GUJARAT TECHNOLOGICAL UNIVERSITY
SCHOOL OF ENGINEERING AND TECHNOLOGY

Institute Code: 137

Internal Review 1 A

Semester : III

Extracting Insights From University Data Using AI Powered Visualization

Prepared by:
Khushi Bhatt

Enrolment Number:
241370795002

Internal Guide:
Prof Mahesh Panchal

Designation:
Assistant Professor

Presentation Outline

- Introduction of the Research Domain
- Literature Review
- Comparative Analysis
- Identified Research Gap
- Problem Statement
- Proposed Methodology
- Objectives of the work
- Timeline Chart
- References

Introduction of the Research Domain

Universities has a huge amount of data—about admissions, academics, teachers, and student activities. But most of this data is not fully used because it is large, difficult to handle, and the tools to study it are limited.

AI-Powered Visualization can:

- Generate interactive visuals and statistical reports
- Extract insights from messy, large datasets
- Present information in clear and dynamic ways
- Support faster, data-driven decisions

Why University Data?

- Covers admissions, academics, faculty, and student engagement records.
- Provides insights into institutional performance and trends.
- Supports better decision-making and strategic planning.
- Automation reduces manual reporting effort and saves time.

Introduction of the Research Domain

Why AI-Powered Visualization?

- Traditional tools struggle with scale and diversity of modern data
- Deep learning detects hidden patterns and relationships
- Transforms raw data into interactive visuals (charts) and summaries
- Makes insights accessible for both technical and non-technical users

Need for This Research:

- Current analytics rely on manual processing (slow, error-prone)
- Static reports lack real-time insights
- Institutions struggle to extract timely, reliable insights
- Difficulty handling messy, large-scale datasets
- Lack of dynamic, interactive dashboards

Literature Review

Sr. No.	Paper Title	Publication	Year
1.	Data2Vis – Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks	IEEE Computer Graphics and Application (IEEE)	2019

- **Objective:**
  To automatically generate valid and meaningful data visualizations directly from tabular datasets.

- **Methodology:**
  - Represented dataset schemas (field names + types) as token sequences.
  - Used a Seq2Seq RNN (LSTM with attention) to translate dataset tokens into Vega-Lite specification tokens.
  - Trained the model on large dataset–visualization pairs to learn common mapping patterns.
  - Compiled the generated Vega-Lite specs into Vega and rendered final charts with the Vega runtime.

- **Result:**
  Data2Vis produced syntactically valid Vega-Lite specifications in ~80–90% of cases. The generated charts matched common visualization practices, proving the feasibility of an end-to-end neural approach for visualization recommendation.

Literature Review

Sr. No.	Paper Title	Publication	Year
2.	ADVISor – Automatic Visualization Answer for Natural-Language Question on Tabular Data	IEEE Pacific Visualization Symposium(IEEE)	2021

- **Objective:**
  To automatically generate relevant data visualizations in response to natural language questions over tabular data, without requiring technical knowledge from users.
- **Methodology:**
  - Employed pre-trained language models(BERT) to encode natural language queries and table schemas.
  - Designed a multi-task neural network to identify relevant attributes, aggregation functions, and visualization types.
  - Used rule-based templates to map predictions into visualization structures.
  - Developed an annotation generator to highlight key data insights in visual answers.
- **Result:**
  ADVISor effectively converted natural language questions into accurate visualizations, outperforming baseline methods. Produced clear, user-friendly charts that helped users quickly understand insights.

Literature Review

Sr. No.	Paper Title	Publication	Year
3.	MultiVision – Designing Analytical Dashboards with Deep Learning-Based Recommendation	IEEE Transactions on Visualization and Computer Graphics (IEEE)	2022

- **Objective:**
  To recommend multi-view dashboards by combining deep learning models for single-chart quality and multi-chart complementarity, enabling balanced and diverse analytics dashboards.
- **Methodology:**
  - Trained a neural network to score single-chart quality based on relevance and readability.
  - Designed a second model using provenance data to evaluate dashboard-level quality (diversity and complementarity).
  - Implemented a mixed-initiative system allowing users to refine and adapt system recommendations interactively.
- **Result:**
  MultiVision successfully generated dashboards with high-quality, non-redundant, and complementary charts. User studies confirmed its effectiveness in improving both dashboard relevance and diversity compared to baseline systems.

