

Model	Tuned Hyperparameters	Optimal Values
Logistic Regression	<pre>lg = LogisticRegressionCV(n_jobs=-1, random_state= 1234) lg_param_grid = {     'Cs': [6,8,10,15,20],     'max_iter': [60,80,100] } lg_cv= GridSearchCV(lg,lg_param_grid,cv=5,scoring="accuracy", n_jobs=-1, verbose=3) lg_cv.fit(x_train_normalized,y_train)  print("Best Score:" + str(lg_cv.best_score_)) print("Best Parameters: " + str(lg_cv.best_params_))</pre>	<p>Fitting 5 folds for each of 15 candidates, totalling 75 fits</p> <p>Best Score:0.6356404077730116</p> <p>Best Parameters: {'Cs': 6, 'max_iter': 60}</p>
SVM	<pre>svc = svm.SVC(random_state=1234,kernel='rbf',C= 10, gamma= 5 , tol = 1e-2,verbose = 1) svc.fit(x_train_normalized, y_train) print('train score',svc.score(x_train_normalized,y_train)) print('test score',svc.score(x_test_normalized,y_test))</pre>	<p>[LibSVM]train score 0.6650755767700876</p> <p>test score 0.6668181818181819</p>
XG Boost	<pre>params = {     'min_child_weight': [10,20],     'gamma': [1.5, 2.0, 2.5],     'colsample_bytree': [0.6, 0.8, 0.9],     'max_depth': [4,5,6] }  xgb = XGBClassifier(learning_rate=0.5, n_estimators=100, objective='binary:logistic', nthread=8) fitmodel = GridSearchCV(xgb, param_grid=params, cv=5, refit=True, scoring="accuracy", n_jobs=-1, verbose=3) fitmodel.fit(x_train_normalized, y_train) print(fitmodel.best_estimator_ , fitmodel.best_params_ , fitmodel.best_score_)</pre>	<p>Fitting 5 folds for each of 36 candidates, totalling 180 fits</p> <p>XGBClassifier(base_learning_rate=0.5, booster=gbtree, colsample_bynode=0.8, colsample_bytree=0.8, device=-1, early_stopping_rounds=100, eval_metric=auc, eval_monitor=auc, feature_selector=best, gamma=2.5, gpu_id=-1, importance_type=weight, interaction_constraints=None, learning_rate=0.5, max_bin=256, max_cat_threshold=10, max_cat_to_onehot=4, max_delta_step=0.001, max_depth=6, max_features=None, max_leaf_nodes=None, max_min_child_weight=10, max_monotone_constraints=None, max_n_estimators=100, max_parallelism='auto', max_regex_length=0, min_child_weight=10, missing=None, monotone_constraints=None, multi_threading=True, n_estimators=100, n_jobs=-1, nthread=8, num_parallel_tree=3, ...)</p>

Random Forest	<pre>rf = RandomForestClassifier() rf_param_grid = {     'n_estimators': [200,300,500],     'criterion': ['entropy','gini'],     'max_depth': [7,8,60,80,100],     'max_features': ['auto', 'sqrt', 'log2'] } rf_cv = GridSearchCV(rf,rf_param_grid,cv=7,scoring="accuracy", n_jobs=-1, verbose=3) rf_cv.fit(x_train,y_train)  print("Best Score:" + str(rf_cv.best_score_)) print("Best Parameters: " + str(rf_cv.best_params_))</pre>	<pre>t  warn( train 0.7104216388225935 test 0.6818181818181818</pre>
---------------	---	--

### Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric																																																																																																
MODELS	<table><thead><tr><th></th><th>Name</th><th>Accuracy</th><th>f1_score</th><th>Recall</th><th>Precision</th></tr></thead><tbody><tr><td>0</td><td>logistic regression</td><td>59.27</td><td>74.43</td><td>100.00</td><td>59.27</td></tr><tr><td>1</td><td>logistic regression CV</td><td>63.27</td><td>66.78</td><td>62.27</td><td>71.99</td></tr><tr><td>2</td><td>XGBoost</td><td>65.59</td><td>69.63</td><td>66.56</td><td>73.00</td></tr><tr><td>3</td><td>Ridge classifier</td><td>59.27</td><td>74.43</td><td>100.00</td><td>59.27</td></tr><tr><td>4</td><td>KNN</td><td>63.27</td><td>68.39</td><td>67.02</td><td>69.81</td></tr><tr><td>5</td><td>Random Forest</td><td>67.18</td><td>70.36</td><td>65.72</td><td>75.71</td></tr><tr><td>6</td><td>Support Vector Classifier</td><td>59.27</td><td>74.43</td><td>100.00</td><td>59.27</td></tr></tbody></table>		Name	Accuracy	f1_score	Recall	Precision	0	logistic regression	59.27	74.43	100.00	59.27	1	logistic regression CV	63.27	66.78	62.27	71.99	2	XGBoost	65.59	69.63	66.56	73.00	3	Ridge classifier	59.27	74.43	100.00	59.27	4	KNN	63.27	68.39	67.02	69.81	5	Random Forest	67.18	70.36	65.72	75.71	6	Support Vector Classifier	59.27	74.43	100.00	59.27	<table><thead><tr><th></th><th>Name</th><th>Accuracy</th><th>f1_score</th><th>Recall</th><th>Precision</th></tr></thead><tbody><tr><td>0</td><td>logistic regression</td><td>59.27</td><td>74.43</td><td>100.00</td><td>59.27</td></tr><tr><td>1</td><td>logistic regression CV</td><td>63.27</td><td>66.78</td><td>62.27</td><td>71.99</td></tr><tr><td>2</td><td>XGBoost</td><td>65.27</td><td>68.84</td><td>64.72</td><td>73.52</td></tr><tr><td>3</td><td>Ridge classifier</td><td>59.27</td><td>74.43</td><td>100.00</td><td>59.27</td></tr><tr><td>4</td><td>KNN</td><td>63.27</td><td>68.39</td><td>67.02</td><td>69.81</td></tr><tr><td>5</td><td>Random Forest</td><td>67.55</td><td>66.38</td><td>54.06</td><td>85.98</td></tr><tr><td>6</td><td>Support Vector Classifier</td><td>67.05</td><td>69.06</td><td>62.04</td><td>77.86</td></tr></tbody></table>		Name	Accuracy	f1_score	Recall	Precision	0	logistic regression	59.27	74.43	100.00	59.27	1	logistic regression CV	63.27	66.78	62.27	71.99	2	XGBoost	65.27	68.84	64.72	73.52	3	Ridge classifier	59.27	74.43	100.00	59.27	4	KNN	63.27	68.39	67.02	69.81	5	Random Forest	67.55	66.38	54.06	85.98	6	Support Vector Classifier	67.05	69.06	62.04	77.86
		Name	Accuracy	f1_score	Recall	Precision																																																																																												
	0	logistic regression	59.27	74.43	100.00	59.27																																																																																												
	1	logistic regression CV	63.27	66.78	62.27	71.99																																																																																												
	2	XGBoost	65.59	69.63	66.56	73.00																																																																																												
	3	Ridge classifier	59.27	74.43	100.00	59.27																																																																																												
	4	KNN	63.27	68.39	67.02	69.81																																																																																												
	5	Random Forest	67.18	70.36	65.72	75.71																																																																																												
6	Support Vector Classifier	59.27	74.43	100.00	59.27																																																																																													
	Name	Accuracy	f1_score	Recall	Precision																																																																																													
0	logistic regression	59.27	74.43	100.00	59.27																																																																																													
1	logistic regression CV	63.27	66.78	62.27	71.99																																																																																													
2	XGBoost	65.27	68.84	64.72	73.52																																																																																													
3	Ridge classifier	59.27	74.43	100.00	59.27																																																																																													
4	KNN	63.27	68.39	67.02	69.81																																																																																													
5	Random Forest	67.55	66.38	54.06	85.98																																																																																													
6	Support Vector Classifier	67.05	69.06	62.04	77.86																																																																																													

### Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest	The Random Forest model was selected for its superior performance, exhibiting high accuracy during hyperparameter tuning. Its ability to handle complex relationships, minimize overfitting, and optimize predictive accuracy aligns with project objectives, justifying its selection as the final model