

Project Name

COVID-19 Patient's Stay

Domain - Machine Learning

Group Information

1. **Vithika Pungliya**
2. **Adwait Bhosale**
3. **Chetan Chaudhari**



GitHub

Github link :

https://github.com/RoboSpark-2021/robospark-2021-FT-Covid_Stay

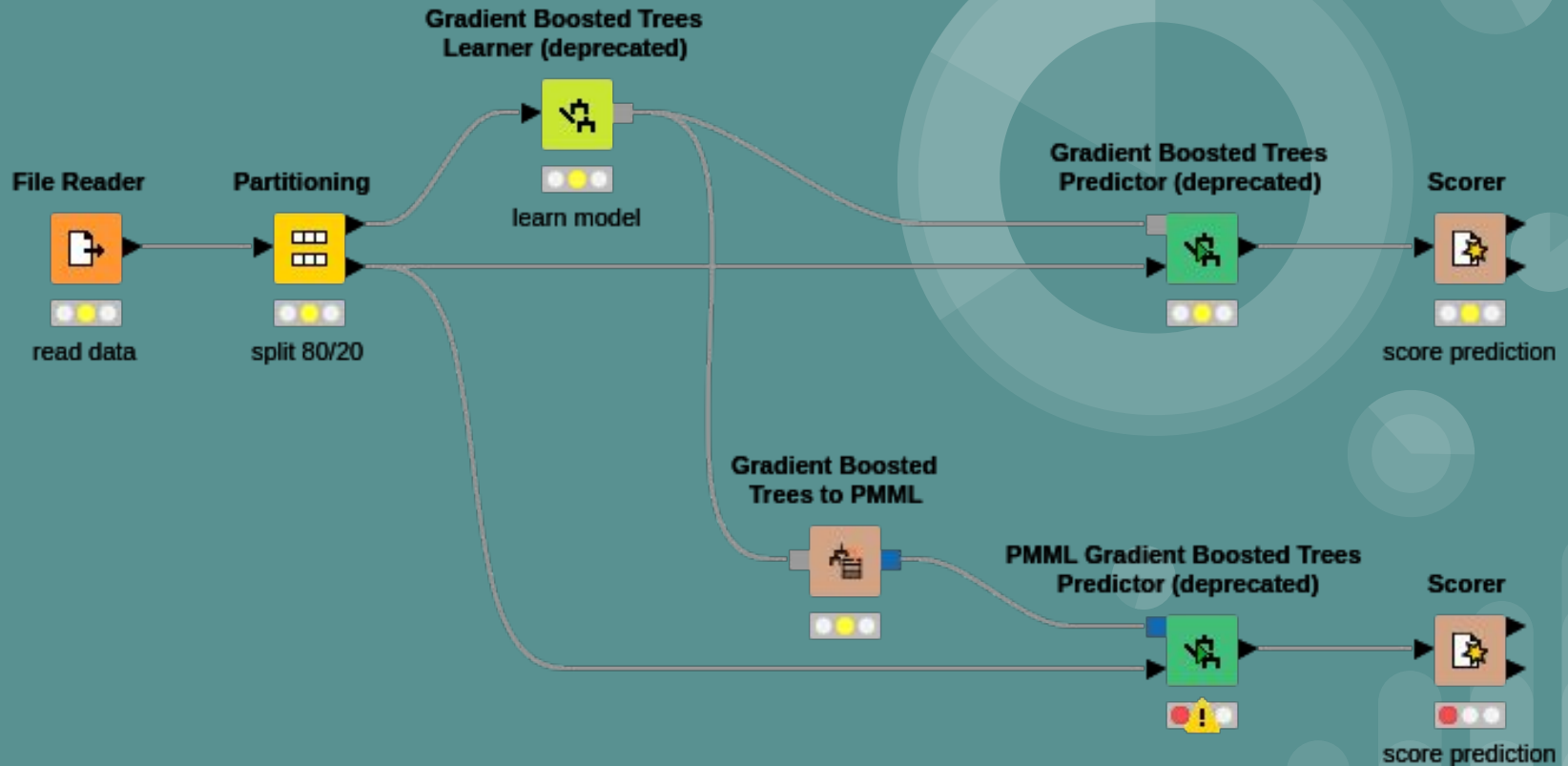
Project Algorithm

- 1.Gradient Boosting Classifier
- 2.Hist Gradient Boosting Classifier
- 3.Ridge Classifier
- 4.Random Forest Classifier

