Insurance Ai Project

Insurance Risk Classification

---

1. Executive Summary

Machine learning models categorize insurance claims into High, Medium, and Low risk levels for fraud. This proactive approach analysing claim data to identify suspicious patterns, enabling targeted investigation of high-risk claims and efficient processing of low-risk ones. This strategy optimizes resource allocation and reduces financial losses from fraud.

---

2. Exploratory Data Analysis (EDA)

- **Data Overview:** 100,000 insurance claim records with 10 features.
- **Class Distribution:** fraudulent claims are normal imbalanced.
- **Key Patterns:**
    - if customer have the fraudulent claim likely to be involved in high Risk.
    - Health & auto have the high risk to make fraud.
    - High correlation between property age and claim amount.
- **Visuals Used:**
    - Hist plots for provide distribution of data.
    - Bar plots & Count plots give the detail of Risk class.
    - Heatmap for correlations between feature.

---

3. Model Training & Evaluation

- **Preprocessing Steps:**
    - One hot encoding for nominal categorical variables.
    - Label encoding of ordinal categorical variables.
    - Select the feature's & target metrics to train test split.
    - Feature Scaling.
- **Models Used:**
    - Random Forest
- **Performance Metrics:**

- Best model: **Random Forest**
- Accuracy: 100%
- Precision: 1.00
- Recall: 1.00
- F1-score: **1.00**

- **Cross validation** used to evaluate the model performance.

---

4. Challenges Faced & Improvements

- **Unavailable Relevant Features**: The dataset lacked some important real world features, so synthetic data was generated based on real time conditions to the dataset.
- **Handling Large Datasets:** Due to the large number of rows, so down sample the data to enable efficient plotting and visualization.
- **Data Noise:** Removed irrelevant features and handled inconsistent entries.

Insurance Claim Amount Prediction

---

1. Executive Summary

Machine learning models predict insurance claim amounts. This approach uses data analysis to identify patterns and build regression models, enabling estimation of claim costs. This helps in pricing policies, reserving funds, and managing financial risk.

---

2. Exploratory Data Analysis (EDA)

- **Data Overview**: 100,000 insurance records with 11 features.
- **Key Patterns:**
  - Property age increase claim amount also will increase.
  - If customer claim history is high the claim amounts will increase.
  - High risk customers have high claim amount.
- **Visuals Used:**
  - Box plots to show claim amount distribution across categorical features.
  - Cat plots & Count plots for categorical feature distributions.

- Line plot shows the trend, pattern over a continuous range over the time.
- Heatmap for feature correlations.

---

3. Model Training & Evaluation

- **Preprocessing Steps:**
  - One hot encoding for nominal categorical variables.
  - Label encoding of ordinal categorical variables.
  - Select the feature's & target metrics to train test split.
  - Feature Scaling.
- **Models Used:**
  - Linear Regression
  - Ridge Regression
  - Lasso Regression
  - ElasticNet Regression
- **Performance Metrics:**
  - Best model: **Linear Regression**
  - MAE: 1147
  - MSE: 2053469
  - RMSE: 1432
  - F1-score: **0.98**
- **Cross validation** used to evaluate the model performance.

---

5. Challenges Faced & Improvements

- **Unavailable Relevant Features**: The dataset lacked some important real world features, so synthetic data was generated based on real time conditions to the dataset.
- **Handling Large Datasets:** Due to the large number of rows, so down sample the data to enable efficient plotting and visualization.
- **Data Noise:** Removed irrelevant features and handled inconsistent entries.

Customer Segmentation

---

1. Executive Summary

Customer segmentation involves dividing customers into distinct groups based on shared characteristics and behaviour. This approach uses clustering algorithms and data analysis to identify patterns in customer behaviour and purchasing history. The customer segments enable targeted marketing, personalized services, and strategic decision making, ultimately improving customer satisfaction and business efficiency.

---

2. Exploratory Data Analysis (EDA)

- **Data Overview**: 53,502 insurance records with 12 features.
- **Key Patterns:**
  - Male & female both can involves all policy type.
  - Most people have two policy upgrades.
  - Business & Group policy types highly purchased.
- **Visuals Used:**
  - Hist plot will show the distribution.
  - Count plots for categorical feature distributions.
  - Heatmap for feature correlations.

---

3. Model Training & Evaluation

- **Preprocessing Steps:**
  - One hot encoding for nominal categorical variables.
  - Label encoding of ordinal categorical variables.
  - Select the features to train model.
  - Feature Scaling.
- **Models Used:**
  - KMeans
  - DBSCAN
- **Performance Metrics:**
  - Best model: **KMeans**
  - Silhouette score: 0.11

---

5. Challenges Faced & Improvements

- **Unavailable Relevant Features**: The dataset lacked some important real world features, so synthetic data was generated based on real time conditions to the dataset.
- **Handling Large Datasets:** Due to the large number of rows, so down sample the data to enable efficient plotting and visualization.
- **Data Noise:** Removed irrelevant features and handled inconsistent entries.

