|  |  |
| --- | --- |
| **Project Title** | **AI-Powered Intelligent Insurance Risk Assessment and Customer Insights System** |
| **Skills take away From This Project** | **Python (TensorFlow, PyTorch, Scikit-Learn, Hugging Face Transformers)**  **SQL/NoSQL Databases for storing policyholder data**  **AWS/Azure/GCP for cloud deployment**  **Streamlit/Flask for interactive UI** |
| **Domain** | **Insurance** |

**Problem Statement:**

Insurance companies face challenges in assessing risk, predicting claim amounts, detecting fraudulent claims, and understanding customer sentiments. Traditional methods rely heavily on manual evaluation, leading to inefficiencies and biased decision-making. With the advancement of AI and machine learning, automated risk assessment and customer analysis can improve efficiency, accuracy, and customer experience.

This project aims to build a comprehensive AI-powered system integrating various machine learning and deep learning techniques to optimize insurance processes.

#### **Objectives:**

1. **Risk Classification &** **Claim Prediction (Supervised Learning - Classification & Regression)**
   * Predict whether an insurance claim is **high-risk or low-risk** based on customer profiles, medical history, claim history, and policy details (Classification).
   * Predict the **expected claim amount** using regression models based on historical data.
2. **Customer Segmentation (Unsupervised Learning - Clustering)**
   * Segment policyholders into different categories (e.g., high-risk, low-risk, young professionals, elderly customers) using clustering techniques to tailor personalized insurance plans.
3. **Fraud Detection (Associative Learning & Anomaly Detection)**
   * Identify fraudulent claims by detecting unusual patterns using association rule mining and anomaly detection techniques.
4. **Feature Engineering & Dimension Reduction (****PCA, t-SNE, Autoencoders)**
   * Reduce the dimensionality of high-dimensional customer data for better model performance and explainability.
5. **Automated Multilingual Insurance Document Translation (****Hugging Face Transformers)**
   * Build a **translator** to convert insurance policies, terms & conditions, and claim documents into multiple languages using transformers.
6. **Sentiment Analysis of Customer Reviews & Complaints**
   * Analyze customer sentiments from feedback, emails, and chat support to detect **dissatisfaction trends and areas of service improvement**.
7. **Text Summarization for Insurance Policies (****Abstractive & Extractive Summarization)**
   * Summarize lengthy insurance policy documents into **concise and understandable formats** using NLP techniques.
8. **Text Generation for Automated Insurance Responses**
   * Develop a **chatbot or AI assistant** using **Hugging Face Transformers** to generate automated yet personalized responses for customer queries.

**Business Use Cases:**

Insurance companies face multiple challenges in assessing customer risk, predicting claims, identifying fraudulent activities, and enhancing customer experience. The traditional manual processes are slow, error-prone, and unable to leverage vast amounts of historical data efficiently.

Key pain points include:

* **Risk Assessment Complexity** – Difficulty in accurately classifying high-risk vs. low-risk policyholders.
* **Fraudulent Claims** – Rising financial losses due to undetected fraudulent claims.
* **Inefficient Claims Processing** – Slow and manual claim evaluation process leads to delayed settlements.
* **Lack of Personalization** – Inability to tailor insurance policies to customer needs.
* **Multilingual Barriers** – Serving diverse customer bases with language constraints.
* **Poor Customer Experience** – Customers struggle with lengthy documents and slow responses to queries.

**Approach:**

✅ **Data Collection & Preprocessing:** Gather insurance claim data, policyholder profiles, and feedback.

✅ **Model Training & Optimization:** Use Supervised (Classification & Regression), Unsupervised (Clustering), and NLP models.

✅ **Deployment & Integration:** Integrate with existing CRM, claim processing, and customer service platforms.

✅ **Monitoring & Continuous Improvement:** Use real-time dashboards to track model performance and make improvements.

**Results:**

📈 **Revenue Growth:**

* Personalized insurance pricing increases customer acquisition and retention by **15-20%**.
* Faster claims processing **reduces customer churn by 25%**.

📉 **Cost Reduction:**

* AI-powered fraud detection saves **millions in fraudulent payouts**.
* Automated chatbots **reduce customer support costs by 30-40%**.

