OESON Data Science Project 3 and 4

Oscar Bowie
Data Science Batch-2023

Introduction

This projects focuses on a dataset of students with the aim to provide an exploratory data analysis, and develop models that predict whether a student is ‘Enrolled’, ‘Dropped Out’, or ‘Graduated’. Furthermore, this project will explore the differences between data preprocessing methods as well as model selection. This will be carried out predominantly through the SKLearn and Pandas libraries.

Dataset

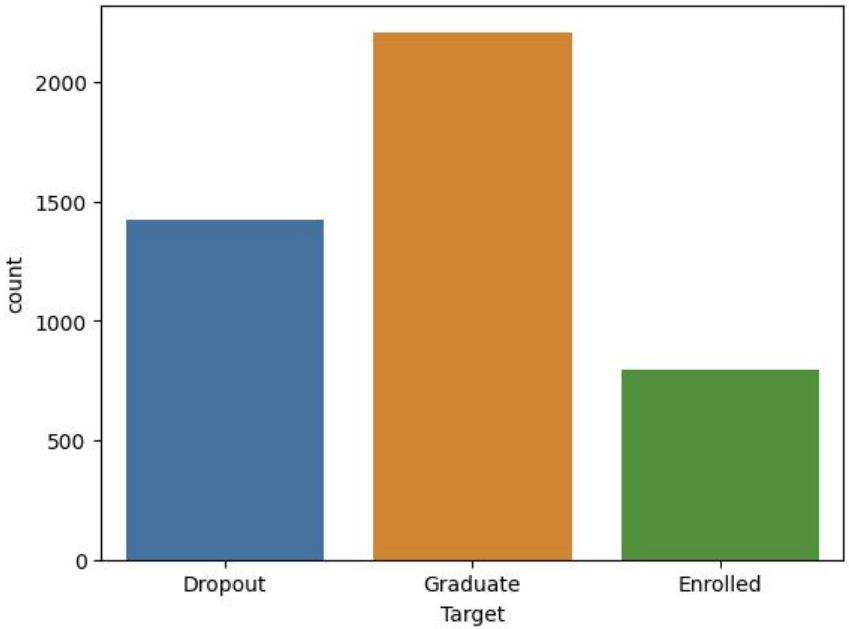
- Total of 34 Columns
- ‘Target’ contains information on graduation status
- Most data formatted as categorical data stored as numeric values for lookup
- Also contains some numeric data: GDP, inflation rate, unemployment rate, etc.

#	Column	Non-Null Count	Dtype
0	Marital status	4424 non-null	int64
1	Application mode	4424 non-null	int64
2	Application order	4424 non-null	int64
3	Course	4424 non-null	int64
4	Daytime/evening attendance	4424 non-null	int64
5	Previous qualification	4424 non-null	int64
6	Nacionality	4424 non-null	int64
7	Mother's qualification	4424 non-null	int64
8	Father's qualification	4424 non-null	int64
9	Mother's occupation	4424 non-null	int64
10	Father's occupation	4424 non-null	int64
11	Displaced	4424 non-null	int64
12	Educational special needs	4424 non-null	int64
13	Debtor	4424 non-null	int64
14	Tuition fees up to date	4424 non-null	int64
15	Gender	4424 non-null	int64
16	Scholarship holder	4424 non-null	int64
17	Age at enrollment	4424 non-null	int64
18	International	4424 non-null	int64
19	Curricular units 1st sem (credited)	4424 non-null	int64
20	Curricular units 1st sem (enrolled)	4424 non-null	int64
21	Curricular units 1st sem (evaluations)	4424 non-null	int64
22	Curricular units 1st sem (approved)	4424 non-null	int64
23	Curricular units 1st sem (grade)	4424 non-null	float64
24	Curricular units 1st sem (without evaluations)	4424 non-null	int64
25	Curricular units 2nd sem (credited)	4424 non-null	int64
26	Curricular units 2nd sem (enrolled)	4424 non-null	int64
27	Curricular units 2nd sem (evaluations)	4424 non-null	int64
28	Curricular units 2nd sem (approved)	4424 non-null	int64
29	Curricular units 2nd sem (grade)	4424 non-null	float64
30	Curricular units 2nd sem (without evaluations)	4424 non-null	int64
31	Unemployment rate	4424 non-null	float64
32	Inflation rate	4424 non-null	float64
33	GDP	4424 non-null	float64
34	Target	4424 non-null	object