Literature Review

Sr. No.	Paper Title	Publication	Year
4.	DashBot – Insight-Driven Dashboard Generation Based on Deep Reinforcement Learning	IEEE Transactions on Visualization and Computer Graphics(IEEE)	2023

- **Objective:**
  To automate dashboard creation by using deep reinforcement learning to select, arrange, and optimize charts, ensuring diversity and insightfulness.
- **Methodology:**
  - Modeled dashboard design as a Markov Decision Process (MDP) with states (current dashboard), actions (chart selection), and rewards (dashboard quality).
  - Implemented a deep reinforcement learning agent to sequentially choose charts and layouts.
  - Encoded visualization design rules into the reward function to ensure effectiveness, expressiveness, and diversity.
  - Used probabilistic ranking to refine and order final dashboard recommendations.
- **Result:**
  DashBot produced dashboards that were more insightful, diverse, and well-structured compared to rule-based methods.

Literature Review

Sr. No.	Paper Title	Publication	Year
5.	VIS+AI – Integrating Visualization with Artificial Intelligence for Efficient Data Analysis	Visual Computing for Industry, Biomedicine, and Art (Springer)	2024

- **Objective:**
  To integrate visualization and artificial intelligence into a unified framework, enabling efficient data analysis through AI-assisted visualization and visualization-enhanced AI.
- **Methodology:**
  - Defined three integration modes: AI for Visualization (AI4VIS), Visualization for AI (VIS4AI), and VIS+AI (mutual integration).
  - Surveyed 113 research papers to map technical approaches, application domains, and system designs.
  - Analyzed methods across machine learning, deep learning, NLP, and interactive visualization.
  - Proposed a conceptual framework for seamless AI–visualization integration in real-world domains.
- **Result:**
  VIS+AI provided a taxonomy and framework for future systems that combine visualization and AI. It highlighted challenges and opportunities, offering guidance for building explainable, interactive, and efficient analytics solutions.

Literature Review

Sr. No.	Paper Title	Publication	Year
6.	AUTOVI – Empowering Effective Tracing and Visualizations with AI	Internatinal Conference Information Visualisation(IEEE)	2024

- **Objective:**
  To develop an AI-powered system that automates tracing, preprocessing, and visualization of raw datasets, enabling effective and scalable visual analytics.
- **Methodology:**
  - Designed a preprocessing pipeline to clean, normalize, and standardize tabular data.
  - Applied a multi-layer perceptron (MLP) classifier to predict suitable visualization types and required transformations.
  - Integrated transformation tracing to automatically recommend data preparation steps.
  - Incorporated a feedback loop where user corrections refine future predictions.
- **Result:**
  AUTOVI achieved accurate and efficient visualization recommendations while reducing manual preprocessing effort. The feedback mechanism enhanced adaptability, making the system more effective over time.

Literature Review

Sr. No.	Paper Title	Publication	Year
7.	AdaVis – Adaptive and Explainable Visualization Recommendation for Tabular Data	IEEE Transactions on Visualization and Computer Graphics(IEEE)	2024

Objective:

To provide adaptive and explainable visualization recommendations for tabular data.

Methodology:

- Constructed a knowledge graph to represent relations between datasets, attributes, and visualization choices.
- Used box embeddings to capture one-to-many mappings, where one dataset feature may correspond to multiple visualization options.
- Applied a GNN with attention to learn contextual embeddings and highlight influential features.
- Implemented an inference framework to generate adaptive recommendations directly from learned representations.

Result:

AdaVis improved recommendation accuracy and explainability over prior systems. The attention mechanism enabled fine-grained interpretability at dataset and feature levels, showing users why a visualization was recommended.