⚡ **Operational Efficiency:**

* Claims settlement time reduced from **weeks to hours**.
* 24/7 multilingual customer support without human intervention.

🔍 **Improved Customer Experience:**

* Simplified policy documents & faster responses increase **customer trust & loyalty**.

**Project Evaluation metrics:**

## **1️⃣ Model Performance Metrics (AI & ML Effectiveness)**

### **🟢 Classification & Regression Models (Risk Classification & Claim Prediction)**

✅ **Accuracy** – Measures how well the model correctly classifies risk categories.  
✅ **Precision & Recall (F1-score)** – Balances false positives and false negatives in fraud detection and risk classification.  
✅ **Mean Absolute Error (MAE) & Root Mean Square Error (RMSE)** – Evaluates the accuracy of claim amount predictions.  
✅ **AUC-ROC Score** – Measures the model's ability to distinguish between fraudulent and genuine claims.

### **🟢 Clustering Models (Customer Segmentation)**

✅ **Silhouette Score** – Measures how well clusters are formed and separated.  
✅ **Davies-Bouldin Index** – Lower values indicate better clustering quality.

### **🟢 Fraud Detection Models (Anomaly Detection & Association Rules)**

✅ **False Positive Rate (FPR) & False Negative Rate (FNR)** – Ensures fraudulent claims are caught while minimizing false alarms.  
✅ **Precision-Recall Curve** – Evaluates fraud detection effectiveness.

### **🟢 NLP Models (Sentiment Analysis, Translation, Summarization, Chatbot)**

✅ **BLEU Score & ROUGE Score** – Measures translation and text summarization accuracy.  
✅ **Sentiment Analysis Accuracy** – Evaluates correctness in detecting customer sentiment (Positive, Negative, Neutral).  
✅ **Chatbot Response Time & Relevance Score** – Measures chatbot effectiveness in resolving queries.

## **2️⃣ Business Impact Metrics (Effectiveness in Real-world Scenarios)**

### **🟢 Fraud Detection & Prevention**

✅ **Reduction in Fraudulent Payouts (%)** – Percentage decrease in fraud cases after model deployment.  
✅ **False Claim Detection Rate (%)** – Measures effectiveness in flagging fraudulent claims.

### **🟢 Risk-Based Underwriting Efficiency**

✅ **Reduction in Manual Risk Assessment Time (⏳)** – Time saved per customer policy assessment.  
✅ **Increase in Correct Policy Pricing (%)** – Percentage improvement in risk-based policy premium recommendations.

### **🟢 Customer Segmentation & Personalization**

✅ **Increase in Policy Conversions (%)** – More personalized plans leading to higher sales.  
✅ **Customer Retention Rate (%)** – Better segmentation leads to long-term policyholder engagement.

## **3️⃣ Operational Efficiency Metrics (Process Optimization)**

### **🟢 Claim Processing Efficiency**

✅ **Reduction in Claims Processing Time (⏳)** – Faster settlements reduce delays and improve customer satisfaction.  
✅ **Increase in Auto-Processed Claims (%)** – Percentage of claims processed without manual intervention.

### **🟢 Policy Document Processing & Accessibility**

✅ **Multilingual Translation Accuracy (%)** – Ensures translated documents retain meaning.  
✅ **Summarization Compression Ratio** – Measures reduction in document length while retaining key information.

### **🟢 Chatbot & AI-Driven Customer Support**

✅ **Customer Query Resolution Rate (%)** – Measures AI chatbot effectiveness.  
✅ **Reduction in Support Call Volumes (%)** – Indicates how many queries are handled automatically.

## **4️⃣ Customer Experience Metrics (User Satisfaction & Engagement)**

✅ **Net Promoter Score (NPS)** – Measures overall customer satisfaction with AI-powered services.  
✅ **Customer Satisfaction Score (CSAT)** – Captures user feedback on AI chatbot interactions, claims processing, and personalized policies.  
✅ **User Engagement Metrics (Chatbot Conversations, Policy Page Views, Feedback Ratings, etc.)** – Monitors customer interactions with AI tools.  
✅ **Reduction in Customer Complaints (%)** – Fewer complaints indicate improved service quality.