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Nacionality	Mother's qualification	Father's qualification	Mother's occupation	...
0	1	8	5	2	1	1	1	13	10	6	...
1	1	6	1	11	1	1	1	1	3	4	...
2	1	1	5	5	1	1	1	22	27	10	...
3	1	8	2	15	1	1	1	23	27	6	...
4	2	12	1	3	0	1	1	22	28	10	...

Target Feature

Around half of all entries classified as ‘Graduate’, with the second most as ‘Dropout’ and the least ‘Enrolled’



Principal Component Analysis

For comparison, two principal component values where selected being 2, and 5 components.

```
X = df.drop('Target', axis=1)
y = df['Target']

pca = PCA(n_components=2)
X = pd.DataFrame(pca.fit_transform(X))
```

```
X = df.drop('Target', axis=1)
y = df['Target']

pca = PCA(n_components=5)
X = pd.DataFrame(pca.fit_transform(X))
```

Machine Learning Methods

- Random Forest
- Gaussian Modelling
- Decision Tree
- Support Vector Machine
- Logarithmic Regression

All methods undergo a 5-fold cross validation accuracy analysis.

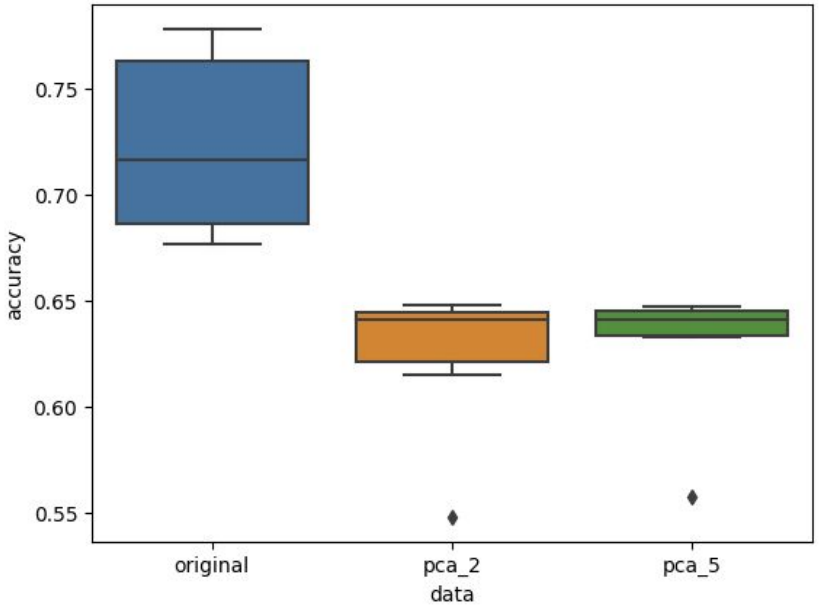
Will retrieve both in sample and out of sample accuracy.

```
kf = KFold(n_splits=5, shuffle=True, random_state=0)
kf.get_n_splits(X)

test_metrics = pd.DataFrame(columns=c)
for i, (train_index, test_index) in enumerate(kf.split(X)):
  random_forest = RandomForestClassifier(n_estimators=100, random_state=0)
  random_forest.fit(X.iloc[train_index], y.iloc[train_index,])

  #train data results
  Y_prediction_train = random_forest.predict(X.iloc[train_index])
  d = pd.DataFrame(metrics.classification_report(y.iloc[train_index],
                   Y_prediction_train, digits=3, output_dict=True))
  d = d.reset_index().rename(columns={'index':'metric'})
  d['fold'] = i
  d['set'] = 'train'
  test_metrics = pd.concat([test_metrics,d])

  #test data results
  Y_prediction_test = random_forest.predict(X.iloc[test_index])
  d = pd.DataFrame(metrics.classification_report(y.iloc[test_index],
                   Y_prediction_test, digits=3, output_dict=True))
  d = d.reset_index().rename(columns={'index':'metric'})
  d['fold'] = i
  d['set'] = 'test'
  test_metrics = pd.concat([test_metrics,d])

#storing results
random_forest_metrics = test_metrics
test_metrics['model'] = 'random_forest'
test_metrics['data'] = 'original'
results = pd.concat([results,test_metrics])
```

