CSCI4150U: Data Mining

Lab 03 - Model Selection and Evaluation

Syed Naqvi
Student ID: 100590852

November 1, 2024

Abstract

This report walks through the preprocessing steps, model selection, and evaluation methods used for Naive Bayes, k-Nearest Neighbors, and Decision Tree classifiers to improve accuracy for a dating site recommendation system. Models are trained using a training dataset with the final model selection based on performance metrics evaluated using an unseen test dataset.

1 Introduction

In this experiment, we apply standard data preprocessing techniques, explore feature relationships, and select optimal models for classification using Naive Bayes, k-Nearest Neighbors (K-NN), and Decision Tree algorithms. Each model is evaluated using cross-validation, and the best models are selected based on accuracy and generalizability.

2 Preprocessing and Exploration

Data preprocessing involves standardizing features to ensure they have a mean of zero and a standard deviation of one, enhancing model performance. We visualize the distribution of standardized features and inspect feature pair correlations.

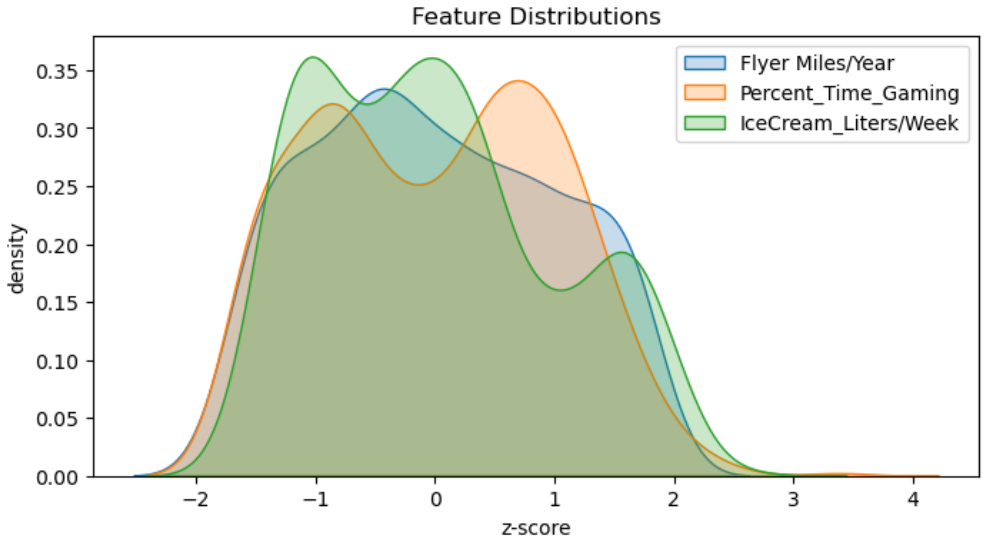

Figure 1: Distribution of Standardized Features

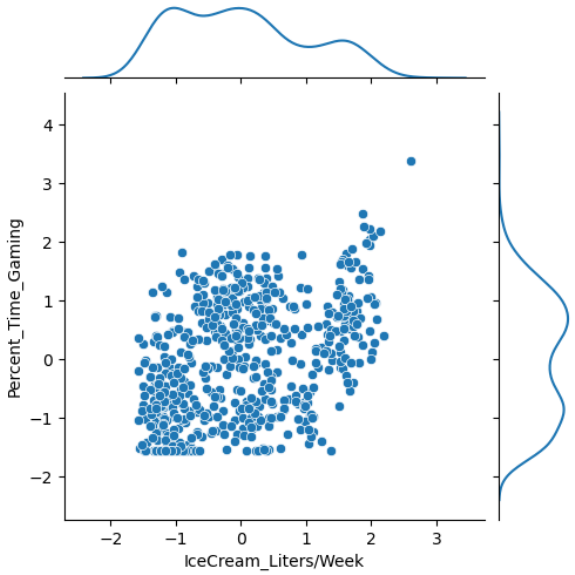

Figure 2: Percentage Time Gaming vs Ice Cream Liters/Week

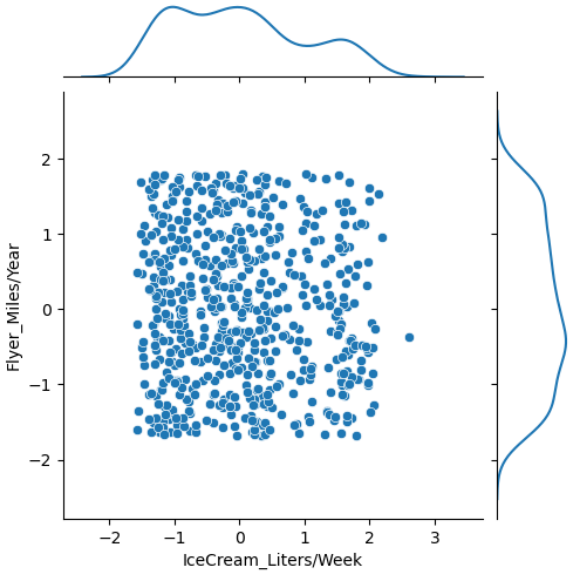

Figure 3: Flyer Miles/Year vs Ice Cream Liters/Week

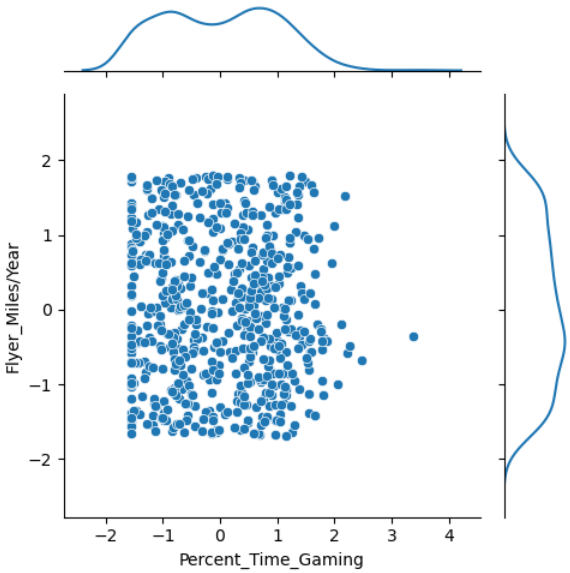

Figure 4: Flyer Miles/Year vs Ice Cream Liters/Week

3 Naive Bayes Classification (Gaussian Distribution)

3.1 Model Validation

Naive Bayes is selected due to minimal correlation between features, suggesting a high degree of independence. The cross-validation results for the Gaussian Naive Bayes model are presented in Figure 5.

```
gnb = GaussianNB()
NaiveBayesResults = cross_validate_df(dating_X, dating_Y,
                                      scoring='accuracy', model=gnb,
                                      folds=10)
display(NaiveBayesResults.mean())
```

✓ 0.0s

```
test_score    0.931667
dtype: float64
```

Figure 5: Naive Bayes Cross Validation Results