## **5️⃣ Financial Metrics (Revenue & Cost Impact)**

✅ **Increase in Revenue from Policy Sales (%)** – Direct impact on insurance product sales due to AI-driven personalization.  
✅ **Cost Savings from Fraud Detection ($)** – Amount saved by preventing fraudulent claims.  
✅ **Reduction in Operational Costs ($)** – Lower expenses due to automation in claim processing and customer support.  
✅ **ROI (Return on Investment) (%)** – Evaluates financial benefits vs. AI implementation costs.

### **Final Success Indicators:**

✅ **Achieving a 90%+ accuracy rate** in risk classification and fraud detection.  
✅ **Reducing claims processing time by 40-60%** through AI automation.  
✅ **Detecting at least 70% fraudulent claims** with anomaly detection models.  
✅ **Increasing customer satisfaction (CSAT) by 25%** through multilingual support and automated chatbot services.  
✅ **Reducing customer churn by 20-30%** due to personalized policy recommendations.  
✅ **Generating at least 3X ROI** on AI implementation within the first year.

**Technical Tags:**

#### **Machine Learning & Deep Learning**

**✅ Supervised Learning  
✅ Unsupervised Learning  
✅ Classification (Risk Assessment, Fraud Detection)  
✅ Regression (Claim Prediction, Premium Calculation)  
✅ Clustering (Customer Segmentation, Market Analysis)  
✅ Anomaly Detection (Fraudulent Claim Detection, Outlier Analysis)  
✅ Association Rule Learning (Pattern Detection in Fraud & Customer Behavior)  
✅ Dimensionality Reduction (PCA, t-SNE, Autoencoders)**

#### **Natural Language Processing (NLP) & Generative AI**

**✅ Sentiment Analysis (Customer Feedback Analysis, Complaint Resolution)  
✅ Machine Translation (Multilingual Insurance Documents & Chatbot Support)  
✅ Text Summarization (Insurance Policy Summarization, Claim Document Review)  
✅ Text Generation (Automated Chatbot for Claim Processing & Policy Assistance)  
✅ Named Entity Recognition (NER) (Extracting Policyholder Details, Claims Information)  
✅ Hugging Face Transformers (BERT, GPT, T5, LLama, Falcon, etc.)  
✅ Large Language Models (LLMs) for Insurance Automation**

#### **Data Engineering & Big Data**

**✅ ETL (Extract, Transform, Load) for Insurance Data Pipelines  
✅ Feature Engineering for Risk Modeling  
✅ Time Series Analysis for Claim Forecasting  
✅ NoSQL Databases (MongoDB, Cassandra) for Policyholder Records  
✅ SQL for Structured Insurance Data (PostgreSQL, MySQL, Snowflake, BigQuery)  
✅ Cloud Data Warehousing (AWS Redshift, Snowflake, Google BigQuery)  
✅ Stream Processing for Real-Time Fraud Detection (Apache Kafka, Spark Streaming)**

#### **AI Model Deployment & MLOps**

**✅ Model Deployment using Flask/FastAPI for API-based Insurance Systems  
✅ AWS SageMaker / Azure ML / Google Vertex AI for AI Model Hosting  
✅ Docker & Kubernetes for Scalable AI Model Deployment  
✅ CI/CD Pipelines for ML (GitHub Actions, MLflow, Kubeflow)  
✅ Model Monitoring & Drift Detection**

#### **Insurance-Specific AI Use Cases**

**✅ Automated Underwriting & Risk Assessment  
✅ Predictive Claim Analytics  
✅ Insurance Chatbot for Customer Support  
✅ AI-powered Insurance Fraud Detection  
✅ Personalized Insurance Recommendation Engine  
✅ Claims Document Processing with NLP**

**Data Set:**

### **Insurance Risk & Claim Dataset**

📌 **Source:** Kaggle, OpenML, Government Open Data Repositories (e.g., NAIC, IRDAI)  
📌 **Format:** CSV, JSON, Parquet  
📌 **Description:** Contains historical **insurance policyholder** details, **claims history**, and **fraudulent claims**.

