Model Name	Hyperparameters	Score Train/(Validation)/Test
`BalancedBaggingClassifier(DecisionTreeClassifier())`	`'max_samples': 0.3, 'base_estimator__min_samples_split': 2, 'base_estimator__min_samples_leaf': 5, 'base_estimator__max_depth': 14`	0.76/0.69/0.634*
`BalancedBaggingClassifier(DecisionTreeClassifier())`	`'max_samples': 0.5, 'base_estimator__min_samples_split': 2, 'base_estimator__min_samples_leaf': 7, 'base_estimator__max_depth': 30`	0.90/0.69/0.62*
BalancedBagging Classifier with Random Under Sampler	`{'max_samples': 0.5, 'base_estimator__min_samples_split': 2, 'base_estimator__min_samples_leaf': 10, 'base_estimator__max_depth': 20}`	0.82/0.69/0.63556*
CatBoost	(verbose=1, n_estimators=300, depth=3, min_data_in_leaf = 4,\ random_seed=67, learning_rate=0.9, task_type="GPU")	0.73/0.70
CatBoost	(verbose=1, n_estimators=400, depth=5, min_data_in_leaf = 2,\ random_seed=67, learning_rate=0.5, task_type="GPU")	0.75/0.70
CatBoost	(verbose=1, n_estimators=500, depth=10, min_data_in_leaf = 1,\ random_seed=67, learning_rate=0.2, task_type="GPU")	0.90/0.69
Easy ensemble on CatBoost	base_estimator=CatBoostClassifier(verbose=1, n_estimators=100, depth=10, min_data_in_leaf = 1,\ random_seed=67, learning_rate=0.2),n_jobs = 4, random_state=42, sampling_strategy='auto',verbose=1	0.78/0.70
Easy ensemble on CatBoost	(base_estimator=CatBoostClassifier(),n_jobs = 4, random_state=42, sampling_strategy='not majority',verbose=1)	0.77/0.71
Easy ensemble	(base_estimator=CatBoostClassifier(verbose=1,	0.94/0.70

on CatBoost	n_estimators=300, depth=10, min_data_in_leaf = 3,\ random_seed=67, learning_rate=0.5),n_jobs = 4, random_state=42, sampling_strategy='all',verbose=1)	
Balanced Bagging on Decision Tree	(base_estimator=DecisionTreeClassifier(min_weight_fraction_leaf=0.0001,max_depth=10), sampling_strategy='auto', replacement=False, random_state=42, n_jobs = -1, verbose=1, oob_score=1,warm_start=0)	0.72/0.68
Balanced Bagging on Decision Tree	BalancedBaggingClassifier(base_estimator=DecisionTreeClassifier(min_weight_fraction_leaf=0.000001,max_depth=20), sampling_strategy='not majority', replacement=False, random_state=42, n_jobs = -1, verbose=1, oob_score=1,warm_start=0)	0.90/0.66
Balanced Bagging on CatBoost	BalancedBaggingClassifier(CatBoostClassifier(verbose=1, n_estimators=500, depth=10, min_data_in_leaf = 1,\ random_seed=67, learning_rate=0.2), sampling_strategy='not majority', replacement=False, random_state=42, n_jobs = 4, verbose=1, oob_score=1,warm_start=0)	0.91/0.70
Balanced Bagging on CatBoost	model = BalancedBaggingClassifier(CatBoostClassifier(verbose=1, n_estimators=100, depth=10, min_data_in_leaf = 4,\ random_seed=67, learning_rate=0.5), sampling_strategy='not majority', replacement=False, random_state=42, n_jobs = 4, verbose=1, oob_score=1,warm_start=0)	0.84/0.70
Balanced Bagging on	BalancedBaggingClassifier(CatBoostClassifier(verbose=1, n_estimators=70, depth=7,	0.75/0.70

CatBoost	min_data_in_leaf = 5,\
	random_seed=67, learning_rate=0.6),
	sampling_strategy='not majority',
	replacement=False,
	random_state=42, n_jobs = 4, verbose=1,
	oob_score=1,warm_start=0)	

LGBMClassifier	Default	0.73/0.70
LGBMClassifier	lgb.LGBMClassifier(learning_rate=0.3,num_leaves=7,max_depth=10,min_child_samples=100,max_bin=100,subsample=0.7,subsample_freq=1,\ colsample_bytree=0.9,min_child_weight=0,scale_pos_weight=85,verbose=1)	0.72/0.69

LGBMClassifier	model = lgb.LGBMClassifier(learning_rate=0.3,max_depth=15,min_child_samples=10,max_bin=100,subsample=0.7,subsample_freq=1,\
	colsample_bytree=0.9,min_child_weight=0,scale_pos_weight=80,verbose=1)
		0.73/0.68

LGBMClassifier	lgb.LGBMClassifier(learning_rate=0.3,max_depth=12,min_child_samples=30,max_bin=100,subsample=0.7,subsample_freq=1,\ colsample_bytree=0.9,min_child_weight=0,is_unbalance='true',verbose=1)	0.75/0.70

