Data Mining Final Practical File

24/48029

Vivaan Singh Adhikari November 24, 2025

Assignment Overview

Document Contents:

Assignment Details:

- **Course:** Data Mining DSE
- **Instructor:** Prof. Archana Gahalaut
- **Hardware:** No specifications
- **Software:** Python, Pandas, Typst(documentation)

Introduction

This assignment entails my solutions to the question assigned as per the course's guidelines. All the final files are available on https://github.com/user7537/coursework/

Database used : https://archive.ics.uci.edu/dataset/28/japanese+credit+screening

Code

Q2: Apply data pre-processing techniques such as standardization/normalization, transformation, aggregation, discretization/binarization, sampling etc. on any dataset

```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler, MinMaxScaler,
KBinsDiscretizer, OneHotEncoder

df = pd.read_csv("japanese_credit_screening/crx.data",
                 header=None,
                 na_values="?")

col_names = [f"A{i}" for i in range(1, 17)]
df.columns = col_names


# numerical columns
num_cols = ["A2","A3","A8","A11","A14","A15"]
# categorical columns
cat_cols = [c for c in df.columns if c not in num_cols]

df[num_cols] = df[num_cols].apply(pd.to_numeric, errors="coerce")
df[num_cols] = df[num_cols].fillna(df[num_cols].median())
df[cat_cols] = df[cat_cols].fillna(df[cat_cols].mode().iloc[0])


std_df = df.copy()
scaler = StandardScaler()
std_df[num_cols] = scaler.fit_transform(std_df[num_cols])
std_df.to_csv("q2_standardized.csv", index=False)


norm_df = df.copy()
norm_df[num_cols] = MinMaxScaler().fit_transform(norm_df[num_cols])
norm_df.to_csv("q2_normalized.csv", index=False)

trans_df = df.copy()
for col in num_cols:
    trans_df[col] = np.log1p(trans_df[col] - trans_df[col].min() + 1)
trans_df.to_csv("q2_log_transformed.csv", index=False)

agg_df = df.copy()
# Group by a categorical variable and compute numeric means
agg = agg_df.groupby("A1")[num_cols].mean()
agg.to_csv("q2_aggregated_by_A1_mean.csv")


disc_df = df.copy()
disc = KBinsDiscretizer(n_bins=3, encode="ordinal",
strategy="quantile")
disc_df[num_cols] = disc.fit_transform(disc_df[num_cols])
disc_df.to_csv("q2_discretized.csv", index=False)
```

```python
    bin_df = df.copy()
    median_A11 = bin_df["A11"].median()
    bin_df["A11_bin"] = (bin_df["A11"] > median_A11).astype(int)
    bin_df.to_csv("q2_binarized_A11.csv", index=False)

    sample_df = df.sample(frac=0.3, random_state=42)
    sample_df.to_csv("q2_sampled.csv", index=False)

    print("Generated all preprocessing outputs.")
```