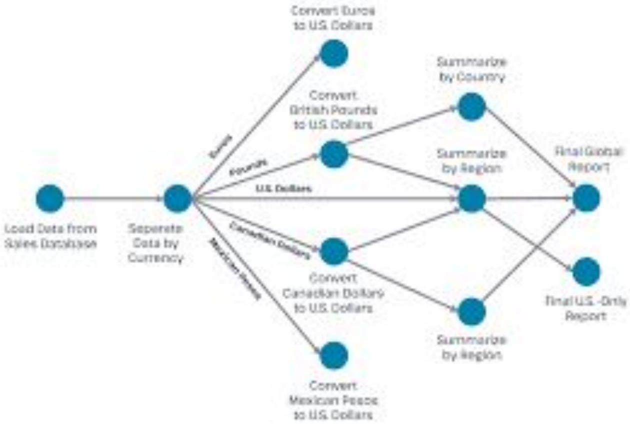
# Extract Data Analytics Pipeline from Research Papers Using LLM and RAG

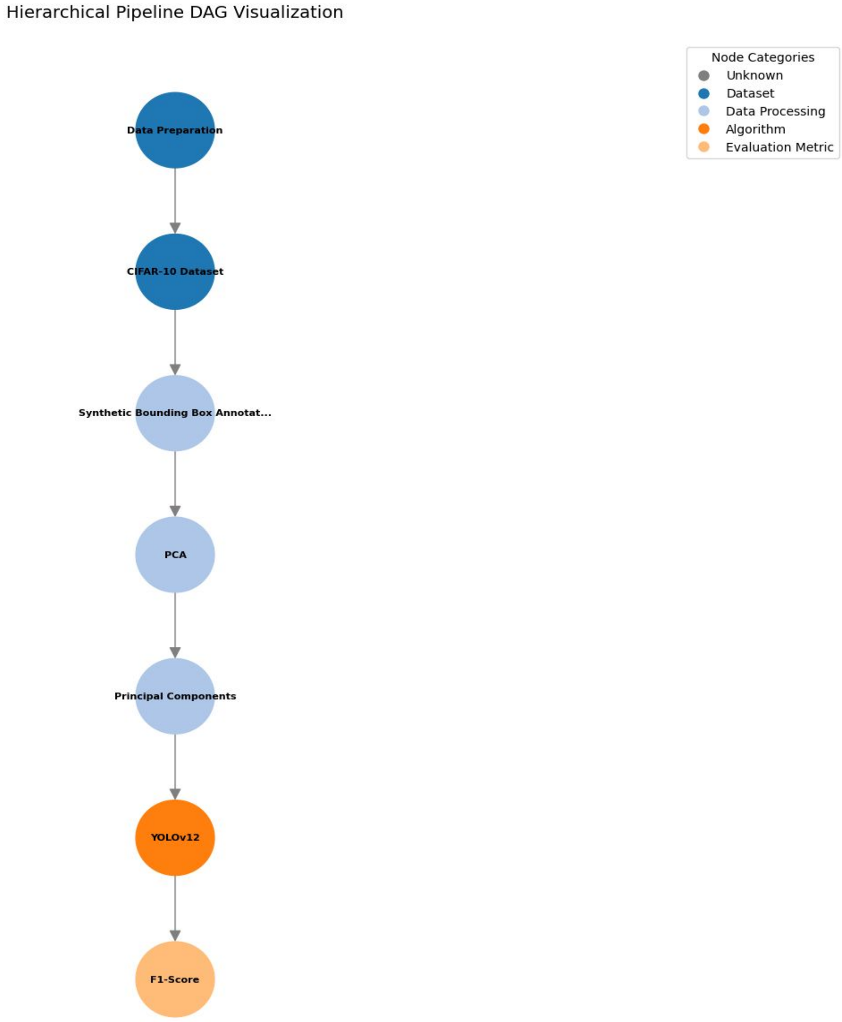
Name - Varun Karwa
Matrkr Nr - 428443
Supervisor -  Dr David A Selby

# Motivation




An automated system capable of extracting structured data analytics pipeline from research papers.

# What is Data Analytics Pipeline?



# Problem Statement

Can automated system comprised of LLMs and RAG, extract and generate data analytics pipeline from research papers containing accurate information for replication?

