### **QUANTUM WOLF**

### **DATA INTELLIGENCE RESEARCH LAB**

Date:08-02-2025-12-02-2025

Name: ABDUL ZUHAIL M

**Automated Misinformation Detection Pipeline Using KNIME**

**1️.Problem Statement**

The rapid spread of fake news and misleading information online has become a significant challenge in the digital age. Misinformation can influence public opinion, create social unrest, and even impact political and economic stability. Traditional methods of fact-checking are time-consuming and cannot keep up with the speed at which misinformation spreads. There is a critical need for an automated, real-time solution to detect and classify misinformation effectively.

**2. Solution**

To address this problem, we propose an **Automated Misinformation Detection Pipeline** using **KNIME**, a powerful data analytics platform. This pipeline collects data from various sources (social media, news websites, and forums), preprocesses the data using Natural Language Processing (NLP), and classifies content as real or fake using AI/ML models like BERT and GPT-based architectures. The system also includes a visualization to monitor misinformation trends in real-time.

**3.Working**

**Step 1: Install and Set Up KNIME**

**1.1 Install Required Extensions**

To ensure KNIME has all the necessary functionalities, install the following extensions:

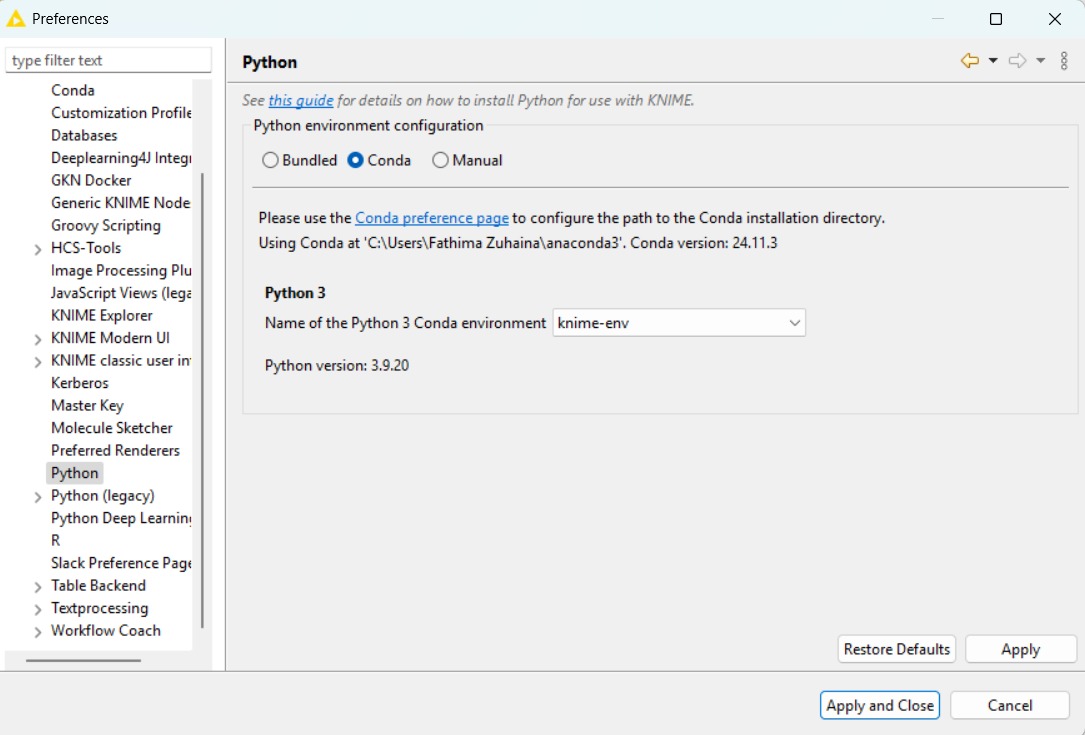
1. Open KNIME Analytics Platform.
2. Navigate to File > Install KNIME Extensions.
3. In the search bar, enter and install the following extensions:
   * KNIME Python Integration (KNIME Python Integration extension)
   * KNIME Text Processing (KNIME Textprocessing extension)
   * KNIME Deep Learning – TensorFlow (KNIME Deep Learning - TensorFlow 2 Integration)
4. Click Next > Accept the license agreement > Finish.
5. Restart KNIME to apply the changes.