Literature Review

Sr. No.	Paper Title	Publication	Year
8.	Waltzboard – Multi-Criteria Automated Dashboard Design for Exploratory Analysis	IEEE Pacific Visualization Symposium(IEEE)	2025

- **Objective:**
  To generate dashboards automatically by optimizing multiple criteria such as specificity, diversity, coverage, and parsimony, supporting exploratory analysis.
- **Methodology:**
  - Proposed five evaluation measures: Relevance, novelty, variety completeness, simplicity
  - Developed a three-phase search algorithm to explore dashboard candidates, Filtering / Pruning and Evaluation & Ranking.
  - Built an interactive interface allowing users to adjust weights of evaluation criteria and regenerate dashboards in real time.
- **Result:**
  Waltzboard produced dashboards that balanced multiple design criteria better than single-metric approaches. User studies validated that the system enabled more effective exploratory analysis with diverse and useful dashboards.

Comparative Analysis

Sr No.	Paper Title	Model And Datasets	Finding/Findings	Statistical Parameter
1.	AdaVis – Adaptive and Explainable Visualization Recommendation for Tabular Data (IEEE 2024)	**Datasets:** VizML corpus (30k dataset-visualization pairs,) Two column data set **Model:** feature extraction, Knowledge Graph, Attention-based GNN, Box Embeddings, Rule-based Explanation	Delivered adaptive and explainable visualization recommendations. Attention mechanism enabled fine-grained interpretability of results.	Accuracy : 0.8216 Best overall recommendation
2.	ADVISor – Automatic Visualization Answer for Natural-Language Question on Tabular Data (IEEE PacificVis 2021)	**Datasets:** WikiSQL (80,654 questions, 24,241 tables) **Model:** BERT encoding, Attention classifiers, Softmax aggregation type, Rule-based chart selection and annotation	Converted natural language queries to visualizations with high accuracy. Outperformed baseline systems in attribute, aggregation, and chart selection.	Accuracy 85.3%, query mapping 83.1%

Comparative Analysis

Sr No.	Paper Title	Model And Datasets	Finding/Findings	Statistical Parameter
3.	AUTOVI – Empowering Effective Tracing and Visualizations with AI (IEEE IV 2024)	**Datasets:** 25 Kaggle datasets (manual labeling); **Model**: MLP classifier, Modular web UI (metadata extraction, feedback, profile reports)	Automated preprocessing + visualization, reducing manual effort. Feedback mechanism made it adaptable and more effective over time.	Training Accuracy ≈ 90% (chart classification) Visualization time: 3.7 min vs 43.3 min (manual)
4.	DashBot – Insight-Driven Dashboard Generation Based on Deep Reinforcement Learning (IEEE Transactions on Visualization and Computer Graphics 2023)	**Datasets:** Vega datasets collection(cars, jobs, penguins, Seattle weather, movies); **Model:** DRL (A3C),, Bi-LSTM	Used deep reinforcement learning to generate more insightful dashboards. Achieved diversity and quality without large labeled datasets.	78% of ratings preferred DashBot dashboards (overall quality)

Comparative Analysis

Sr No.	Paper Title	Model And Datasets	Finding/Findings	Statistical Parameter
5.	MultiVision – Designing Analytical Dashboards With Deep Learning-Based Recommendation (IEEE Transactions on Visualization and Computer Graphics 2022)	**Datasets:** Table2Charts corpus (3.9M pairs), dataset, chart); **Model**: Siamese Neural Network, Bi-LSTM (single/multi-chart), Learning-to-rank	Created non-redundant, diverse dashboards by combining deep models. User studies confirmed improved relevance and diversity of dashboards.	Single-chart accuracy = 97.86%, Multi-chart accuracy = 94.05% Top-10 Recall = 87%
6.	Waltzboard – Multi-Criteria Automated Dashboard Design for Exploratory Analysis (IEEE PacificVis 2025)	**Datasets:** 6 Vega datasets; **Model**: Multi-criteria optimization (MILP), 3-phase search (sampling, search, refinement)	Optimized dashboards using multiple evaluation criteria (specificity, diversity, etc.). User studies validated balanced and effective exploratory analysis.	37% improvement over random baseline