# Research Goals

- Accuracy of automated pipeline extraction
- Efficiency of RAG-based retrieval
- Superiority over existing solutions

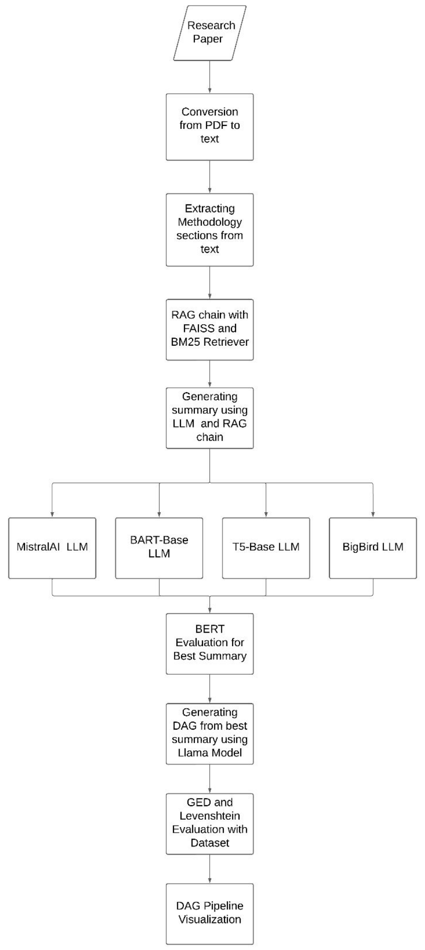
# Dataset

- No dataset available with summaries and structured pipelines in DAG format.
- Solution?

# Synthetic Benchmark Dataset

- Contains 15 synthetic research paper.
- Dataset includes
  - Title of the paper
  - Content of the paper
  - Manually generated DAG pipeline
- Used for evaluation purpose.

# Experimental Setup



# Experimental Setup: Benchmark Dataset

- Created Dataset of 15 synthetic research papers using LLM
- Pipeline in research paper were made up of predefined set of datasets, algorithms, data pre-processing methods, evaluation metrics.
- Directed Acyclic Graphs were curated manually for each paper.

# Experimental Setup: Methodology Section Extraction

- Extracted Methodology section due to LLM token input size limitations.
- Resulting in incomplete or no response at all.
- Tried using NLP models for identifying algorithms, etc.
- Extracted sections using Regex.

# Experimental Setup: LLM Setup

- MistralAI, T5-Base, Bart-Base, Bigbird-pegasus LLM were initialized with HuggingfacePipeline
- MistralAI and Llama models were loaded in 4 bit quantization configuration.
- All queries were run on Kaggle GPU due to Huggingface API rate limits.

# Experimental Setup: RAG Chain

- RAG chain was integrated to retrieve relevant content from research papers.
- Ensemble retriever of FAISS and BM25 index were implemented
- FAISS indexing for semantic similarity search
- BM25 indexing for keyword-based retrieval
- Helps reducing LLM hallucination

# Experimental Setup: Querying LLMs

- Queried 4 LLMs sequentially (MistralAI, T5-Base, Bart-Base, Bigbird-pegasus)
- Crafted prompt with retrieved context from RAG and asked to generate summary.
- All summaries were evaluated against extracted methodology section with BERTScore to choose best matching one.

# Experimental Setup: Generation of DAG

- Summary generated were feeded to Llama to generate DAG pipeline for final result.
- Prompt explicitly mentioned to created DAG pipelines containing dataset, algorithms, data processing methods, and evaluation metrics in JSON format.
- LLM returned the result in format of nodes and edges with garbage content
- Used Llama instead of MistralAI.

# Experimental Setup: DAG Structure


```
{
  "nodes": [
    {
      "name": "ImageNet",
      "input": [],
      "category": "Dataset"
    },
    {
      "name": "PCA",
      "input": ["ImageNet"],
      "category": "Data Pre-Processing Method"
    },
    ...
  ],
  "edges": [
    {
      "source": "ImageNet",
      "target": "PCA"
    },
    ....
  ]
}
```

# Experimental Setup: Post-processing of JSON

- Post-processing required due to overcome the issue of incomplete JSON result.
- Node and category names were standardized due to LLM inconsistencies in naming entities.
- Adding missing edges in the result.
- Removing non machine-learning nodes.

