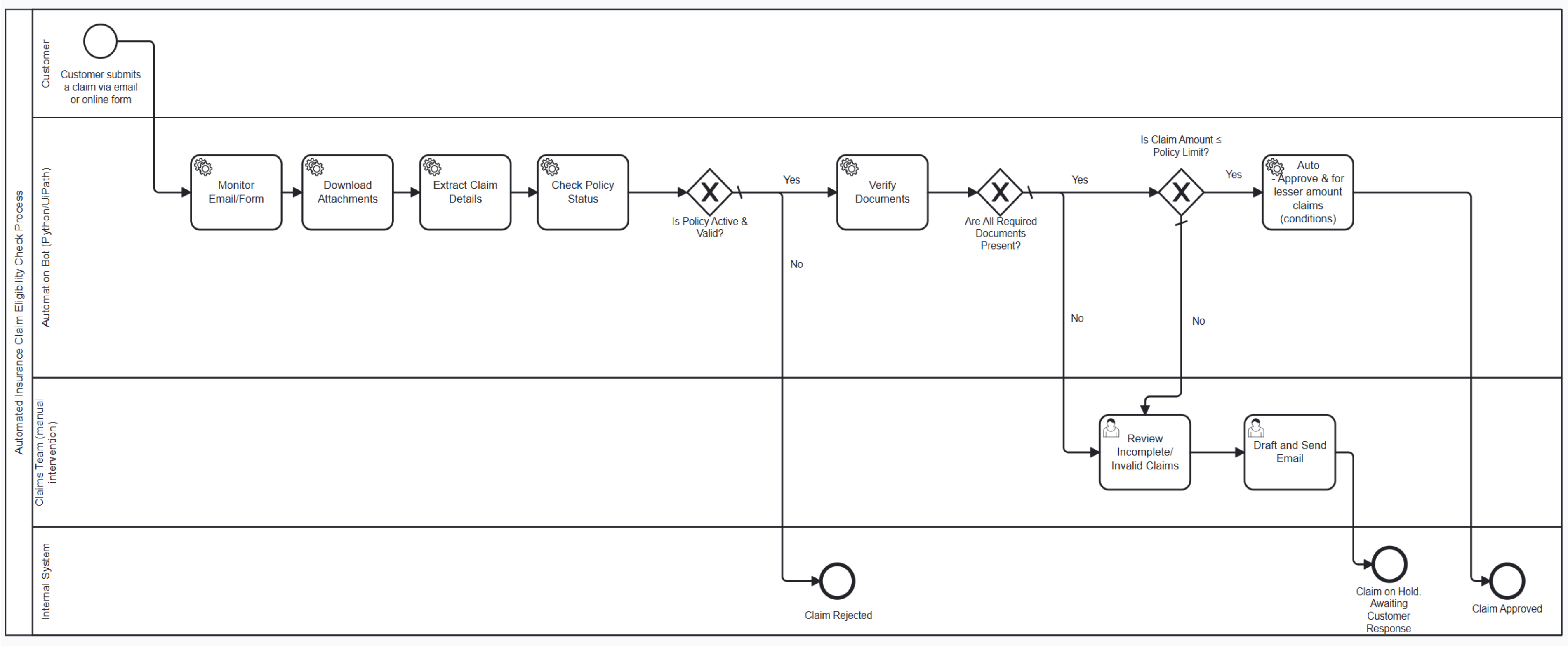
High human effort in low-value tasks

Delays and backlogs during peak times

Errors due to manual data entry and document review

To-Be Process Model (Automated Flow)

- Bots read claims, extract data, validate policies and documents.
- Simple rule engine used for automated approvals or escalations.
- UiPath and Python used for execution.


Automated Insurance Claim Eligibility Check Process
Customer
Customer submits a claim via email or online form
Automation Bot (Python/UiPath)
Monitor Email/Form
Download Attachments
Extract Claim Details
Check Policy Status
Is Policy Active & Valid?
Yes
No
Verify Documents
Are All Required Documents Present?
Yes
No
Is Claim Amount ≤ Policy Limit?
Yes
No
Auto - Approve & for lesser amount claims (conditions)
Claims Team (manual intervention)
Review Incomplete/ Invalid Claims
Draft and Send Email
Internal System
Claim Rejected
Claim on Hold. Awaiting Customer Response
Claim Approved

To-Be Automated Process Flow for Claim Handling

Automation Potential Analysis

Objective: Identify which tasks in the insurance claim process can be automated.

Analysis Summary:

- ➢ Over 80% of tasks in the claim handling process are suitable for automation.
- ➢ Tasks with high automation feasibility:
  - ✓ **Email Inbox Monitoring:** Can be done using Python IMAP or UiPath email activities.
  - ✓ **Document Downloading :** Auto-download from email using UiPath or Python.
  - ✓ **Document Parsing:** Use OCR to extract claim information.
  - ✓ **Policy Verification:** Match policy data via scripts or UiPath filters.
  - ✓ **Document Completeness Check:** Compare required docs list with uploaded files.
  - ✓ **Eligibility Decision:** Apply rules in Python/ UiPath or ML models.
  - ✓ **Status Notification:** Send updates using automated email logic.
  - ✓ **Audit Logging:** Store claim decisions with timestamps.