Comparative Analysis

Sr No.	Paper Title	Model And Datasets	Finding/Findings	Statistical Parameter
7.	VIS+AI – Integrating Visualization with Artificial Intelligence for Efficient Data Analysis (2023, Review Paper)	**Datasets**: Survey of 113 papers **Model**: 3-loop framework (VIS4AI, AI4VIS, VIS+AI)	Proposed a conceptual framework integrating AI & visualization (AI4VIS, VIS4AI). Highlighted future opportunities for explainable and interactive analytics.	Not Available
8.	Data2Vis – Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks (2019, IEEE)	**Datasets**: Trained on thousands of dataset–visualization pairs from Vega-Lite examples. **Model:** Sequence-to-Sequence RNN (LSTM); Encoder for dataset schema, Decoder for Vega-Lite specification.	generate valid Vega-Lite specifications in 80–90% cases. Proved feasibility of end-to-end neural models for visualization recommendation.	80–90% of generated charts were syntactically valid

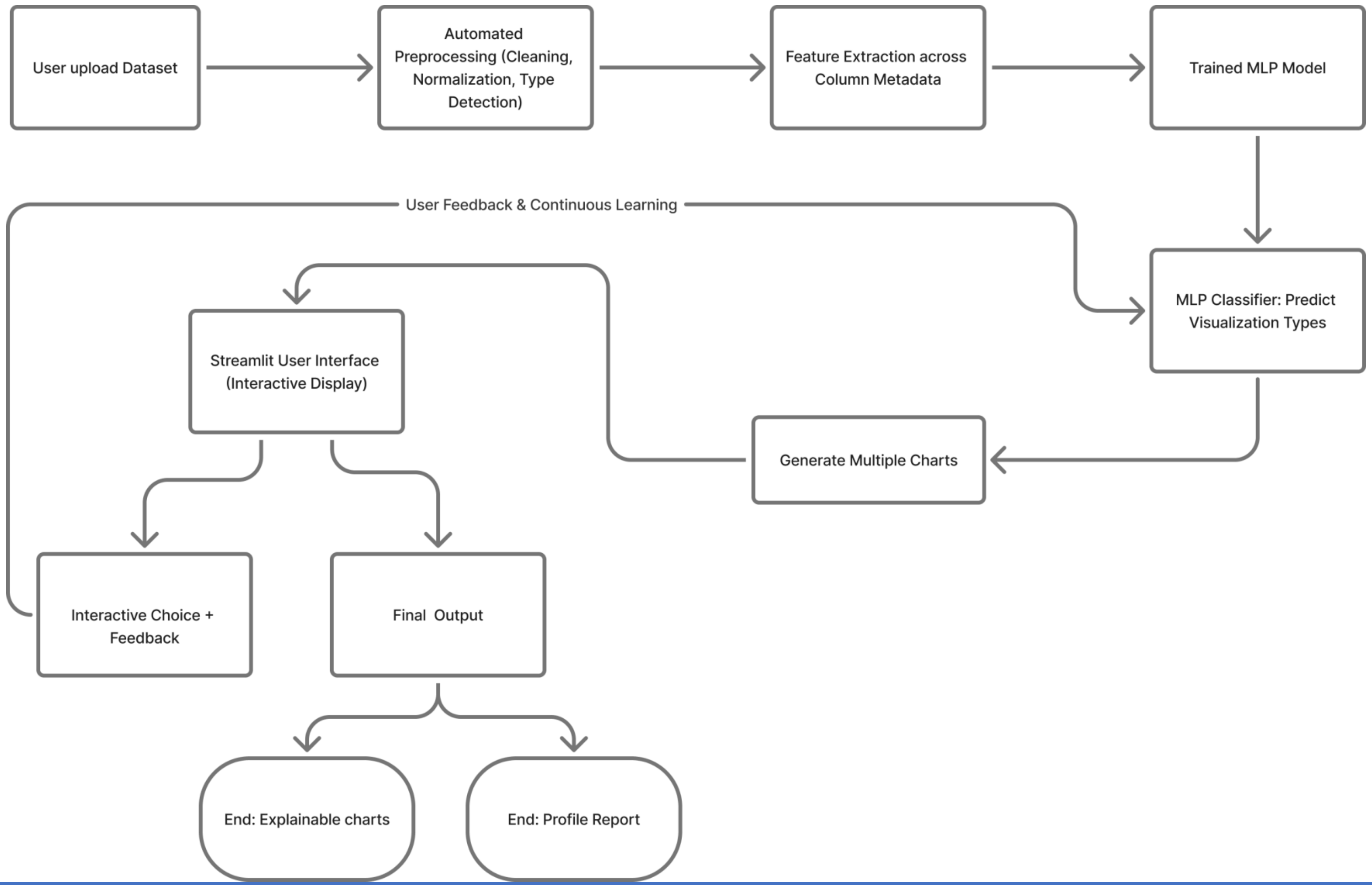
Identified Research Gap

- **Limited preprocessing** – Only AUTOVI provides some preprocessing, but it is not robust; all other systems lack automated preprocessing [3], [1], [2], [5], [4], [7], [6], [8].
- **Restricted visualization support** – Many systems only generate basic charts and do not support complex, expressive, or customized visualizations [1], [5], [7].
- **Scalability issues** – Several systems face difficulty handling large, high-dimensional, or complex datasets, reducing real-world applicability [3], [4], [6], [7].
- **Low interactivity** – Dashboards and visualizations are mostly static, with little support for dynamic exploration or cross-chart interactions [4], [7].
- **Explainability challenges** – Some systems (e.g., AUTOVI, AdaVis) provide partial traces, but most do not offer clear reasoning behind visualization or dashboard recommendations [3], [1], [8].

Problem Statement

Existing AI-powered visualization systems are limited in handling university data effectively. They offer weak preprocessing, generate only basic visualizations, and lack clear explanations for their results. These systems also struggle with large or complex datasets and provide little interactivity for dynamic data exploration. Hence, there is a need for an AI-driven framework that can automatically preprocess data, create interactive visualizations, and deliver explainable insights to support data-driven decision-making in universities.

Proposed Flowchart



Implementation details of existing solutions

Imported Libraries and modules

```
from google.colab import drive