# Experimental Setup: Pipeline Evaluation

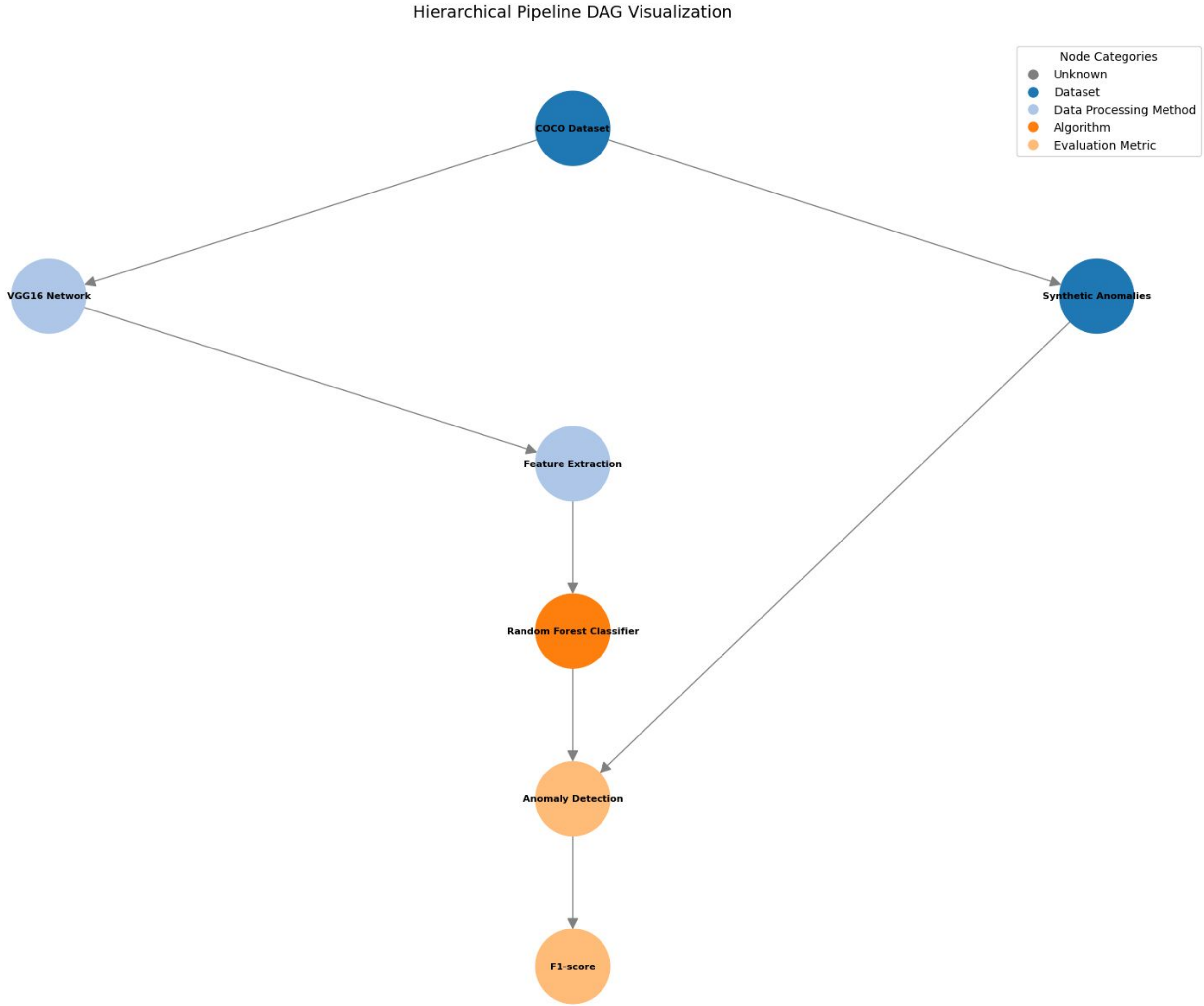
- Generated pipelines were assessed by Graph edit distance and Levenshtein Similarity evaluation metrics.
- Exact GED measure was calculated for nodes less than 15
- Approximate GED measure with greedy matching was calculated for more than 15 nodes.
- Levenshtein distance measured for nodes with same category and normalized and converted in similarity.