**Step 2: Set Up Python in KNIME**

To use Python in KNIME:

1. Go to **KNIME → Preferences → KNIME → Python.**
2. Set up the Python 3 environment (recommended: Anaconda/Miniconda).
3. Created a Python environment (knime\_env) and installed required libraries (pandas, transformers, plotly, etc.).
4. Restart KNIME to apply changes.

**KNIME Python Configuration**



**Step 3: Data Collection**

**Overview**

News articles on misinformation are collected using NewsAPI. The extracted data is structured and stored for further analysis.

**Data Source: NewsAPI**

* API Endpoint:

**https://newsapi.org/v2/everything?q=misinformation&apiKey=bc7c3b237282442b92fd234148e413f2**

* Retrieves real-time articles mentioning "misinformation."

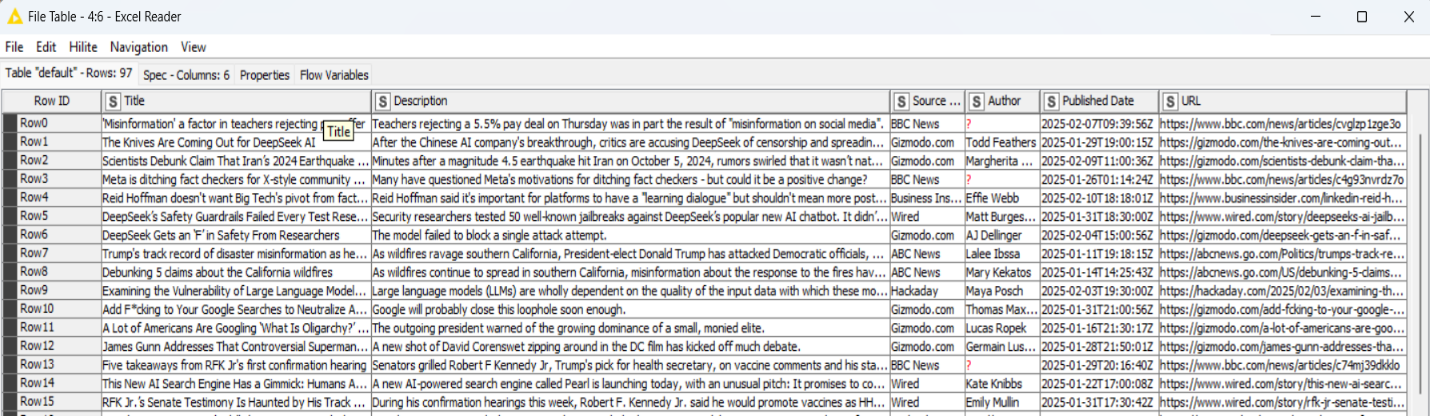
**KNIME Workflow & Nodes Used**

1. **GET Request Node** → Fetches news data from the API.
2. **JSON Path Node** → Extracts key fields (title, description, content, source.name, publishedAt, URL).
3. **Column Filter Node** → Keeps only relevant columns.
4. **Ungroup Node** → Flattens structured data.
5. **Excel Writer Node** → Saves the cleaned dataset.

**Final Output**

* Excel file containing structured news data.
* Columns: Title, Description, Content, Source, Published Date, URL.
* Next Step: Text preprocessing for misinformation classification.

**Expected Output**



**Step 4: Data Cleaning and Preprocessing**

**Overview**

This step cleans and preprocesses the collected news data to prepare it for analysis. It removes unwanted characters, stopwords, and inconsistencies to improve model accuracy.