drive.mount('/content/drive')


DATA_PATH = "/content/drive/MyDrive/autovi_data"
OUTPUT_PATH = "/content/drive/MyDrive/autovi_csv"
COMBINED_FILE = os.path.join(OUTPUT_PATH, "combined_info.csv")

os.makedirs(OUTPUT_PATH, exist_ok=True)

import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import joblib
```

Implementation details of existing solutions

Data Loading and Initialization

```
DATA_PATH = "/content/drive/MyDrive/autovi_data"
OUTPUT_PATH = "/content/drive/MyDrive/autovi_json"
os.makedirs(OUTPUT_PATH, exist_ok=True)


def preprocess_dataset(file_path, output_path):
    try:
        # Handle CSV and Excel
        if file_path.endswith('.csv'):
            df = pd.read_csv(file_path, encoding='ISO-8859-1', low_memory=False)
        elif file_path.endswith(('.xls', '.xlsx')):
            df = pd.read_excel(file_path)
        else:
            print(f" Unsupported file type: {file_path}")
            return
    except Exception as e:
        print(f"Could not load {file_path}: {e}")
        return
```

Implementation details of existing solutions

Initializes dataset metadata (name, rows, columns)

```python
dataset_name = os.path.splitext(os.path.basename(file_path))[0]
rows = []

for col in df.columns:
    s = df[col]
    col_info = {
        "Dataset": dataset_name,
        "Column Name": col,
        "Data Type": str(s.dtype),
        "Num Missing": int(s.isnull().sum()),
        "Num Unique": int(s.nunique(dropna=True)),
    }

    col_info["Unique Ratio"] = float(s.nunique(dropna=True) / len(s)) if len(s) > 0 else 0
```

Implementation details of existing solutions

Automated Data Profiling and Visualization Suggestion

```
if pd.api.types.is_numeric_dtype(s):
    s_filled = s.fillna(0)

    col_info.update({
        "Mean": float(s_filled.mean()),
        "Std": float(s_filled.std()),
        "Min": float(s_filled.min()),
        "Max": float(s_filled.max()),
        "Skew": float(s_filled.skew()),
        "Kurtosis": float(s_filled.kurtosis()),
        "P25": float(s_filled.quantile(0.25)),
        "P50": float(s_filled.median()),
        "P75": float(s_filled.quantile(0.75)),
        "Top Values": None
    })
```

```
else:
    s_filled = s.fillna("missing").astype(str)
    top_values = s_filled.value_counts().head(5).to_dict()

    col_info.update({
        "Mean": None,
        "Std": None,
        "Min": None,
        "Max": None,
        "Skew": None,
        "Kurtosis": None,
        "P25": None,
        "P50": None,
        "P75": None,
        "Top Values": str(top_values)
    })
```