📊 **Key Variables:**

* **Policy\_ID** (Unique identifier)
* **Customer\_Age** (Numeric)
* **Gender** (Categorical: Male/Female/Other)
* **Policy\_Type** (Categorical: Health, Auto, Life, Property)
* **Annual\_Income** ($$$)
* **Vehicle\_Age / Property\_Age** (Numeric)
* **Claim\_History** (Count of past claims)
* **Fraudulent\_Claim (Target)** (Binary: 1 = Fraud, 0 = Genuine)
* **Premium\_Amount** (Numeric)
* **Claim\_Amount** ($$$)
* **Risk\_Score** (Categorical: Low, Medium, High)

### **🔹 Customer Feedback & Sentiment Dataset**

📌 **Source:** Kaggle (Customer Review Datasets), Trustpilot, Insurance Company Data  
📌 **Format:** CSV, JSON  
📌 **Description:** Contains **customer feedback, complaints, and sentiments** extracted from reviews and social media.

📊 **Key Variables:**

* **Review\_ID** (Unique Identifier)
* **Customer\_ID**
* **Review\_Text** (Unstructured Text)
* **Sentiment\_Label** (Positive, Negative, Neutral)
* **Rating (1-5 stars)**
* **Service\_Type** (Claim, Policy Purchase, Customer Support)

### **🔹 Fraudulent Insurance Claims Dataset**

📌 **Source:** Kaggle, European Insurance Fraud Dataset, OpenML  
📌 **Format:** CSV, JSON  
📌 **Description:** Contains historical fraudulent claims with **anomaly detection labels**.

📊 **Key Variables:**

* **Claim\_ID**
* **Claim\_Date**
* **Policyholder\_ID**
* **Claim\_Amount**
* **Claim\_Type** (Medical, Vehicle, Home Damage)
* **Suspicious\_Flags** (Boolean)
* **Fraud\_Label** (1 = Fraud, 0 = Genuine)

### **🔹 Insurance Multilingual Policy Document Dataset**

📌 **Source:** Public Government Insurance Policies, Kaggle Text Datasets  
📌 **Format:** TXT, PDF, JSON  
📌 **Description:** Collection of **insurance policies in multiple languages** for NLP-based translation and summarization.

📊 **Key Variables:**

* **Policy\_ID**
* **Policy\_Text (English, Spanish, French, Hindi, etc.)**
* **Summarized\_Text** (Generated using NLP models)

### **🔹 Customer Segmentation Dataset (Unsupervised Learning - Clustering)**

📌 **Source:** Kaggle, Insurance Companies' CRM Data  
📌 **Format:** CSV, JSON  
📌 **Description:** Used for **clustering customers into risk profiles**.

📊 **Key Variables:**

* **Customer\_ID**
* **Demographic Data (Age, Income, Location, etc.)**
* **Number of Active Policies**
* **Total Premium Paid**
* **Claim Frequency**
* **Policy Upgrades**

**Data Set Explanation:**

## **1️⃣ Dataset 1: Insurance Risk & Claims Dataset**

📌 **Context:**This dataset contains **historical policyholder data, claim details, and fraud labels**. It is used to train models for **risk classification, claim prediction, and fraud detection**.

📌 **Content:**

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| **Policy\_ID** | String | Unique identifier for each policy |
| **Customer\_Age** | Integer | Age of policyholder |
| **Gender** | Categorical | Male, Female, Other |
| **Policy\_Type** | Categorical | Health, Auto, Life, Property |
| **Annual\_Income** | Float | Customer's yearly income |
| **Claim\_History** | Integer | Number of previous claims |
| **Fraudulent\_Claim** | Binary | 1 = Fraud, 0 = Genuine (Target Variable) |
| **Premium\_Amount** | Float | Amount paid for the policy |
| **Claim\_Amount** | Float | Amount claimed in a particular case |
| **Risk\_Score** | Categorical | Low, Medium, High (Derived feature) |

📌 **Preprocessing Steps:**✅ **Handling Missing Values**

* Fill missing numeric values using **median imputation** (for Income, Age).
* Fill missing categorical values using **mode imputation** (for Policy Type, Gender).