**KNIME Workflow & Nodes Used**

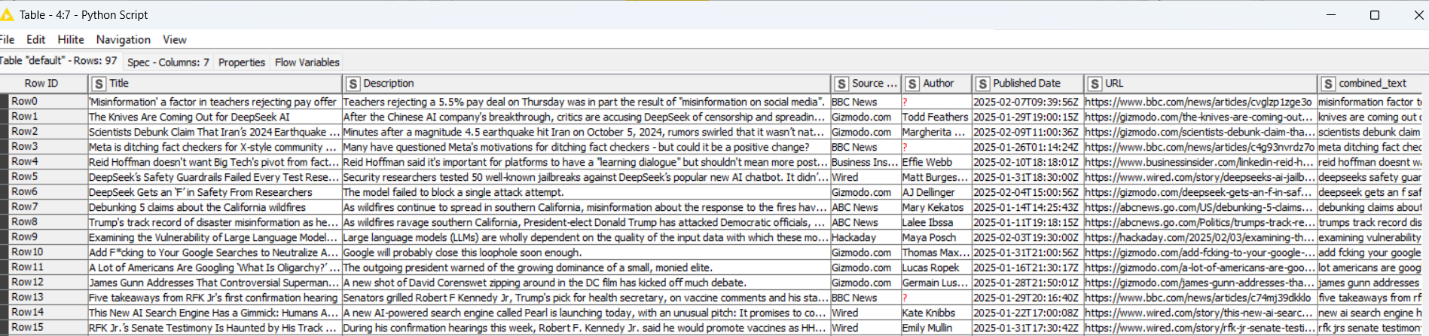
1. **Excel Reader Node** → Imports data from the previously saved Excel file.
2. **Python Script Node** → Performs text cleaning using Python (pandas, re).

**Key Cleaning Steps**

* **Remove Special Characters & Punctuation** → Cleans symbols like ! @ # $.
* **Remove Stopwords** → Eliminates common words like "the," "is," "and."
* **Remove Extra Spaces** → Trims unnecessary whitespaces.
* **Convert to Lowercase** → Ensures consistency (e.g., "Fake News" → "fake news").

**Final Output**

* Cleaned text stored in a new Excel file.
* Ready for further processing and classification**.**

**Expected Output**

**Step 5: Classification Using AI/ML Models**

**Overview**

This step classifies news articles as real or fake using a pre-trained RoBERTa model in KNIME. The model processes text and predicts whether an article is trustworthy or misleading.