# Experimental Setup: Pipeline Visualization

- JSON converted into graphs using NetworkX library.
- Graphs were visualized with matplotlib to show the sequential steps of execution.

# Experimental Setup: ArXiv Papers Pipeline Generation

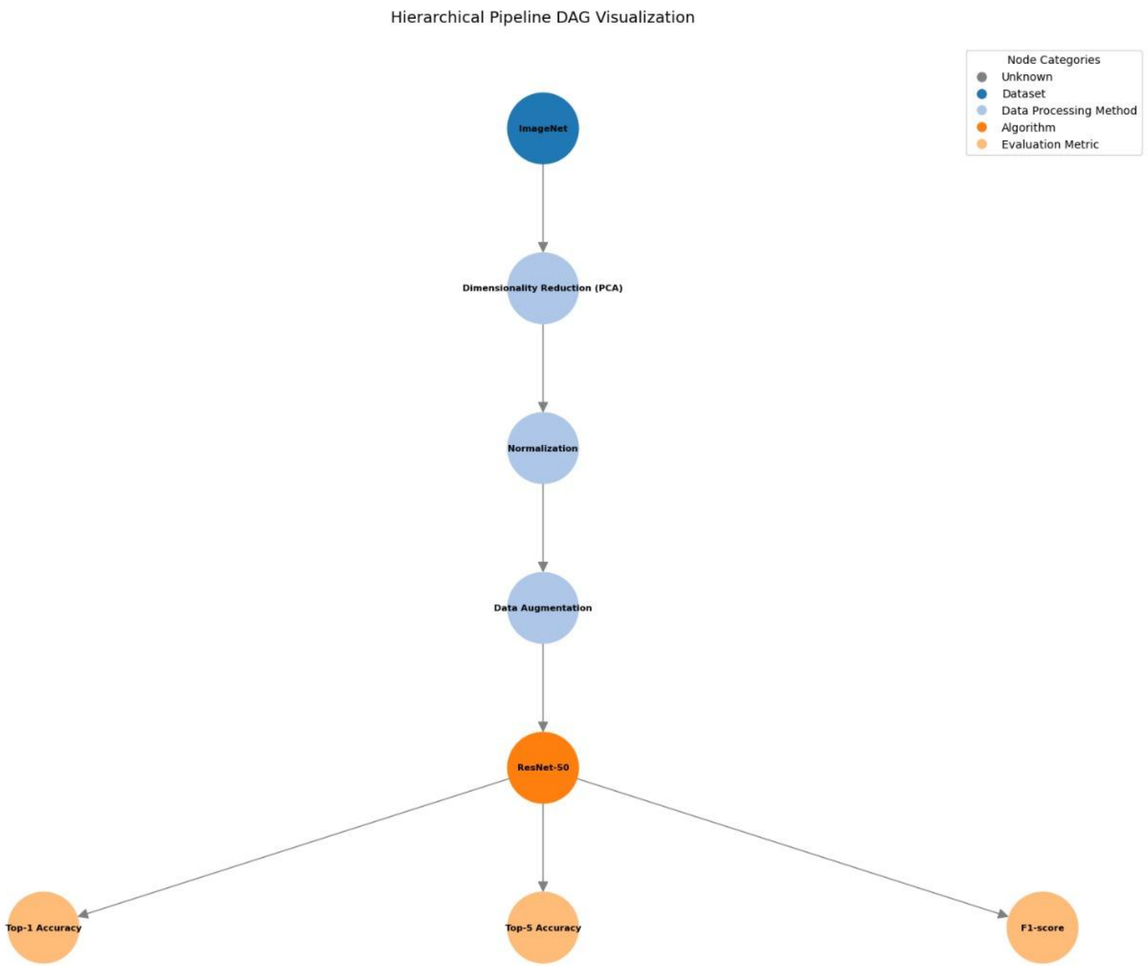
- Links of Research Papers were provided to system to generate pipeline.
- Done to test if the system can produce pipeline for actual research papers.
- Manually evaluated for some papers, accurate upto good extent.