Implementation details of existing solutions

Automated Dataset Metadata Extraction and Combination Module

```
    csv_file = os.path.join(output_path, f"{dataset_name}_metadata.csv")
    result_df.to_csv(csv_file, index=False)
    print(f" Processed {file_path}, saved CSV at {csv_file}")

    return result_df


all_results = []

for file in os.listdir(DATA_PATH):
    if file.endswith(('.csv', '.xls', '.xlsx')):
        file_path = os.path.join(DATA_PATH, file)
        result_df = preprocess_dataset_to_csv(file_path, OUTPUT_PATH)
        if not result_df.empty:
            all_results.append(result_df)

# Combine all dataset metadata into one master file
if all_results:
    combined_df = pd.concat(all_results, ignore_index=True)
    combined_df.to_csv(COMBINED_FILE, index=False)
    print(f"\nCombined all metadata files into {COMBINED_FILE}")
    print(f"Total rows: {len(combined_df)} | Columns: {len(combined_df.columns)}")
else:
    print(" No datasets processed.")
```

Implementation details of existing solutions

Dataset Management and Batch Processing Module

```
CSV_PATH = "/content/drive/MyDrive/autovi_csv"
MLP_FEATURE_PATH = "/content/drive/MyDrive/autovi_features"
os.makedirs(MLP_FEATURE_PATH, exist_ok=True)


def csv_to_features(csv_file):
    df = pd.read_csv(csv_file)

    # Keep only required columns
    numeric_keys = [
        "Mean", "Std", "Min", "Max", "Skew", "Kurtosis",
        "P25", "P50", "P75", "Num Missing", "Num Unique", "Unique Ratio"
    ]
    categorical_keys = ["Data Type", "Predicted Visualization"]

    # Fill missing numeric values with 0
    df[numeric_keys] = df[numeric_keys].fillna(0)

    # Extract numeric and categorical data
    numeric_features = df[numeric_keys].to_numpy()
    categorical_data = df[categorical_keys].astype(str).to_numpy()

    # One-hot encode categorical columns
    encoder = OneHotEncoder(sparse_output=False, handle_unknown="ignore")
    cat_encoded = encoder.fit_transform(categorical_data)

    # Combine numeric + categorical features
    combined_features = np.hstack([numeric_features, cat_encoded])

    # Standardize all combined features
    scaler = StandardScaler()
    features_scaled = scaler.fit_transform(combined_features)

    # Create output DataFrame
    feature_df = pd.DataFrame(features_scaled, index=df["Column Name"])

    dataset_name = os.path.splitext(os.path.basename(csv_file))[0]
    feature_file = os.path.join(MLP_FEATURE_PATH, f"{dataset_name}_features.csv")
    feature_df.to_csv(feature_file)

    print(f"✅ Features saved for {dataset_name} → {feature_file}")

    return feature_df

for file in os.listdir(CSV_PATH):
    if file.endswith("_metadata.csv"):
        csv_file = os.path.join(CSV_PATH, file)
        csv_to_features(csv_file)
```

Implementation details of existing solutions

Automatic Detection of Visualization Label Column

```
REPORTS_PATH = "/content/drive/MyDrive/autovi_csv/combined_info.csv"   # Combined m
MODEL_PATH = "/content/drive/MyDrive/autovi_model"
os.makedirs(MODEL_PATH, exist_ok=True)

if not os.path.exists(REPORTS_PATH):
    raise FileNotFoundError("combined_info.csv not found in autovi_csv folder!")

combined_df = pd.read_csv(REPORTS_PATH)
print(f"Loaded combined metadata file — Total rows: {len(combined_df)}")

possible_label_cols = [
    "Predicted Visualization", "Visualization Label",
    "visualization_label", "Predicted_Visualization"
]
label_col = None
for col in combined_df.columns:
    if col.strip() in possible_label_cols:
        label_col = col
        break

if not label_col:
    raise ValueError(" No valid visualization label column found!")

print(f"Using '{label_col}' as visualization label column.")
```

