Supervised Classifiers On Enrollment Data

Student Sam Nowak

Overview

This project applies supervised learning classifiers to a dataset to identify the strongest predictive model. I used a variety of classifiers and methods learned throughout the course, including SVM, Decision Tree, KNN, Random Forest, Ensemble, and Naive Bayes. The report also includes an ablation test and learning curve plots to analyze model performance and determine whether classifiers are overfitting or underfitting.

Data

The dataset contains enrollment information from recent years in the Computer Science and Mathematics department at EIU. The goal is to predict whether a student is likely to attend EIU after applying.

Before using the data with classifiers, I cleaned and transformed it into numerical format. I removed irrelevant columns such as `Graduation Term`, `Id`, `Class`, `Gateway`, `First Time Freshman`, and `New Transfer`. For example, `Graduation Term` implies current enrollment, and `Id` is a unique identifier with no predictive value. `Gateway` had a constant value and provided no variation.

I consolidated location-based columns (`STATE`, `ZIP`, `COUNTRY`, `INTERNATIONAL`) into a new `Location` variable. Most students came from Illinois, particularly the Chicagoland area, so I segmented ZIP codes accordingly into more different regions in Illinois.

Grade/test score columns like `HS GPA`, `ACT`, and `SAT` were bucketed into three ranges—low, average, and high—based on historical averages. For example, `HS GPA` was split into [0–3.38], [3.39–3.48], and [3.49–5], mapping to 1 (low), 2 (average), and 3 (high). I handled null values by mapping them to 0 or 1, and mapped financial columns like `EFC/SAI`, `Pell`, `Grant`, and `Scholarships` in increments of 1000. Binary columns with 'Y'/'N' values were converted to 1 and 0.

Building the Classifiers

I created a generalized function that runs a given classifier from scikit-learn with different hyperparameters using k-fold cross-validation. The function returns the best model based on a scoring metric—recall in this case—and displays a confusion matrix to show precision, recall, and F1-score. It also returns the best parameters and performance metrics. We chose recall because it is more important since we don't want to think a student is not coming to EIU, but actually does.

First Results

After testing all classifiers, the top performers in terms of recall were:

- **SMOTE + SVM**: Recall = 88.29%, with parameters:

```
{'C': 10, 'gamma': 1, 'kernel': 'rbf'}
```

- **Random Forest**: Recall = 73.52%, with parameters:

```
{'clf__bootstrap': False, 'clf__class_weight': 'balanced', 'clf__max_depth': 5,
 'clf__max_features': 'sqrt', 'clf__min_samples_leaf': 1, 'clf__min_samples_split': 10,
 'clf__n_estimators': 2000}
```

- **Regular SVM**: Recall = 70.58%, with parameters:

```
{'C': 100, 'gamma': 0.1, 'kernel': 'rbf'}
```

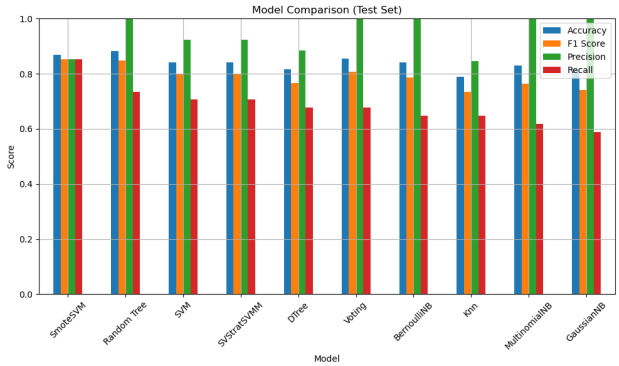

Figure 1: Sorting based on recall performance

Ablation Test

Next, I tested whether certain features were degrading classifiers performance using an ablation test. This involves removing one feature at a time and retraining each model. I then plotted the change in recall for each classifier.

If the change in recall is **above** the x-axis, removing the feature improved performance—indicating that the feature may have introduced noise. If it's **below**, the feature likely added predictive value.

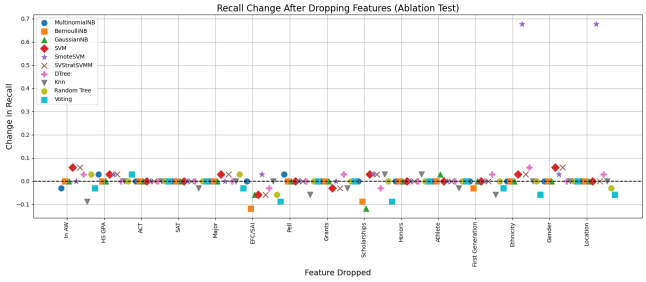

Figure 2: Change in recall after removing features (Ablation Test)

After identifying the most harmful features, I removed them and re-tested the classifiers. However, I found that removing them did not improve any model's overall performance. I re-plotted all the models with the updated classifiers. Note that not all classifiers were re-trained since some had no features that introduced noise.