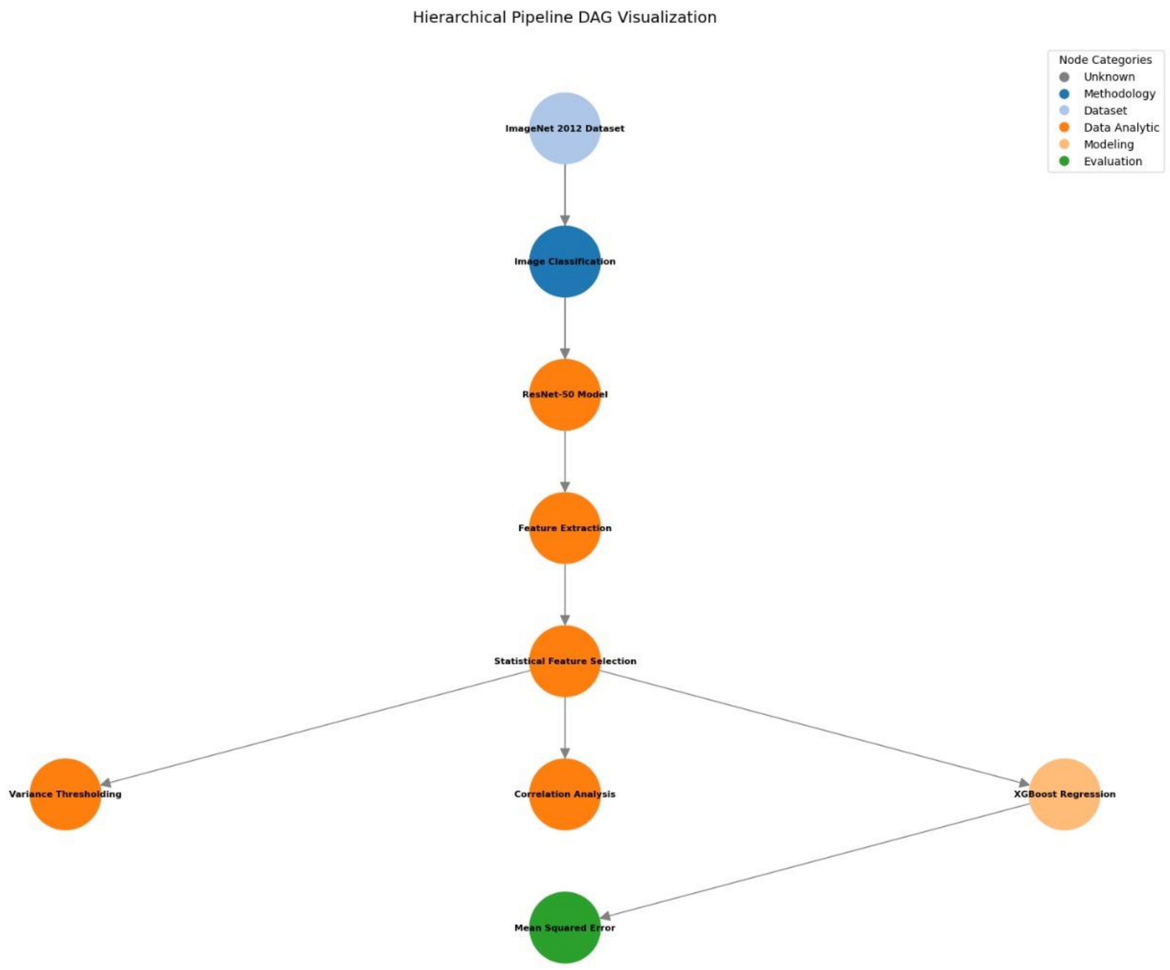


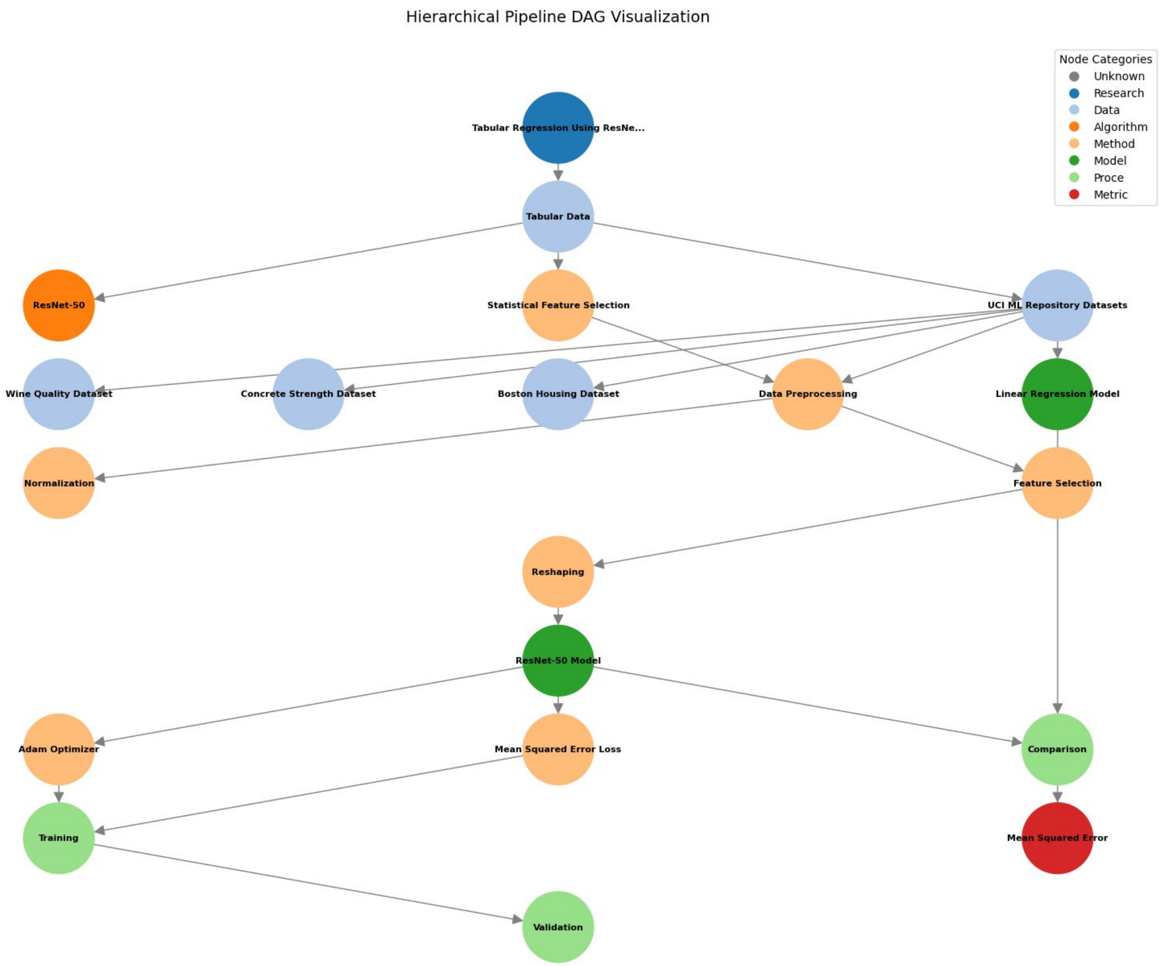
Hierarchical Pipeline DAG Visualization
Node Categories
Unknown
Dataset
Data Processing Method
Algorithm
Evaluation Metric
COCO Dataset
VGG16 Network
Synthetic Anomalies
Feature Extraction
Random Forest Classifier
Anomaly Detection
F1-score