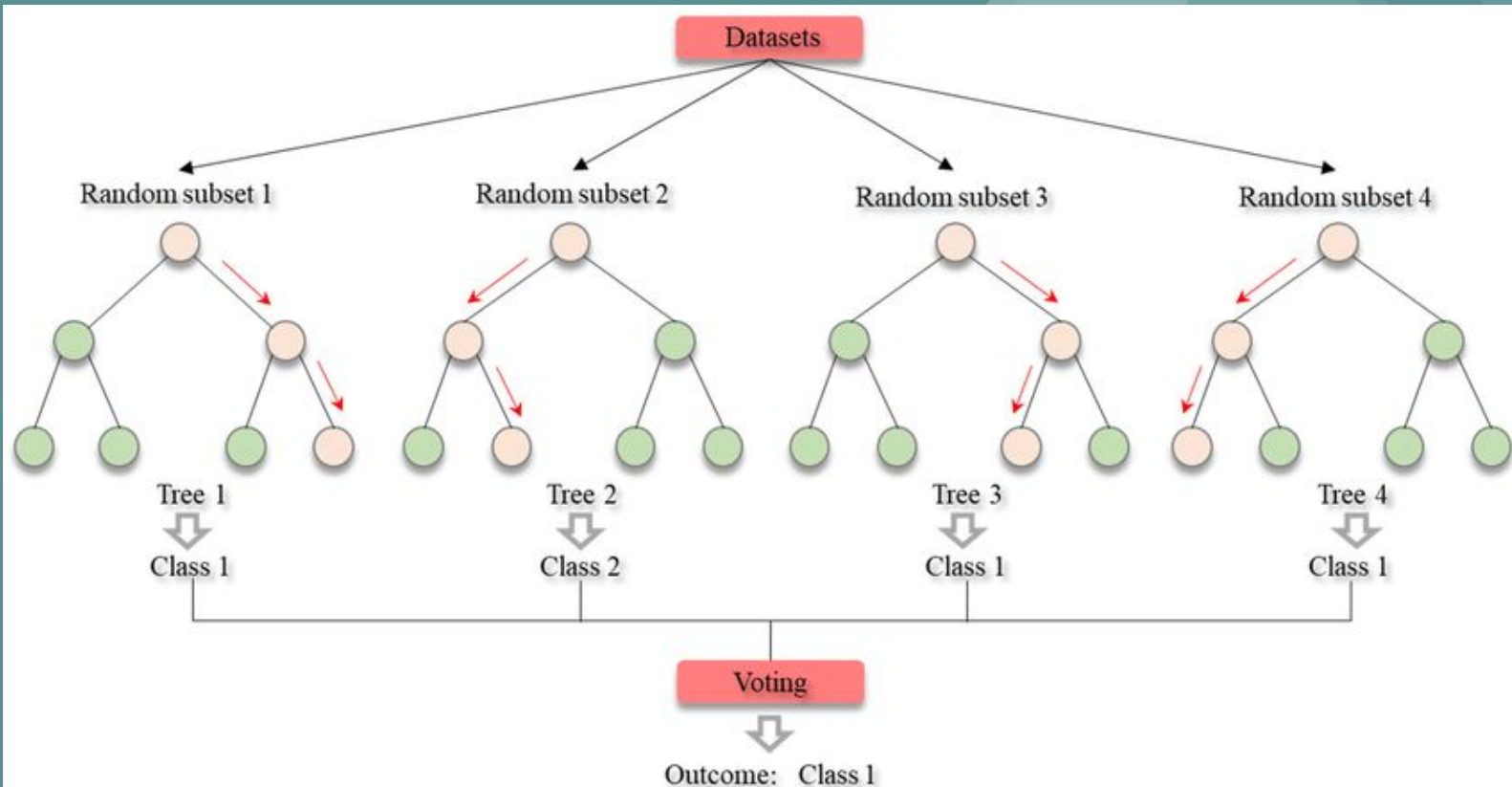
Project Algorithm

- 1.Gradient Boosting Classifier
- 2.Hist Gradient Boosting Classifier
- 3.Ridge Classifier
- 4.Random Forest Classifier

Gradient Boosting Classifier



Random Forest Classifier



Problems Faced

- **What amount of the data to be taken ?**
- **Which columns to be taken ?**
- **Required lot of time to tune hyperparameter.**

Alternative solutions

- **We can use implement the same project using deep learning and neural networks.**

Code Snippets

Using the Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
model_3=RandomForestClassifier()
model_3.fit(X_train,y_train)

RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                        criterion='gini', max_depth=None, max_features='auto',
                        max_leaf_nodes=None, max_samples=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, n_estimators=100,
                        n_jobs=None, oob_score=False, random_state=None,
                        verbose=0, warm_start=False)
```

```
[213] model_3.score(X_test,y_test)

0.3177966101694915
```

Using the Ridge Classifier

```
[216] from sklearn.linear_model import RidgeClassifier
model_5=RidgeClassifier()
model_5.fit(X_train,y_train)
```

```
RidgeClassifier(alpha=1.0, class_weight=None, copy_X=True, fit_intercept=True,
                max_iter=None, normalize=False, random_state=None,
                solver='auto', tol=0.001)
```

```
[217] model_5.score(X_test,y_test)

0.3495762711864407
```

Using the Gradient Boosting Classifier

```
from sklearn.ensemble import GradientBoostingClassifier
model_6=GradientBoostingClassifier()
model_6.fit(X_train,y_train)

GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                           learning_rate=0.1, loss='deviance', max_depth=3,
                           max_features=None, max_leaf_nodes=None,
                           min_impurity_decrease=0.0, min_impurity_split=None,
                           min_samples_leaf=1, min_samples_split=2,
                           min_weight_fraction_leaf=0.0, n_estimators=100,
                           n_iter_no_change=None, presort='deprecated',
                           random_state=None, subsample=1.0, tol=0.0001,
                           validation_fraction=0.1, verbose=0,
                           warm_start=False)
```

```
[219] model_6.score(X_test,y_test)

0.3728813559322034
```

Hist Gradient Boosting Classifier

```
[222] from sklearn.datasets import make_classification
from sklearn.experimental import enable_hist_gradient_boosting
from sklearn.ensemble import HistGradientBoostingClassifier
```

```
model_7=HistGradientBoostingClassifier()
model_7.fit(X_train,y_train)
```

```
HistGradientBoostingClassifier(l2_regularization=0.0, learning_rate=0.1,
                               loss='auto', max_bins=255, max_depth=None,
                               max_iter=100, max_leaf_nodes=31,
                               min_samples_leaf=20, n_iter_no_change=None,
                               random_state=None, scoring=None, tol=1e-07,
                               validation_fraction=0.1, verbose=0,
                               warm_start=False)
```

Comparison of Accuracies- OUTPUT