4 K-Nearest Neighbors (K-NN) Classification

4.1 Model Selection

To determine the optimal k value, we evaluate k values in the range $[1, 30]$ using 10-fold cross-validation. The average accuracy per k-value across 100 iterations is recorded, with $k = 18$ yielding the highest accuracy.

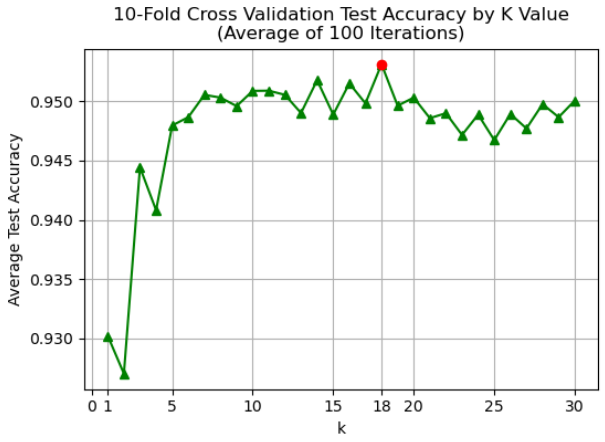

Figure 6: Accuracy vs. K for K-NN

```
numNeighbors = np.arange(1,31)
testAcc = np.zeros(len(numNeighbors))
iterations = 100
for iteration in range(iterations):
    dating_X, dating_Y = shuffle(dating_X, dating_Y)
    for k in numNeighbors:
        clf = KNeighborsClassifier(n_neighbors=k, metric='minkowski', p=2)
        acc = cross_validate_df(model=clf, X=dating_X,
                                Y=dating_Y, scoring='accuracy', folds=10)
        testAcc[k-1] += acc.mean().iloc[0]
    if (iteration%10 == 0):
        print(f"iteration {iteration} completed.")
testAcc /= iterations
```

✓ 3m 27.5s

Figure 7: Code for Best K Selection

5 Decision Tree Classification

5.1 Model Selection

We create decision trees with varying depths, using entropy and Gini impurity measures, and apply 10-fold cross-validation to identify the depth and impurity measure yielding the highest accuracy.

```
maxDepths = [2,3,4,5,6,7,8,9,10,15,20,25,30,35,40,45,50]
testAcc_GINI = np.zeros(shape=(len(maxDepths)))
testAcc_ENTROPY = np.zeros(shape=(len(maxDepths)))
repititions = 100
for rep in range(repititions):
    dating_X, dating_Y, shuffle(dating_X, dating_Y)
    for i,d in enumerate(maxDepths):
        clf_GINI = DecisionTreeClassifier(criterion='gini', max_depth=d)
        clf_ENTROPY = DecisionTreeClassifier(criterion='entropy', max_depth=d)
        gini_acc = cross_validate_df(model=clf_GINI, scoring='accuracy', X=dating_X, Y=dating_Y, folds=10)
        entropy_acc = cross_validate_df(model=clf_ENTROPY, scoring='accuracy', X=dating_X, Y=dating_Y, folds=10)
        testAcc_GINI[i] += gini_acc.mean().iloc[0]
        testAcc_ENTROPY[i] += entropy_acc.mean().iloc[0]
    if(rep%10 == 0):
        print(f"repitition {rep} compete")
testAcc_ENTROPY /= repititions
testAcc_GINI /= repititions
✓ 2m 9.2s
```