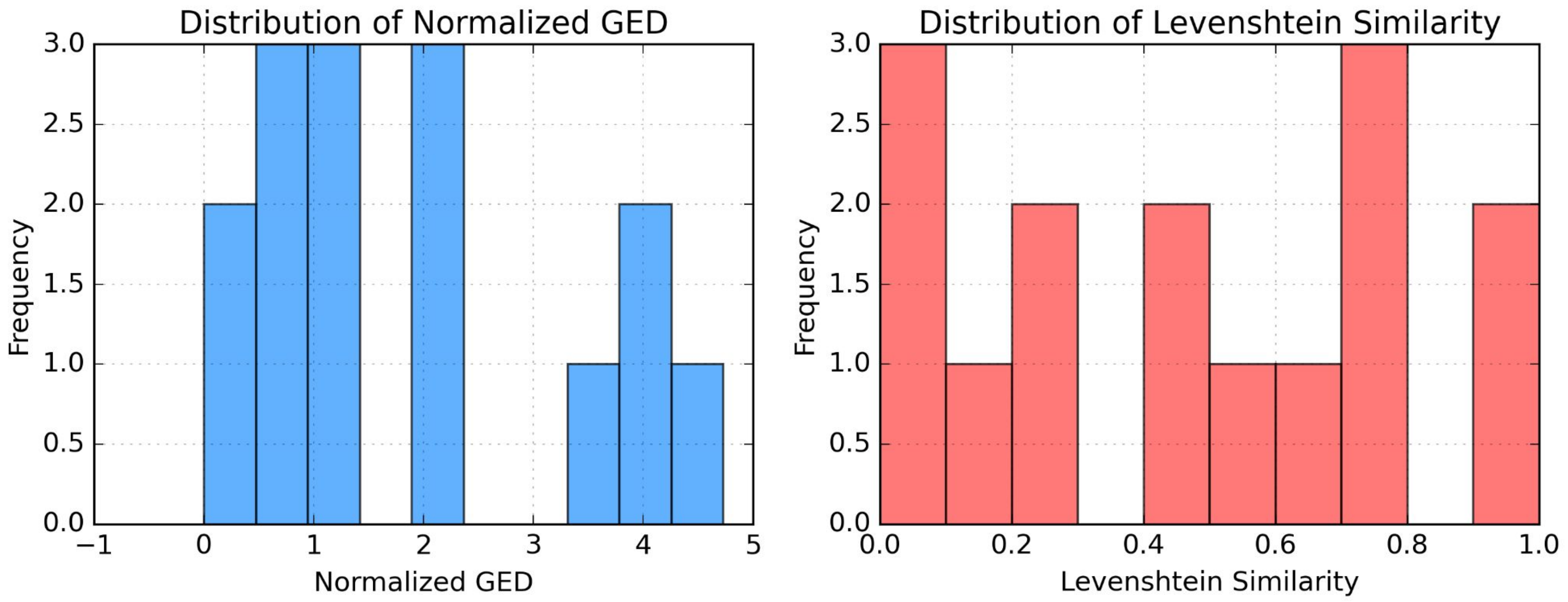
# Results

- According to BERTScore all 4 LLMs provided accurate summaries for different papers.
- RAG chain demonstrated efficient retrieval of relevant context validated by high BERTScores
- Two papers exhibited near-perfect similarity, while two papers revealed significant divergence.
- For some papers, GED and Levenshtein both were higher.
- Overall, there was general pattern of complementing values of both metrics.