LGBMClassifier	model = lgb.LGBMClassifier(learning_rate=0.1,max_depth=15,min_child_samples=5,max_bin=100,subsample=0.7,subsample_freq=1,reg_alpha=100,reg_lambda=100,\
	colsample_bytree=0.9,min_child_weight=0,is_unbalance='true',verbose=1,n_estimators=500)
		0.77/0.70

LGBMClassifier	lgb.LGBMClassifier(learning_rate=0.1,max_depth=10,min_child_samples=100,max_bin=100,subsample=0.7,subsample_freq=1,reg_alpha=500,reg_lambda=500,\	0.73/0.70

	colsample_bytree=0.9,min_child_weight=0,is_unbalance='true',verbose=1,n_estimators=500)	
LGBMClassifier	lgb.LGBMClassifier(learning_rate=0.1,max_depth=10,min_child_samples=100,max_bin=100,subsample=0.7,subsample_freq=1,reg_alpha=100,reg_lambda=100,\ colsample_bytree=0.9,min_child_weight=0,is_unbalance='true',verbose=1,n_estimators=500)	0.75/0.70

EasyEnsemble on XGBoost	model = EasyEnsembleClassifier(base_estimator=XGBClassifier(sampling_method='gradient_based',eta = 0.2, max_depth = 10,\
	verbosity=2, gamma=10, tree_method = 'gpu_hist', verbose=1),n_jobs = 4, random_state=42, sampling_strategy='all',verbose=1)
		0.73/0.70

EasyEnsemble on XGBoost	model = EasyEnsembleClassifier(base_estimator=XGBClassifier(sampling_method='gradient_based',eta = 0.9, max_depth = 7,\
	verbosity=2, gamma=1, tree_method = 'gpu_hist', verbose=1),n_jobs = 4, random_state=42, sampling_strategy='auto',verbose=1)
		0.87/0.70

EasyEnsemble on XGBoost	model = EasyEnsembleClassifier(base_estimator=XGBClassifier(sampling_method='gradient_based',eta = 0.9, max_depth = 7,\
	verbosity=2, gamma=100, tree_method = 'gpu_hist', verbose=1),n_jobs = 4, random_state=42, sampling_strategy='auto',verbose=1)
		0.70/0.68

EasyEnsemble on XGBoost	model = EasyEnsembleClassifier(base_estimator=XGBClassifier(sampling_method='gradient_based',eta = 1, max_depth = 20,\
	verbosity=2, gamma=30, tree_method = 'gpu_hist', verbose=1),n_jobs = 4, random_state=42, sampling_strategy='auto',verbose=1)
		0.71/0.69

EasyEnsemble on AdaBoost	Default	0.69/0.67

EasyEnsemble on AdaBoost	EasyEnsembleClassifier(base_estimator=AdaBoostClassifier(n_estimators=500, random_state=42,algorithm='SAMME.R',learning_rate=0.1),n_jobs = 4, random_state=42, sampling_strategy='all',verbose=1)	0.70/0.68
SGDClassifier	SGDClassifier(max_iter=2000, tol=1e-3, loss='log',learning_rate='adaptive',class_weight={0:1,1:5},eta0=0.5)	0.70/0.68
SGDClassifier	model = SGDClassifier(max_iter=2000, tol=1e-3, loss='log',learning_rate='adaptive',class_weight='balanced',eta0=0.5,warm_start=True)	0.69/0.67
EasyEnsemble	model = EasyEnsembleClassifier(base_estimator=SGDClassifier(max_iter=2000, tol=1e-3, loss='modified_huber',learning_rate='adaptive',class_weight='balanced',eta0=0.5,warm_start=True)\ ,n_jobs = 4, random_state=42, sampling_strategy='auto',verbose=1)	0.70/0.68
LGBM	lgbm.LGBMClassifier(n_estimators=500, learning_rate=0.01, colsample_bytree= 0.7296143102441466, num_leaves = 8)	0.69/0.69/0.68

Voting

Model	Params	Score

easyEnsemble on SGD	base = SGDClassifier(max_iter=1000, loss = 'log', verbose=1, n_jobs=4, class_weight="balanced", random_state=67)
	model = EasyEnsembleClassifier(base_estimator=base,\ n_jobs = 4, random_state=42, sampling_strategy='all',verbose=1, n_estimators=300)
		0.69/0.68

LGBM	lgbm.LGBMClassifier(n_estimators=500, learning_rate=0.01, colsample_bytree= 0.7296143102441466, num_leaves = 8)	0.69/0.69/0.68
Balanced random forest	BalancedRandomForestClassifier(n_esti	0.68/0.68

	mators=200, random_state=42, criterion = 'gini',\
	max_depth = 5, class_weight='balanced')	

XGBoost(Submission)	XGBClassifier(learning_rate=0.3, sampling_method='gradient_based',\
	eta = 0.5, max_depth = 20, subsample = 0.5,\
	colsample_bytree = 0.7296143102441466, verbosity=2,\
	objective='binary:logistic', gamma=20,\
	tree_method = 'gpu_hist')
		0.71/0.70

* The test scores were initially low as we were predicting the binary value 0/1 instead of probabilities.

In cases where there are only 2 scores mentioned, it refers to Train score and Test score, as validation had been removed for them.