Results

Data: Adjusted, Unadjusted, and Accuracy

Original dataset over all the models provided the greatest accuracy by a large margin, followed by 2 Principal Component, with a larger variance, but with an average higher than 5 Principal Component, which had a smaller variance, but a slightly lower average accuracy.

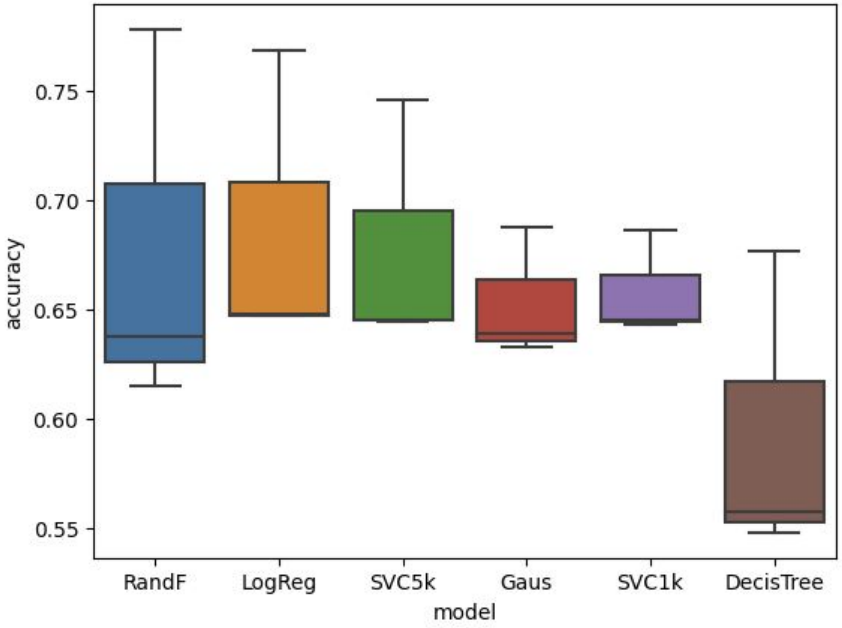


Results

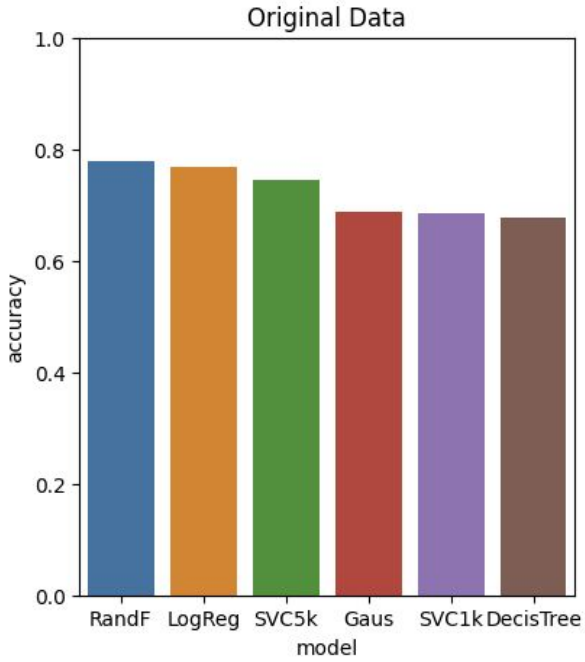
Accuracy between Model Choice

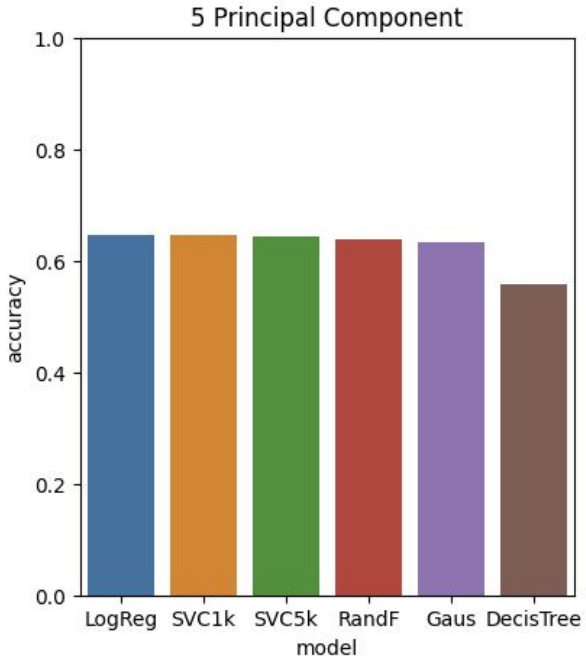
Accuracies produced by the models over all different data types found that Random Forest had the greatest variance in accuracy, however high variance is likely due to the lower performance random forest has on the PCA datasets, and performs best on the original dataset as seen below.

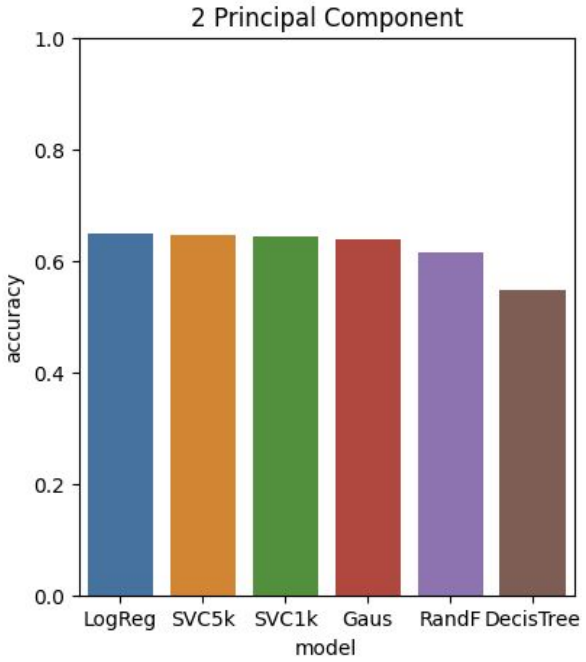
Decision Tree was found to be the worst model for this data, with Support Vector 5000 iterations and 1000 iterations having a small difference, Gaussian performing second worst, and Logarithmic regression performing second best, with a higher average than Random Forest on adjusted data.