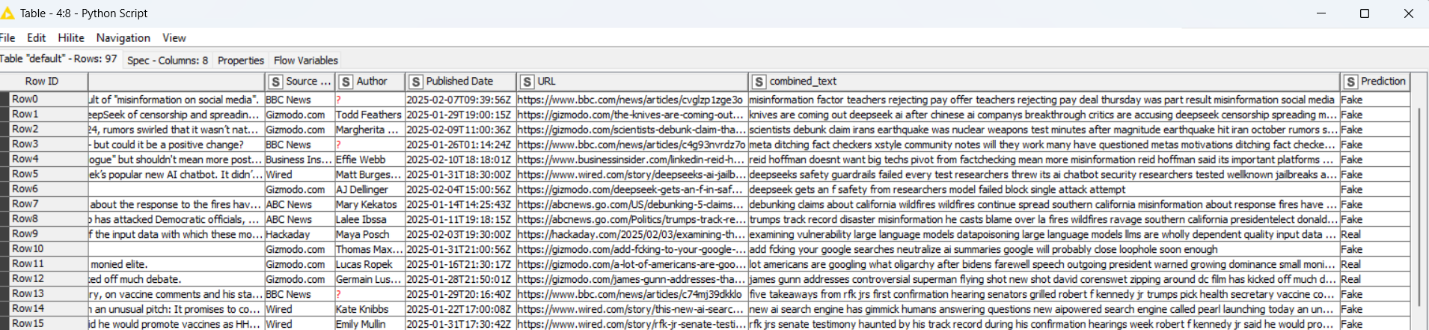
**KNIME Workflow & Nodes Used**

1. **Python Script Node** → Runs classification using Python (pandas, torch, transformers).
2. **Pre-trained RoBERTa Model** → A transformer-based model for text classification
3. **Text Tokenization**:
   1. The input text is tokenized using the RoBERTa tokenizer.
   2. The text is split into words/subwords, special tokens are added, and the text is padded/truncated to a fixed length.
4. **Model Inference**:
   1. The tokenized input is passed to the RoBERTa model.
   2. The model generates raw predictions (logits) for two classes:
      1. Class 0: Negative sentiment → **Fake**.
      2. Class 1: Positive sentiment → **Real**.
5. **Softmax Activation**:
   1. The logits are converted into probabilities using the softmax function.
   2. The probabilities for each class sum to 1.
6. **Class Prediction**:
   1. The class with the highest probability is selected as the final prediction.
   2. If Class 1 has a higher probability, the news is classified as **Real**.
   3. If Class 0 has a higher probability, the news is classified as **Fake**.

**Final Output**

* Each news article is labeled as Fake or Real based on model confidence.
* Results are saved in an Excel file for further analysis and trend monitoring.

**Expected Output**



**Step 6: Visualization of Misinformation Trends**

**Overview**

After classifying news as Real or Fake, data visualization helps analyze misinformation trends effectively.

**Key Insights**

* **Fake News Percentage** → Pie chart shows real vs. fake news distribution.
* **Top Sources** → Bar plot highlights major misinformation sources.
* **Geographical Trends** → Stacked area chart tracks fake news spread over time.

**KNIME Workflow & Nodes Used**

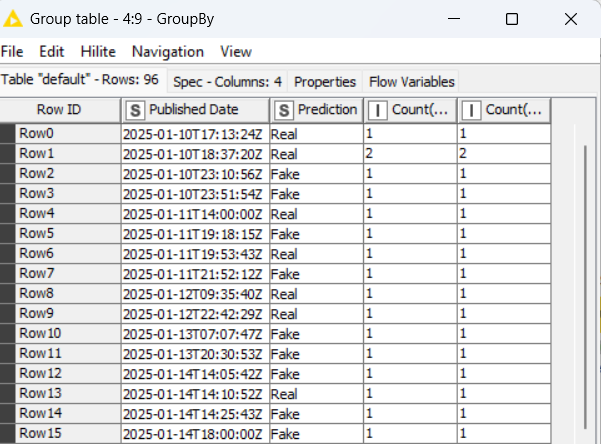
1. **Group By Node** → Aggregates data by source, prediction, or region.
2. **Bar Plot Node** → Displays fake vs. real news distribution.
3. **Stacked Area Chart Node** → Visualizes misinformation trends over time.
4. **Pie Chart Node** → Represents fake news percentage.
5. **Table View Node** → Shows structured data for deeper analysis.

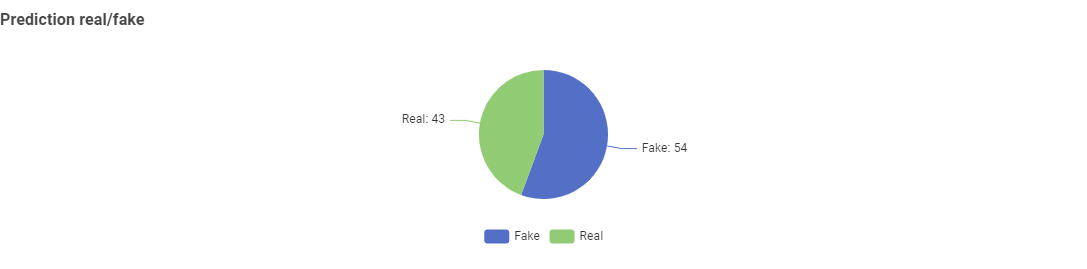
**Final Output**

A clear visual representation of misinformation trends, aiding in the identification of fake news sources and patterns.

