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/

Code

Q4. Use Naive bayes, K-nearest, and Decision tree classification algorithms to build classifiers on any two datasets. Pre-process the datasets using techniques specified in Q2. Compare the Accuracy, Precision, Recall and F1 measure reported for each dataset using the abovementioned classifiers under the following situations: i. Using Holdout method (Random sampling): a) Training set = 80% Test set = 20% b) Training set = 66.6% (2/3rd of total), Test set = 33.3% ii. Using Cross-Validation: a) 10-fold b) 5-fold

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import train_test_split, cross_validate
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier

df1 = pd.read_csv("japanese_credit_screening/crx.data", header=None,
na_values="?")
df1.columns = [f"A{i}" for i in range(1,17)]
num1 = ["A2","A3","A8","A11","A14","A15"]
cat1 = [c for c in df1.columns if c not in num1]
df1[num1] = df1[num1].apply(pd.to_numeric, errors="coerce")
df1[num1] = df1[num1].fillna(df1[num1].median())
df1[cat1] = df1[cat1].fillna(df1[cat1].mode().iloc[0])
X1 = df1.drop("A16", axis=1)
y1 = df1["A16"]

df2 = pd.read_csv("db2/SouthGermanCredit.asc", sep=" ", header=None)
df2.columns =
["laufkont","laufzeit","moral","verw","hoehe","sparkont","beszeit","rate","famges","buerge","wohnz
X2 = df2.drop("kredit", axis=1)
y2 = df2["kredit"]
num2 =
["laufzeit","hoehe","beszeit","rate","wohnzeit","verm","alter","weitkred","bishkred","pers"]
cat2 = [c for c in X2.columns if c not in num2]

def build_pipeline(X, num, cat):
    ct = ColumnTransformer([
        ("num", StandardScaler(), num),
        ("cat", OneHotEncoder(handle_unknown="ignore"), cat)
    ])
    def transform():
        return ct.fit_transform(X)
    return ct, transform()

ct1, X1t = build_pipeline(X1, num1, cat1)
ct2, X2t = build_pipeline(X2, num2, cat2)

models = {
    "NaiveBayes": GaussianNB(),
    "KNN": KNeighborsClassifier(),
    "DecisionTree": DecisionTreeClassifier()
}
```

```
def eval_model(X, y, model_name, model):
    X_train, X_test, y_train, y_test = train_test_split(X, y,
train_size=0.8, random_state=42)
    model.fit(X_train, y_train)
    p1 = model.score(X_test, y_test)

    X_train, X_test, y_train, y_test = train_test_split(X, y,
train_size=0.666, random_state=42)
    model.fit(X_train, y_train)
    p2 = model.score(X_test, y_test)

    cv10 = cross_validate(model, X, y, cv=10,
scoring=["accuracy","precision_macro","recall_macro","f1_macro"])
    cv5 = cross_validate(model, X, y, cv=5,
scoring=["accuracy","precision_macro","recall_macro","f1_macro"])

    print("====", model_name, "====")
    print("80/20 accuracy:", p1)
    print("66/33 accuracy:", p2)
    print("10-fold:", cv10)
    print("5-fold:", cv5)

print("=== Dataset 1 ===")
for name, m in models.items():
    eval_model(X1t, y1, name, m)

print("=== Dataset 2 ===")
for name, m in models.items():
    eval_model(X2t, y2, name, m)

print("done")
```