Results


Original Data
1.0
0.8
0.6
0.4
0.2
0.0
accuracy
RandF
LogReg
SVC5k
Gaus
SVC1k
DecisTree
model


5 Principal Component
1.0
0.8
0.6
0.4
0.2
0.0
accuracy
LogReg
SVC1k
SVC5k
RandF
Gaus
DecisTree
model


2 Principal Component
1.0
0.8
0.6
0.4
0.2
0.0
accuracy
LogReg
SVC5k
SVC1k
Gaus
RandF
DecisTree
model

Conclusion

Highest accuracy model was random forest on the unadjusted dataset with a accuracy of ~0.8. For PCA altered data, the logarithmic regression and support vector machines out performed the random forest model, however both of these on the performed better on the original data than their PCA counterparts.

This concludes that for this data, the greatest accuracy is achieved using a random forest model on unadjusted data.

Top 10 Models

model	data	set	accuracy
random_forest	original	test	0.778028
logreg	original	test	0.768308
svc_5000	original	test	0.745478
gaussian	original	test	0.687835
svc_1000	original	test	0.686040
decisiontree	original	test	0.676757
logreg	pca_2	test	0.647829
logreg	pca_5	test	0.646924
svc_1000	pca_5	test	0.645341
svc_5000	pca_2	test	0.644889