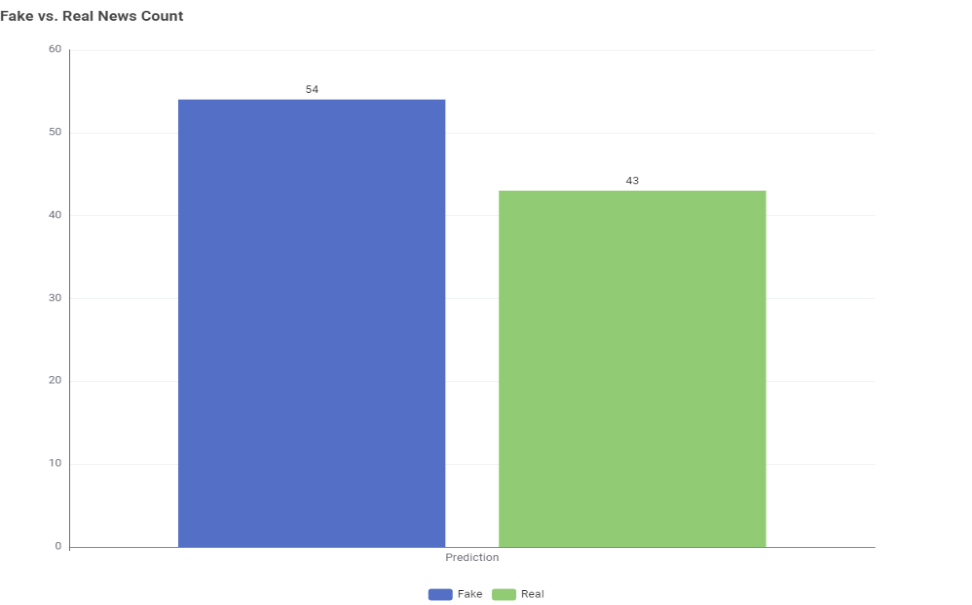
**Expected Output**

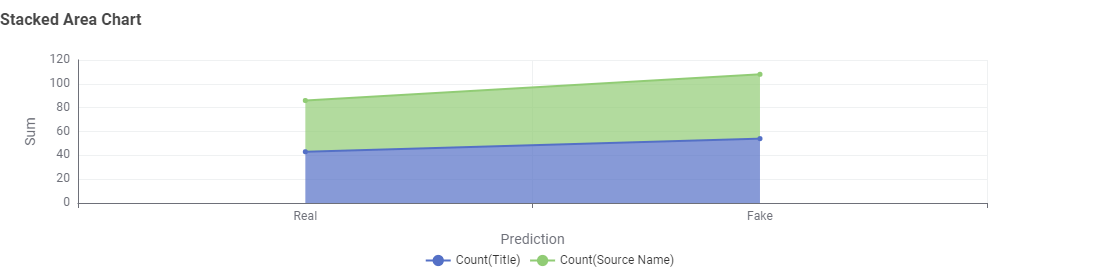
**GroupBy Node**



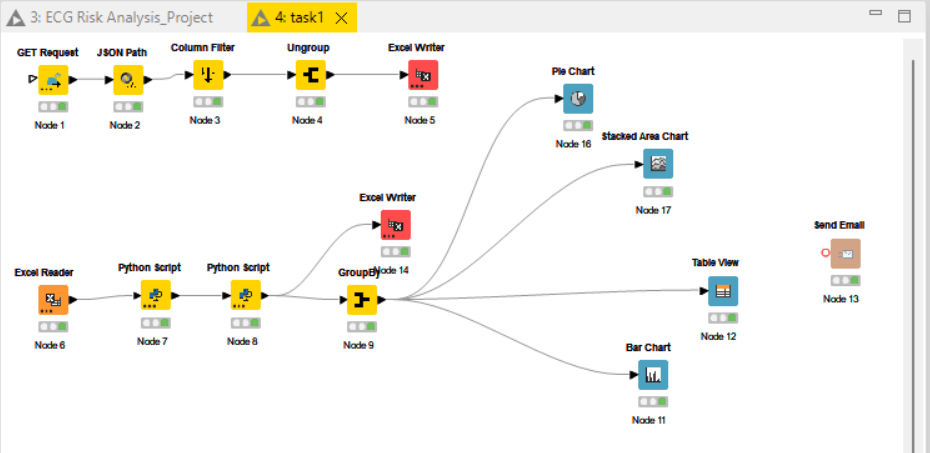
**Pie chart**

**Bar plot**

**Stacked Area Chart**



**Workflow Environment**



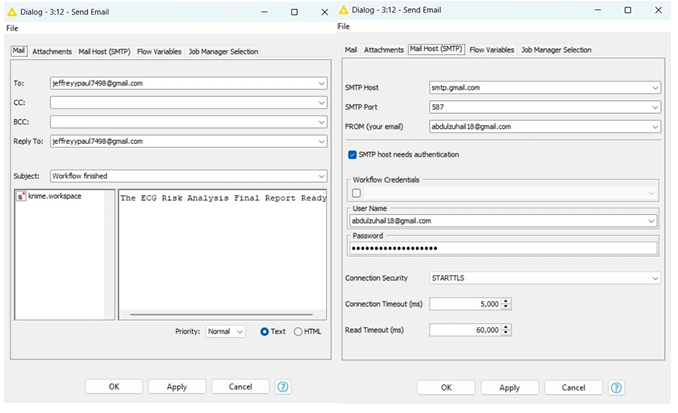
**Step 7: Automate the Data Pipeline in KNIME**

The “Send Email” node sends automated reports via email after processing is complete.

How to Set Up:

* Search for "Send Email" in the Node Repository.
* Drag & Drop "Send Email" into your workflow.
* Double-click the node and configure:
* SMTP Server: smtp.gmail.com (for Gmail) or your company's email server.
* Port: 587
* Enable Authentication: Enabled
* Username & Password: Enter your email credentials.
* Recipient Email: Add email addresses of stakeholders (e.g., client@company.com).
* Subject: "Daily ESG Risk Report"
* Attachment: Attach the processed report (e.g., esg\_risk\_analysis.csv).
* Connect the Email Node at the end of the pipeline.
* Now, KNIME will send automated reports after processing is completed.

**Expected Output**



**4. Problems We Faced During the Project**

During the implementation of our project, we encountered several challenges that required extensive troubleshooting and problem-solving. Below are the key issues we faced and how we resolved them:

**1. KNIME Extensions Installation Issues**

* **Problem:**
  + Errors occurred while installing required KNIME extensions, preventing successful installation.
* **Solution:**
  + Adjusted KNIME preferences, manually downloaded and installed extensions, and ensured all dependencies were correctly configured.

**2. KNIME Platform Configuration**

* **Problem:**