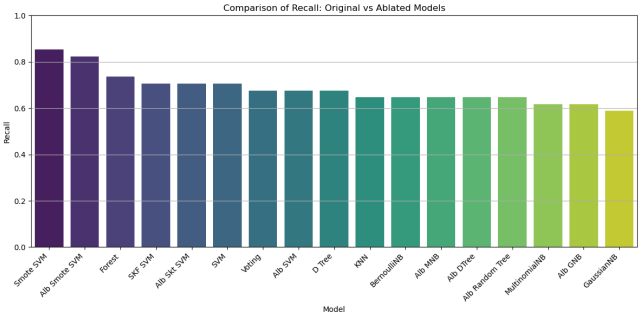

Figure 3: Classifier performance after feature removal

Feature Correlation

I also analyzed feature importance using Random Forest, Decision Tree, and linear correlation. This helps determine which features have the most impact. The rightmost plot shows the linear relationship of each feature with the class labels.

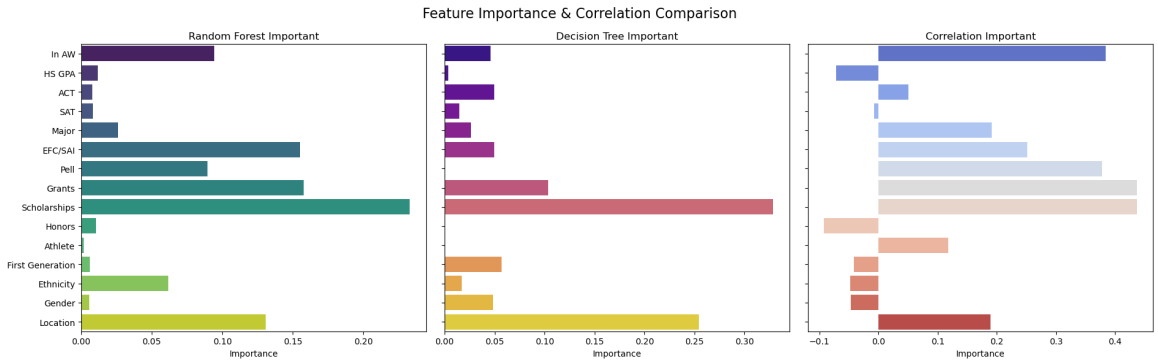

Figure 4: Feature importance and correlation analysis

We can see that Random Forest and Decision Tree assign different importance values compared to linear correlation. This is because some features may effectively split the data but aren't strongly linearly correlated. There may also be interactions between features that can help decide if student will come or not. Therefore some features will have different correction than a general linear test. However most of the feature had similar important throughout the three different plots.

Learning Curve

To evaluate whether our top three models were overfitting or underfitting, I plotted learning curves based on varying training sizes for the SMOTE SVM, Random Forest, and regular SVM.

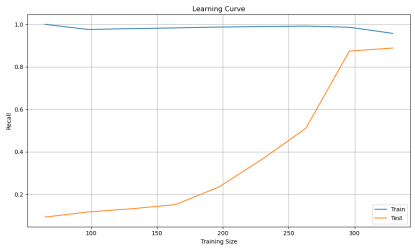

Figure 5: *
(a) SMOTE SVM

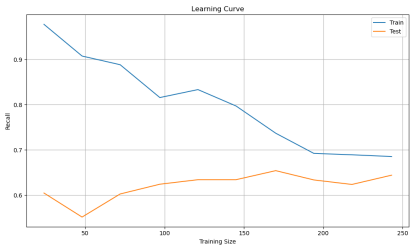

Figure 6: *
(b) Random Forest

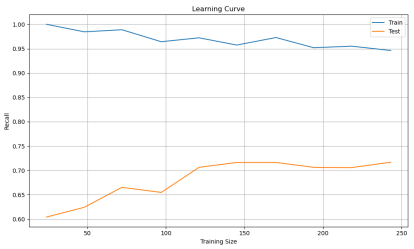

Figure 7: *
(c) Regular SVM

Figure 8: Learning curves for top classifiers

For SMOTE SVM, the training accuracy remains around 95% while the test accuracy increases until it levels off around 85% with 300 training samples. This small gap suggests good generalization. In contrast, Random Forest shows a decrease in training accuracy with little improvement in testing accuracy, indicating underfitting. The regular SVM shows a typical overfitting pattern, with high training performance and stagnant testing performance. Overall, SMOTE SVM shows the best learning behavior and overall performance.

Overall Conclusion

The SMOTE SVM was clearly the best-performing model. It had the highest recall and the most favorable learning curve. While it showed slight signs of overfitting, the small training-test gap suggests it still generalizes well. SMOTE likely helps because the original dataset had fewer than 400 rows, and balancing the class labels with synthetic data improves learning.

Even though we may not yet have a perfect model for predicting enrollment in the Computer Science or Mathematics departments at EIU, we can expect better results as more data becomes available. With continued development and more data, model performance will likely improve significantly.