✅ **Encoding Categorical Data**

* **One-Hot Encoding** for Policy\_Type, Gender.
* **Label Encoding** for Risk\_Score (Low → 0, Medium → 1, High → 2).

✅ **Feature Scaling**

* Normalize numerical features like **Annual\_Income, Claim\_Amount, Premium\_Amount** using **MinMaxScaler**.

✅ **Outlier Detection**

* Use **IQR (Interquartile Range) and Z-score** to remove claim amounts that are suspiciously high.

✅ **Fraud Detection Anomaly Tagging**

* Apply **Isolation Forest & Autoencoders** for additional anomaly detection.

## **2️⃣ Dataset 2: Customer Feedback & Sentiment Dataset**

📌 **Context:**Used for **sentiment analysis of customer complaints and feedback** to improve service quality.

📌 **Content:**

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| **Review\_ID** | String | Unique identifier for each review |
| **Customer\_ID** | String | Unique customer ID |
| **Review\_Text** | String | Actual customer feedback |
| **Sentiment\_Label** | Categorical | Positive, Negative, Neutral |
| **Rating** | Integer | Customer rating (1-5) |
| **Service\_Type** | Categorical | Claim, Policy Purchase, Support |

📌 **Preprocessing Steps:**✅ **Text Cleaning & Tokenization**

* Remove **stopwords, punctuation, and special characters**.
* Tokenize and **lemmatize text** for NLP processing.

✅ **Sentiment Labeling**

* Convert **rating scores** to sentiment labels:
  + **1-2 stars** → Negative
  + **3 stars** → Neutral
  + **4-5 stars** → Positive

✅ **Feature Extraction**

* Use **TF-IDF, Word2Vec, BERT embeddings** to convert text into numerical vectors.

✅ **Class Imbalance Handling**

* If negative feedback is underrepresented, use **SMOTE (Synthetic Minority Over-sampling Technique)**.

## **3️⃣ Dataset 3: Fraudulent Claims Dataset**

📌 **Context:**This dataset is specifically used to **detect fraudulent claims using anomaly detection and classification models**.

📌 **Content:**

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| **Claim\_ID** | String | Unique identifier for each claim |
| **Claim\_Date** | Date | Date of claim submission |
| **Customer\_ID** | String | Unique ID of the policyholder |
| **Claim\_Amount** | Float | Amount requested in the claim |
| **Claim\_Type** | Categorical | Medical, Auto, Home, Life |
| **Suspicious\_Flags** | Binary | 1 = Suspicious, 0 = Normal |
| **Fraud\_Label** | Binary | 1 = Fraud, 0 = Genuine (Target Variable) |

📌 **Preprocessing Steps:**✅ **Feature Engineering**

* Generate **Claim-to-Income Ratio (Claim\_Amount / Annual\_Income)** as a new feature.
* Identify claims filed **within an unusually short period after policy issuance**.

✅ **Outlier Removal & Anomaly Detection**

* Use **Elliptic Envelope, Isolation Forest, and Local Outlier Factor (LOF)** to tag suspicious claims.

✅ **Fraud Score Calculation**

* Assign a **fraud probability score (0-1) using ensemble models (Random Forest + Neural Networks)**.

## **4️⃣ Dataset 4: Multilingual Insurance Policy Dataset**

📌 **Context:**Used to **train machine translation models** for multilingual document translation.

📌 **Content:**

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| **Policy\_ID** | String | Unique identifier for policy document |
| **Policy\_Text\_EN** | String | English version of the policy |
| **Policy\_Text\_FR** | String | French version |
| **Policy\_Text\_ES** | String | Spanish version |
| **Summarized\_Text** | String | AI-generated summary of the policy |

📌 **Preprocessing Steps:**✅ **Text Normalization**

* Convert **text to lowercase**, remove special characters, and expand contractions.

✅ **Tokenization & Embeddings**

* Apply **Byte Pair Encoding (BPE) or WordPiece Tokenization** for efficient text representation.

✅ **Parallel Corpus Alignment**

* Use **sentence alignment techniques** to ensure source and translated texts match correctly.