Automation Proposal & Strategy

Automation Strategy:

- Use either Python for backend logic or UiPath for user-friendly automation workflows.

Tools Used:

- **Python:** Eligibility rules, audit logging, document checks
- **UiPath:** File input/output, process flow, email automation
- **ML Models:** Logistic Regression, Decision Tree, Random Forest

Approach:

- Apply rules to check:
  - □ Policy status
  - □ Document completeness
  - □ Claim amount eligibility
- ML used to predict complex cases

Expected Benefits:

- Faster processing & reduced manual effort
- Enhanced accuracy with ML support
- Automated status notifications and audit trail

Solution Demonstration – Claim Eligibility Check

Automated Task:

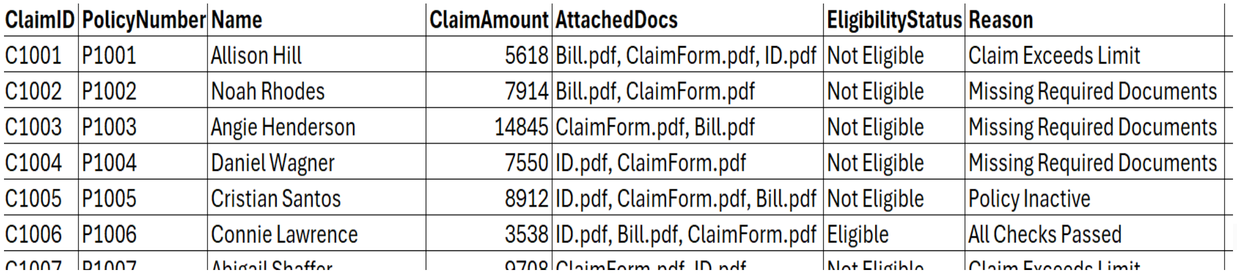
Eligibility Check: One of the most repetitive steps in insurance claim processing.

Python Automation:

- ✓ Reads claim and policy CSV files.
- ✓ Checks if policy is active and valid.
- ✓ Verifies claim amount is within limit.
- ✓ Confirms presence of required documents.
- ✓ Logs results with reason.

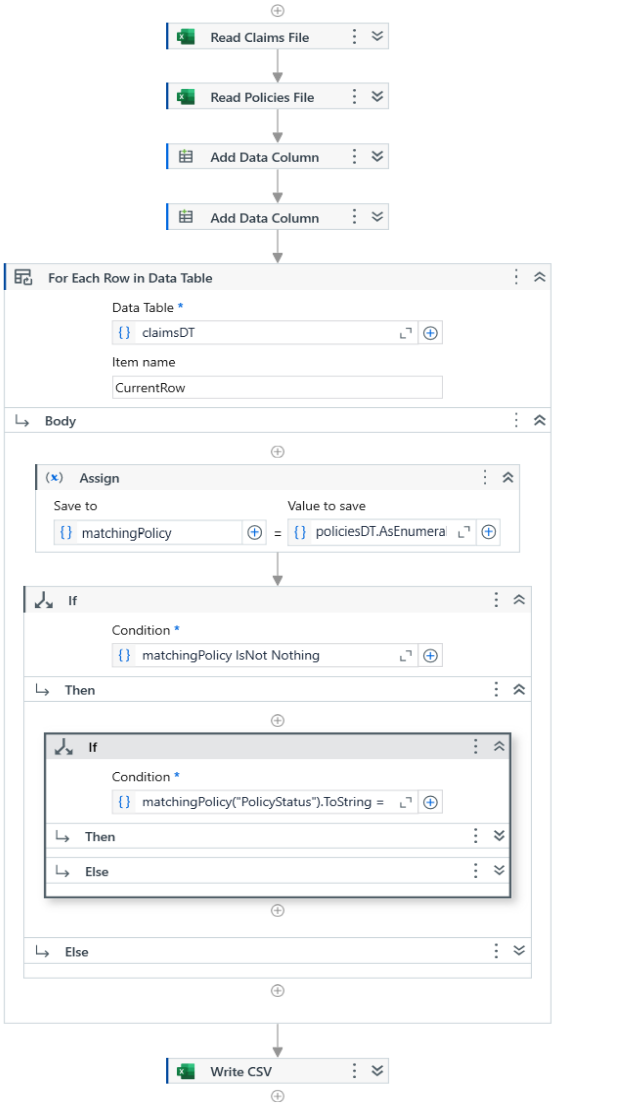
UiPath Workflow:

- ✓ Allows user to upload claim and policy files via interface.
- ✓ Reads the data and applies the business rules from built-in activities and UiPath logic blocks.
- ✓ Saves eligibility output as new CSV.