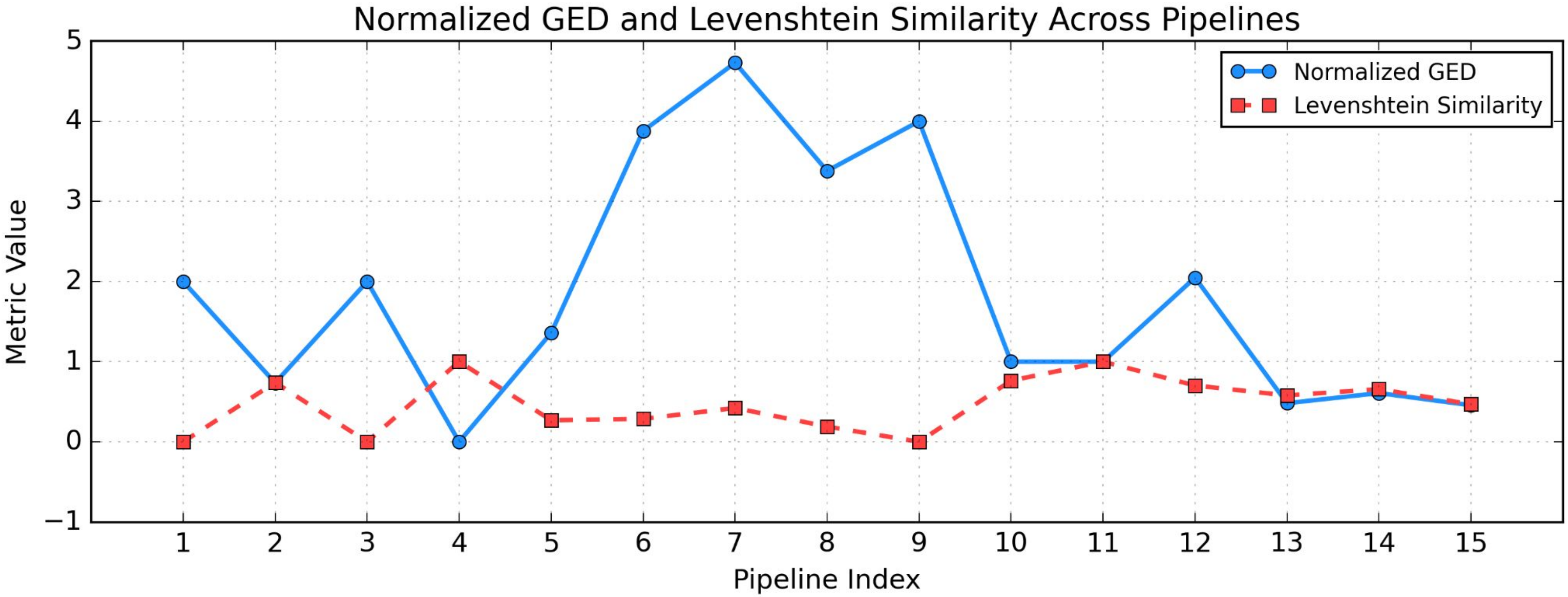

Hierarchical Pipeline DAG Visualization
Node Categories
Unknown
Dataset
Data Processing Method
Algorithm
Evaluation Metric
ImageNet
Dimensionality Reduction (PCA)
Normalization
Data Augmentation
ResNet-50
Top-1 Accuracy
Top-5 Accuracy
F1-score


Hierarchical Pipeline DAG Visualization
Node Categories
Unknown
Methodology
Dataset
Data Analytic
Modeling
Evaluation
ImageNet 2012 Dataset
Image Classification
ResNet-50 Model
Feature Extraction
Statistical Feature Selection
Variance Thresholding
Correlation Analysis
XGBoost Regression
Mean Squared Error


Hierarchical Pipeline DAG Visualization
Node Categories
Unknown
Research
Data
Algorithm
Method
Model
Proce
Metric
Tabular Regression Using ResNe...
Tabular Data
ResNet-50
Statistical Feature Selection
UCI ML Repository Datasets
Wine Quality Dataset
Concrete Strength Dataset
Boston Housing Dataset
Data Preprocessing
Linear Regression Model
Normalization
Feature Selection
Reshaping
ResNet-50 Model
Adam Optimizer
Mean Squared Error Loss
Comparison
Training
Mean Squared Error
Validation



Distribution of Normalized GED
Frequency
Normalized GED
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−1
0
1
2
3
4
5
Distribution of Levenshtein Similarity
Frequency
Levenshtein Similarity
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0



Normalized GED and Levenshtein Similarity Across Pipelines
Normalized GED
Levenshtein Similarity
Metric Value
Pipeline Index
5
4
3
2
1
0
−1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

# Open Problems

- Dataset Limitations
- LLM Inconsistencies
- Evaluation Logic
- Hardware Constraints

# Future Scope

- Benchmark Dataset with more than thousands of papers.
- LLM Fine-Tuning for better results
- Advance NLP Models for handling variable entity names.
- Increasing LLM token input size
- Parallel Executions of LLM query processing
- Exploring other evaluation logic to make conclusion more
- Storing more information such as description for each node in DAG pipeline.

# Conclusion

- LLMs generated summary with high accuracy.
- Accuracy improved when context retrieved using RAG chain.
- Evaluation metric suggested
  - System generated extracted accurate pipeline upto an extent.
  - Lower GED and higher levenshtein value relation was maintained for most papers.
  - Although, higher values of both for some papers showed scope of improvement.
- System tested on original ArXiv papers with manual evaluation of pipelines.

# THANK YOU!