Implementation details of existing solutions

MLP Model Training Setup and Data Preprocessing

```
X = np.hstack([num_features.values, cat_encoded])
le = LabelEncoder()
y = le.fit_transform(combined_df[label_col].astype(str))


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

mlp = MLPClassifier(
    hidden_layer_sizes=(64, 32),
    activation='relu',
    solver='adam',
    batch_size=32,
    max_iter=500,
    random_state=42,
    learning_rate_init=0.001,
    early_stopping=True
)

# Train the model
mlp.fit(X_train, y_train)
```

Implementation details of existing solutions

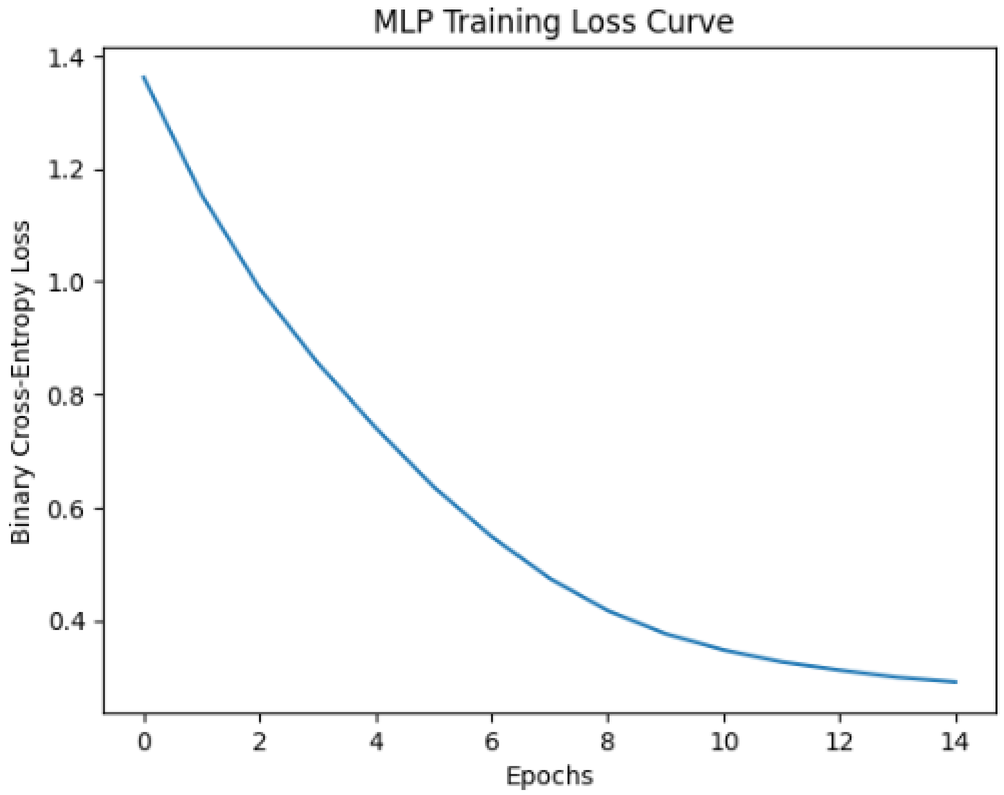
Model Evaluation and Loss Curve Visualization

```
# Evaluate
y_pred = mlp.predict(X_test)
y_prob = mlp.predict_proba(X_test)

acc = accuracy_score(y_test, y_pred)
print(f"\n Model Accuracy: {acc:.2f}")
print("\nClassification Report:\n", classification_report(y_test, y_pred, target_names=le.classes_))

# Show example probabilities
print("\nExample output probabilities (sigmoid layer):")
for cls, prob in zip(le.classes_, y_prob[0]):
    print(f"{cls}: {prob:.3f}")

# Plot loss curve
try:
    plt.plot(mlp.loss_curve_)
    plt.title("AUTOVI MLP Training Loss Curve")
    plt.xlabel("Epochs")
    plt.ylabel("Binary Cross-Entropy Loss")
    plt.show()

except Exception as e:
    print(f"(Plot skipped: {e})")
```