  + Some extensions failed to integrate properly after installation, causing delays.
* **Solution:**
  + Reviewed KNIME’s configuration settings, reinstalled necessary components, and ensured all dependencies were correctly linked.

**3. Python Code Execution Errors**

* **Problem:**
  + Frequent errors occurred while executing Python scripts within KNIME.
* **Solution:**
  + Created a dedicated Python environment with required libraries and debugged each error step by step.

**4. Connection Failures and Repeated Errors**

* **Problem:**
  + Persistent connection failures occurred during component integration.
* **Solution:**
  + Checked network configurations, adjusted firewall settings, and verified API access to stabilize the workflow.

**5. KNIME Workflow Saving Issue**

* **Problem:**
  + KNIME failed to save the project, resulting in the loss of three hours of work.
* **Solution:**
  + Investigated autosave settings and implemented a strict manual saving routine with frequent backups.

**6. Automatic Deletion of Extensions**

* **Problem:**
  + Installed extensions disappeared automatically after a short period.
* **Solution:**
  + Reinstalled extensions, ensured proper storage in the KNIME directory, and modified settings to prevent automatic deletion.

**5.Conclusion**

The Automated Misinformation Detection Pipeline provides an efficient, AI-driven approach to detecting and monitoring misinformation. Using KNIME, we successfully built a system that classifies news, analyzes trends, and generates reports. Future improvements may include integrating Apache Airflow for full automation and expanding the dataset sources for better accuracy.