✅ **Model Fine-Tuning**

* Train **Hugging Face Transformer models (mBART, mT5)** using this dataset.

## **5️⃣ Dataset 5: Customer Segmentation Dataset (Unsupervised Learning)**

📌 **Context:**Used for **clustering customers into segments** based on purchasing behavior and policy preferences.

📌 **Content:**

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| **Customer\_ID** | String | Unique identifier for each customer |
| **Age** | Integer | Age of the customer |
| **Annual\_Income** | Float | Customer's yearly income |
| **Policy\_Count** | Integer | Number of active policies |
| **Total\_Premium\_Paid** | Float | Cumulative premium paid |
| **Claim\_Frequency** | Integer | Number of claims filed |
| **Policy\_Upgrades** | Integer | Number of policy changes |

📌 **Preprocessing Steps:**✅ **Feature Scaling**

* Apply **StandardScaler** for numerical variables.

✅ **Clustering Algorithm Selection**

* Use **K-Means, DBSCAN, or Hierarchical Clustering** for segmentation.

✅ **Feature Selection**

* Use **Principal Component Analysis (PCA)** to reduce dimensionality before clustering.

**Project Deliverables:**

## **1️⃣ Source Code & Implementation**

📌 **GitHub Repository / Zip File Submission**

* Full implementation of the project with **organized and well-structured code**.
* **Python scripts or Jupyter notebooks** used for different ML & NLP models.

📁 **Expected Folder Structure:**

📦 Insurance\_AI\_Project

┣ 📂 data # Raw & processed datasets

┣ 📂 notebooks # Jupyter notebooks for EDA, ML, NLP models

┣ 📂 models # Trained model files (Pickle, ONNX, TensorFlow, PyTorch)

┣ 📂 scripts # Python scripts for data processing & model training

┣ 📂 deployment # API files (Flask/FastAPI), Docker, Streamlit UI

┣ 📂 reports # Project documentation & reports

┣ 📜 README.md # Project overview & setup instructions

┣ 📜 requirements.txt # Dependencies & libraries used

┗ 📜 app.py # Main entry point for deployment

✅ **Source Code Should Include:**

* **Data Preprocessing Scripts** (Handling missing values, feature engineering, scaling).
* **ML Model Training Scripts** (Supervised, Unsupervised, NLP).
* **Evaluation Metrics Calculations** (Accuracy, AUC-ROC, F1-score, Silhouette Score, etc.).
* **API/Deployment Code** (Flask, FastAPI, Streamlit for interactive dashboard).
* **Automated Fraud Detection & Customer Segmentation Modules**.

## **2️⃣ Documentation & Reports**

📌 **Comprehensive Project Documentation** (📜 PDF, Markdown, or Notion)

✅ **README.md (GitHub Submission Required)**

* **Project Overview**
* **Dataset Used** (Source, preprocessing details)
* **Machine Learning Techniques Applied**
* **Deployment Instructions**

✅ **Technical Report (PDF or Notion Document)**

* **Executive Summary** – Overview of the problem and proposed AI solution.
* **Exploratory Data Analysis (EDA)** – Insights from visualizations & statistics.
* **Model Training & Evaluation** – Performance comparison of different models.
* **Challenges Faced & Improvements** – Any issues faced and how they were solved.
* **Future Enhancements** – Potential improvements or extensions.

✅ **Presentation Deck (PowerPoint or Google Slides)**

* **Problem Statement & Business Use Case**
* **Data Insights & Key Findings**
* **ML/NLP Model Approaches**
* **Real-World Applications & Deployment**

## **3️⃣ Model Artifacts & Evaluation Results**

📌 **Trained ML & NLP Models**

* Models saved in **Pickle (.pkl), ONNX, TensorFlow, PyTorch (.pt)** formats.
* Versioning handled using **MLflow or GitHub releases**.

📌 **Model Performance Metrics & Comparisons**

* Tables comparing different ML models on evaluation metrics.
* Charts (Confusion Matrix, AUC-ROC, Precision-Recall Curve).
* Sentiment analysis model accuracy using **BLEU, ROUGE, or F1-score**.