ClaimID	PolicyNumber	Name	ClaimAmount	AttachedDocs	EligibilityStatus	Reason
C1001	P1001	Allison Hill	5618	Bill.pdf, ClaimForm.pdf, ID.pdf	Not Eligible	Claim Exceeds Limit
C1002	P1002	Noah Rhodes	7914	Bill.pdf, ClaimForm.pdf	Not Eligible	Missing Required Documents
C1003	P1003	Angie Henderson	14845	ClaimForm.pdf, Bill.pdf	Not Eligible	Missing Required Documents
C1004	P1004	Daniel Wagner	7550	ID.pdf, ClaimForm.pdf	Not Eligible	Missing Required Documents
C1005	P1005	Cristian Santos	8912	ID.pdf, ClaimForm.pdf, Bill.pdf	Not Eligible	Policy Inactive
C1006	P1006	Connie Lawrence	3538	ID.pdf, Bill.pdf, ClaimForm.pdf	Eligible	All Checks Passed
C1007	P1007	Abigail Shaffer	9708	ClaimForm.pdf, ID.pdf	Not Eligible	Claim Exceeds Limit

Sample Claim Eligibility Results Table



Read Claims File

Read Policies File

Add Data Column

Add Data Column

For Each Row in Data Table

Data Table * claimsDT

Item name CurrentRow

Body

Assign

Save to matchingPolicy = Value to save policiesDT.AsEnumera

If

Condition * matchingPolicy IsNot Nothing

Then

If

Condition * matchingPolicy("PolicyStatus").ToString =

Then

Else

Else

Write CSV

UiPath loop for applying eligibility checks on each claim

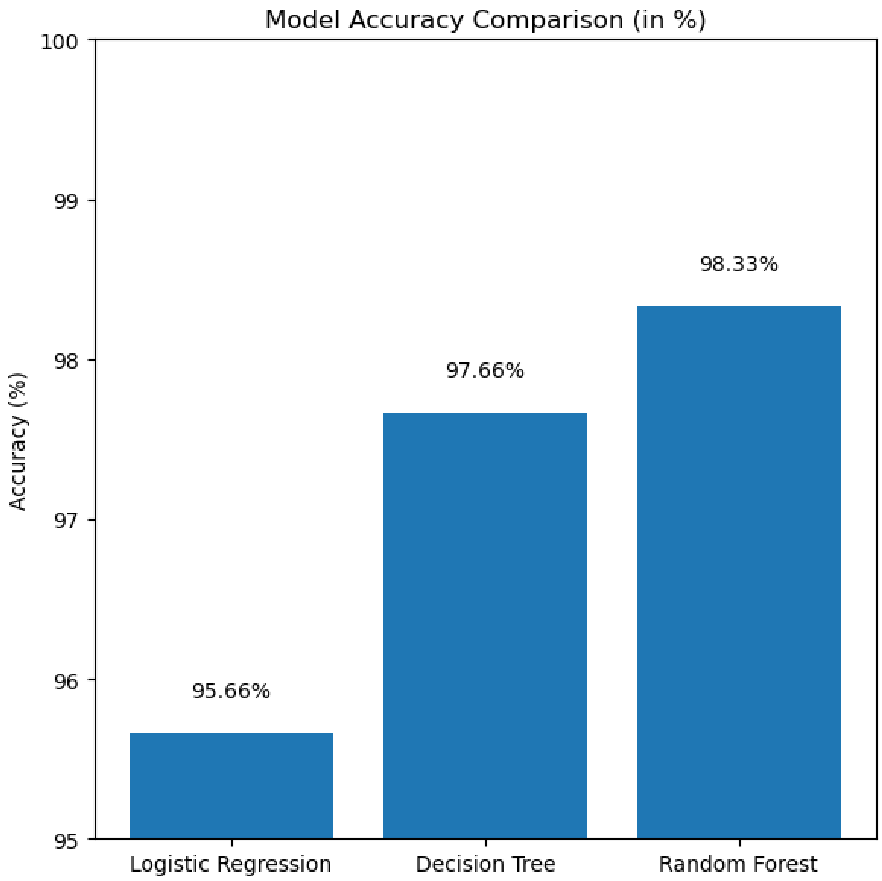
Results and Insights

ML models tested are Logistic Regression, Decision Tree, Random Forest

Accuracy chart of all three models is shown in bar graph

Best model is Random Forest (98.3%)

Impact of ML is it adds confidence to eligibility checks



Accuracy Comparison of ML Models

Conclusion

Intelligent automation reduces manual effort, improves speed and accuracy

Blending rule-based + AI-based logic adds flexibility

Future scope is to extend automation to fraud detection, chatbot queries, unstructured document parsing

Thank you!