

# Academic Success Classification Model

---

## Predictive Modeling

*# Importing needed Libraries*

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from catboost import CatBoostClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import HistGradientBoostingClassifier
from sklearn.ensemble import GradientBoostingClassifier
```

*# Load Data*

```
train_data = pd.read_csv('train.csv')
test_data = pd.read_csv('test.csv')
```

*# Split features and target*

```
x = train_data.drop(columns=['id', 'Target'])
y = train_data['Target']
```

*# Encode Target*

```
le = LabelEncoder()
y = le.fit_transform(y)
```

*# Train-test split*

```
x_train, x_val, y_train, y_val = train_test_split(x, y, test_size=
0.2, random_state= 42, stratify= y)
```

*# CatBoostClassifier*

```
catBoost_model = CatBoostClassifier(verbose= 0, random_state= 42)
catBoost_model.fit(x_train, y_train)
catBoost_pred = catBoost_model.predict(x_val)
```

*# Evaluate CatBoost Model*

```
print('CatBoost Accuracy: ', accuracy_score(y_val, catBoost_pred))
print('CatBoost Classification Report:\n',
classification_report(y_val, catBoost_pred))
```

CatBoost Accuracy: 0.8311552535284893

CatBoost Classification Report:

	precision	recall	f1-score	support
0	0.90	0.83	0.86	5059

1	0.66	0.60	0.62	2988
2	0.85	0.93	0.89	7257
accuracy			0.83	15304
macro avg	0.80	0.79	0.79	15304
weighted avg	0.83	0.83	0.83	15304

*# XGBoost Classifier*

```
xgBoost_model = XGBClassifier(use_label_encoder= False, eval_metric=
'logloss', random_state= 42)
xgBoost_model.fit(x_train,y_train)
xgBoost_pred = xgBoost_model.predict(x_val)
```

*# Evaluate XGBoost Model*

```
print('XGBoost Accuracy: ', accuracy_score(y_val, xgBoost_pred))
print('XGBoost Classification Report:\n', classification_report(y_val,
xgBoost_pred))
```

C:\Users\HP\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11\_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\xgboost\core.py:158: UserWarning: [20:01:47] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0c55ff5f71b100e98-1\xgboost\xgboost-ci-windows\src\learner.cc:740: Parameters: { "use\_label\_encoder" } are not used.

```
warnings.warn(msg, UserWarning)
```

XGBoost Accuracy: 0.8297177208572922

XGBoost Classification Report:

	precision	recall	f1-score	support
0	0.89	0.83	0.86	5059
1	0.65	0.60	0.63	2988
2	0.85	0.92	0.89	7257
accuracy			0.83	15304
macro avg	0.80	0.79	0.79	15304
weighted avg	0.83	0.83	0.83	15304

*# We have recieved best accuracy score for both models, still we can try one more model.*

*# HistGradientBoostingClassifier*

```
hist_model = HistGradientBoostingClassifier(random_state=42)
hist_model.fit(x_train,y_train)
hist_pred = hist_model.predict(x_val)
```

*# Evaluate Hist model*

```
print('HistGradientBoostingClassifier Accuracy: ',
```

```
accuracy_score(y_val, hist_pred))
print('HistGradientBoostingClassifier Classification Report: \n',
classification_report(y_val, hist_pred))
```

HistGradientBoostingClassifier Accuracy: 0.8279534762153685

HistGradientBoostingClassifier Classification Report:

	precision	recall	f1-score	support
0	0.90	0.83	0.86	5059
1	0.65	0.60	0.62	2988
2	0.85	0.92	0.89	7257
accuracy			0.83	15304
macro avg	0.80	0.78	0.79	15304
weighted avg	0.83	0.83	0.83	15304

*# We will use CatBoost model as we have recieved highest accuracy score from that model.*

```
test_vals = test_data.drop(columns=['id'])
```

```
catBoost_test_pred = catBoost_model.predict(test_vals)
```

```
catBoost_test_pred = le.inverse_transform(catBoost_test_pred)
```

```
catBoost_test_pred
```

```
C:\Users\HP\AppData\Local\Packages\
PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-
packages\Python311\site-packages\sklearn\preprocessing\_label.py:153:
DataConversionWarning: A column-vector y was passed when a 1d array
was expected. Please change the shape of y to (n_samples, ), for
example using ravel().
```

```
y = column_or_1d(y, warn=True)
```

```
array(['Dropout', 'Graduate', 'Graduate', ..., 'Dropout', 'Dropout',
'Dropout'], dtype=object)
```

```
output = pd.DataFrame({'id': test_data['id'], 'Target':
```

```
catBoost_test_pred})
```

```
output.head()
```

	id	Target
0	76518	Dropout
1	76519	Graduate
2	76520	Graduate
3	76521	Graduate
4	76522	Enrolled

```
output.to_csv('submission.csv', index= False)
```