Figure 8: Decision Tree Model Selection Code

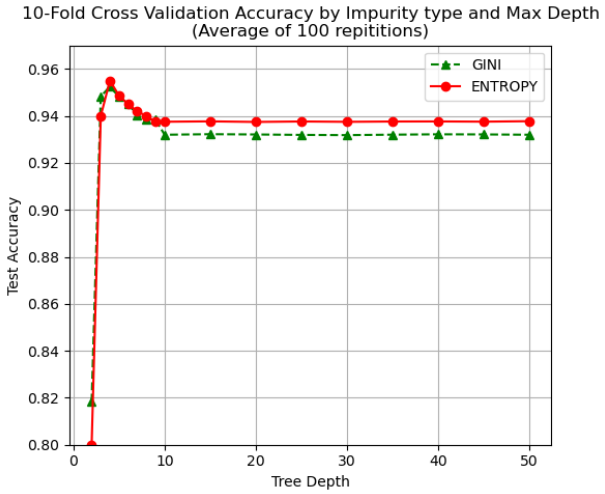

Figure 9: Accuracy by Impurity and Depth

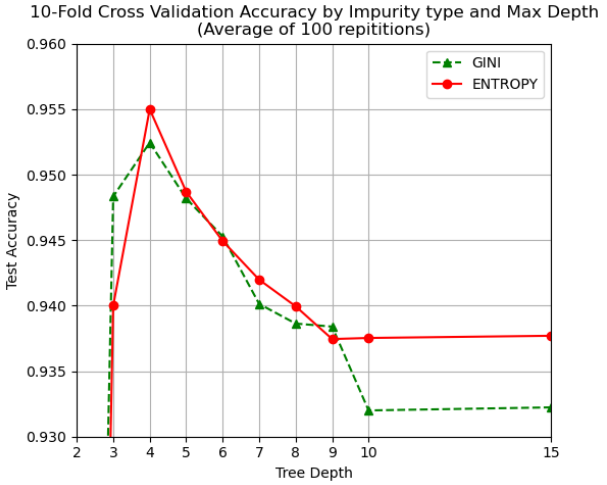

Figure 10: Zoomed-In Accuracy by Impurity and Depth

The optimal decision tree model uses entropy impurity with a maximum depth of 4.

6 Test Set Validation

After model selection, we evaluate the chosen models on the test dataset. The comparison results are shown in Figure 11.

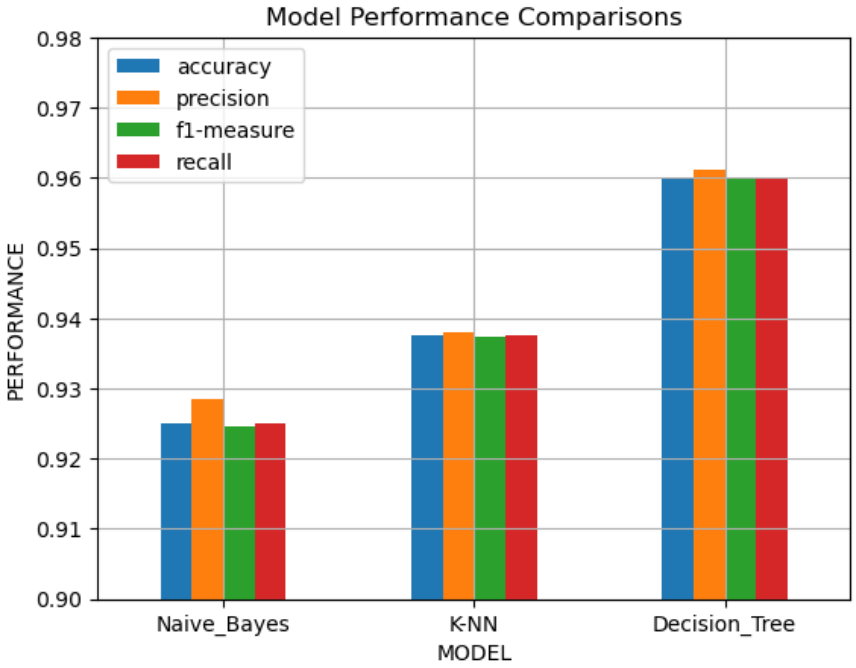

Figure 11: Comparison of Model Performance

	accuracy	precision	f1-measure	recall
Naive_Bayes	0.9250	0.928570	0.924531	0.9250
K-NN	0.9375	0.937960	0.937463	0.9375
Decision_Tree	0.9600	0.961234	0.959920	0.9600

Figure 12: Model Performance Summary

7 Conclusion

Based on the test dataset results, the decision tree model with entropy impurity and a max depth of 4 demonstrates the best classification performance. This model is recommended for predicting customer dating preference based on provided features.