Results and Discussion

Model Evaluation Results and MLP Training Performance

```
Loaded combined metadata file — Total rows: 412
Using 'Predicted Visualization' as visualization label column.

 Model Accuracy: 0.86

Classification Report:
                   precision    recall  f1-score   support

             bar       1.00      0.92      0.96        38
binned_histogram       0.00      0.00      0.00         8
       histogram       0.77      0.96      0.86        28
            none       0.69      1.00      0.82         9

        accuracy                           0.86        83
       macro avg       0.62      0.72      0.66        83
    weighted avg       0.79      0.86      0.82        83


Example output probabilities (sigmoid layer):
bar: 0.245
binned_histogram: 0.200
histogram: 0.368
none: 0.187
```



Objectives of the work

- To design a preprocessing pipeline capable of efficiently handling messy and large-scale tabular datasets.
- To extend visualization capabilities beyond basic charts to multi-chart and interactive dashboards.
- To enhance explainability by integrating reasoning behind each recommended visualization.
- To develop a **user-friendly and interactive system** that enables real-time data analysis and exploration

Timeline Chart

No.	Task	2025				
		July	August	September	October	November
1.	Problem Finding					
2.	Literature Survey					
3.	Research Direction and Define Proposed Methodology					
4.	Implementation of Existing Solutions					
5.	Implementation of Proposed Solutions					
6.	Conclusion					
7.	Final Review and Paper Writing					
8.	Thesis Writing					

Figure 1: Timeline chart

References

1. Zhang, S., Li, H., Qu, H., & Wang, Y. (2024). *AdaVis – Adaptive and explainable visualization recommendation for tabular data*. *IEEE Transactions on Visualization and Computer Graphics*. https://doi.org/10.1109/TVCG.2024.3388553
2. Wang, Q., Luo, Y., Zhang, Y., & Qu, H. (2021). *ADVISor: Automatic visualization answer for natural-language question on tabular data*. In *IEEE Pacific Visualization Symposium (PacificVis)* (pp. 91–100). *IEEE*. https://doi.org/10.1109/PacificVis52677.2021.00010
3. Xie, J., Zhang, X., Huang, W., & Chen, W. (2024). *AUTOVI: Empowering effective tracing and visualizations with AI*. In *IEEE International Conference on Information Visualization (IV)*. *IEEE*. https://doi.org/10.1109/IV64223.2024.00057
4. Zhou, J., Luo, Y., Wang, Q., & Qu, H. (2023). *DashBot: Insight-driven dashboard generation based on deep reinforcement learning*. *IEEE Transactions on Visualization and Computer Graphics, 29*(1), 690–700. https://doi.org/10.1109/TVCG.2023.3314826
5. Zhou, J., Luo, Y., Wang, Q., & Qu, H. (2022). *MultiVision: Designing analytical dashboards with deep learning-based recommendation*. *IEEE Transactions on Visualization and Computer Graphics, 28*(1), 279–289. https://doi.org/10.1109/TVCG.2021.3114826

References

6. Li, G., Li, R., Feng, Y., Zhang, Y., Luo, Y., & Liu, C. H. (2024). *CoInsight: Visual storytelling for hierarchical tables with connected insights*. *IEEE Transactions on Visualization and Computer Graphics*. https://doi.org/10.1109/TVCG.2024.3388553
7. Kim, J., Lee, H., & Choi, J. (2025). *Waltzboard: Multi-criteria automated dashboard design for exploratory analysis*. In *IEEE Pacific Visualization Symposium (PacificVis)*. *IEEE*. https://doi.org/10.1109/PacificVis64226.2025.00032
8. Chen, M., & Zhang, L. (2023). *VIS+AI: Integrating visualization with artificial intelligence for efficient data analysis*. *Journal of Visualization Research, 2*(4), 1–15.

THANK YOU