Insurance Fraud Detection

---

1. Executive Summary

Insurance fraud is a significant issue affecting profitability in the insurance sector. This project focuses on using machine learning models to identify and prevent fraudulent insurance claims. By analysing patterns in historical claim data, the solution enables proactive fraud detection, reducing financial losses.

---

2. Exploratory Data Analysis (EDA)

- **Data Overview:** 100,000 insurance claim records with 7 features.
- **Class Distribution:** fraudulent claims are highly imbalanced.
- **Key Patterns:**
    - if customer have the suspicious flag likely to be involved in fraud.
    - High correlation between suspicious flag and fraud likelihood.
- **Visuals Used:**
    - Bar plot & Cat plot for provide fraud frequency.
    - Count plot of claim amounts (fraud vs. non-fraud).
    - Heatmap for correlations between feature.

---

3. Model Training & Evaluation

- **Preprocessing Steps:**
    - Feature Engineering for policy types, loacations.
    - One-hot encoding for nominal categorical variables.

    - Select the feature's & target metrics to train test split.
    - SMOTE to balance the dataset.
    - Feature Scaling.
- **Models Used:**
    - Logistic Regression
    - Random Forest
- **Performance Metrics:**
    - Best model: **Random Forest**
    - Accuracy: 80%
    - Precision: 0.80
    - Recall: 0.80
    - F1-score: **0.80**
- **Cross validation** used to evaluate the model performance.

---

4. Challenges Faced & Improvements

- **Handling Large Datasets:** Due to the large number of rows, we down sample the data to enable efficient plotting and visualization.
- **Imbalanced Dataset:** Addressed using SMOTE and class weights.
- **Data Noise:** Removed irrelevant features and handled inconsistent entries.

Sentiment Analysis

---

1. Executive Summary

This model implements a sentiment analysis that classifies text into Positive, Negative, or Neutral categories. It utilizes the TextBlob library for polarity scoring, and applies text preprocessing techniques to clean and filter the input. This solution helps in understanding user sentiments from reviews & feedback.

---

2. Text Preprocessing

Steps Involved:

- Convert all text to lowercase for uniformity.

- Remove special characters using regex.
- Tokenize the text using.
- Remove stop words like the, is, and, etc., using a predefined stop word list.
- Joins text to normal sentence format.

---

3. Sentiment Classification Logic

- The cleaned text is passed to TextBlob, which returns a polarity score between -1 and 1.
- Classification is done as follows:
    - **Polarity > 0.1** - **Positive**
    - **Polarity < -0.1** - **Negative**
    - **Else** – **Neutral**
- The just call the function with text.

Text Extraction and Translation

---

1. Executive Summary

This model extracts text from various file formats like **.**txt, .docx, and .pdf to translates the extracted text from a source language to destination language using the googletrans library. It provides a flexible way to process and translate documents in a single pipeline.

---

2. Supported File Formats & Text Extraction Logic

File Extension Check:

- The file extension is extracted and checked using os.path.splitext().

Extraction Process by Format:

- **.txt** - Opens the file in UTF-8 mode and reads the text directly.
- **.docx** - Uses python docx to iterate over all paragraphs and extract the text.
- **.pdf** - Uses pdfplumber to loop through all pages and extract textual content.

Unsupported formats raise a ValueError.

---

3. Translation

- After text extraction, the content is split line by line using split("\n").
- Each line is passed to googletrans.Translator().translate() for translation.
- Translated lines are joined using "\n" and returned.

Text Summarization Using Transformers

---

1. Executive Summary

This model implements a text summarization pipeline using the t5-small model from Hugging Face's Transformers library. The goal is long textual inputs into concise summaries, enabling faster understanding and decision making in insurance policies.

---

2. Tools and Libraries Used

- transformers for using pre trained language models
- Pipeline to encapsulate the summarization step
- FunctionTransformer to wrap the summarizer function
- Joblib for saving and loading the pipeline
- Model used: t5-small lightweight and efficient.

Chatbot Model

---

1. Executive Summary

This model builds a foundational question answer retrieval system using sentence embeddings generated from a pre trained Transformer model. The solution encodes FAQ style questions into numerical vectors using SentenceTransformer, allowing for semantic similarity based chatbot responses.

---

2. Libraries and Tools Used

- Sentence transformers for semantic embedding of questions.
- Joblib for saving and loading the model components like question answer pairs, question embedding and model name.
- Model used all-MiniLM-L6-v2 optimized for speed & performance in semantic search tasks.