📌 **Feature Importance Analysis**

* Explainability techniques like **SHAP, LIME** for model interpretability.

## **4️⃣ Deployment Files (Optional But Recommended)**

📌 **API Development (Flask/FastAPI/Web App)**

* **Deployed API** to predict claim risk, detect fraud, or summarize policy documents.
* **Live Demo Link (if deployed on AWS, GCP, or Heroku)**.

📌 **Interactive Dashboard (Streamlit/Gradio/Flask UI)**

* Real-time visualization of **risk scores, fraud detection results, customer segmentation**.
* Multilingual chatbot integration for insurance query resolution.

## **5️⃣ Video Demo**

📌 **5-10 min project walkthrough video** explaining:

* **Problem Statement & Approach**
* **Model Training Process**
* **Results & Business Impact**
* **Live Demo of API or Dashboard**

### **📢 Final Checklist for Submission**

✅ **Source Code with Clear Documentation**✅ **Well-Written Technical Report (Markdown or PDF)**✅ **Model Artifacts & Evaluation Results**✅ **Presentation Deck**✅ **Deployment Files (if applicable)**✅ **Video Demo (Bonus for better grading/evaluation)**

**Project Guidelines:**

### **✅ General Coding Guidelines**

* Follow **PEP 8 (Python Enhancement Proposal 8)** for consistent formatting.
* Use **meaningful variable names** and avoid one-letter variables (e.g., customer\_age instead of ca).
* Write **modular code** with functions and classes instead of long scripts.
* **Avoid hardcoding values** – use **config files** (config.yaml, .env).
* Use **logging instead of print statements** (logging module in Python).

### **✅ Directory Structure & Organization**

📌 **Use a clear, modular structure** to separate different parts of the project.

**Example Structure:**

📦 Insurance\_AI\_Project

┣ 📂 data # Raw & processed datasets

┣ 📂 notebooks # Jupyter notebooks for EDA, ML, NLP models

┣ 📂 models # Trained model files (Pickle, ONNX, TensorFlow, PyTorch)

┣ 📂 scripts # Python scripts for data processing & model training

┣ 📂 deployment # API files (Flask/FastAPI), Docker, Streamlit UI

┣ 📂 reports # Project documentation & reports

┣ 📜 README.md # Project overview & setup instructions

┣ 📜 requirements.txt # Dependencies & libraries used

┗ 📜 app.py # Main entry point for deployment

✅ Keep **notebooks separate from scripts** (Jupyter Notebooks are for exploration, scripts are for actual processing).  
✅ Store **models separately** in models/ and use versioning.  
✅ Save **datasets in data/ and avoid committing large datasets to Git** (use .gitignore).

### **✅ Code Quality & Formatting Best Practices**

📌 **Maintain clean, structured code** using:

* **Black** for automatic formatting (black .)
* **Flake8** for linting and detecting issues (flake8 .)
* **Docstrings & Comments** for function descriptions

**Timeline:**

10 days

**PROJECT DOUBT CLARIFICATION SESSION ( PROJECT AND CLASS DOUBTS)**

**About Session:** The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.

**Note: Book the slot at least before 12:00 Pm on the same day**

**Timing: Monday-Saturday (4:00PM to 5:00PM)**

**Booking link :<https://forms.gle/XC553oSbMJ2Gcfug9>**

**For DE/BADM project/class topic doubt slot clarification session:**

**Booking link :** [**https://forms.gle/NtkQ4UV9cBV7Ac3C8**](https://forms.gle/NtkQ4UV9cBV7Ac3C8)

**Session timing:**

**For DE: 04:00 pm to 5:00 pm every saturday**

**For BADM 05:00 to 07:00 pm every saturday**

**LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)**

**About Session:** The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.

**Note: This form will Open only on Saturday (after 2 PM ) and Sunday on Every Week**

**Timing:**

**For BADM and DE**

**Monday-Saturday (11:30AM to 1:00PM)**

**For DS and AIML**

**Monday-Saturday (05:30PM to 07:00PM)**

**Booking link :** [**https://forms.gle/1m2Gsro41fLtZurRA**](https://forms.gle/1m2Gsro41fLtZurRA)