Model Optimization and Tuning Phase Template

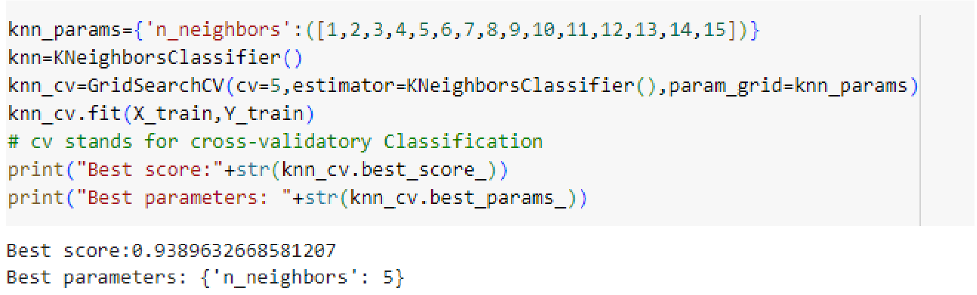
Date	14th July 2024
Team ID	SWTID1720243396
Project Title	Panic Disorder Detection
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

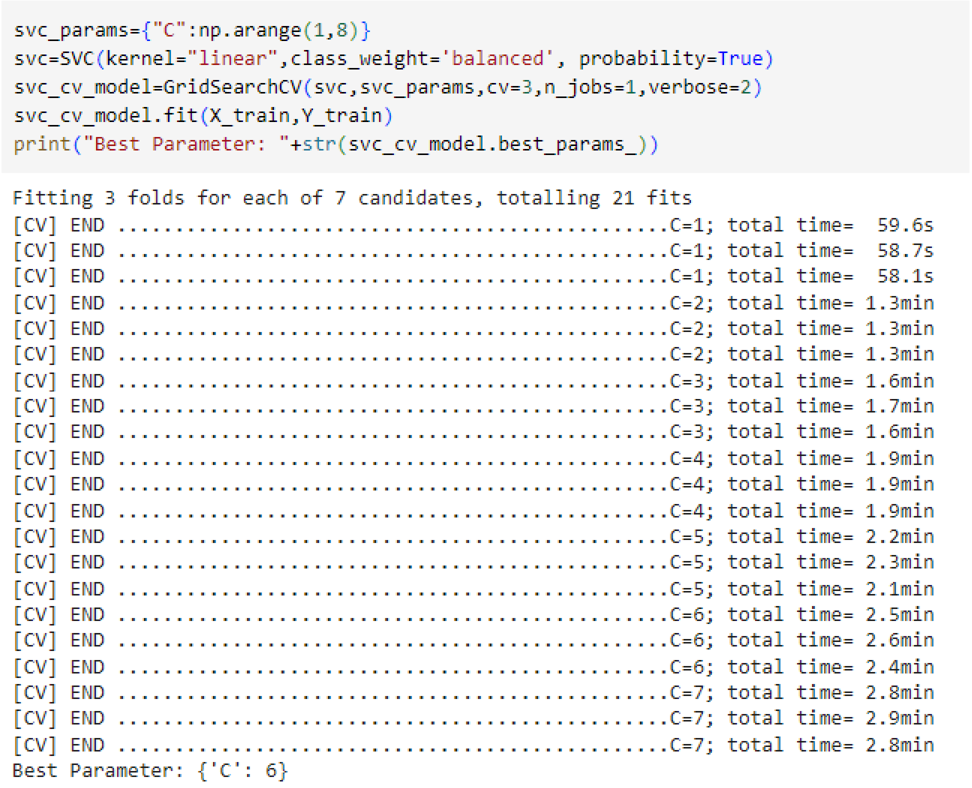
The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters

KNN(with hyper-parameterisation Tuning)	
	`knn_params={'n_neighbors':([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])}`
	`knn=KNeighborsClassifier()`
	`knn_cv=GridSearchCV(cv=5,estimator=KNeighborsClassifier(),param_grid=knn_params)`
	`knn_cv.fit(X_train,Y_train)`
	`# cv stands for cross-validatory Classification`
	`print("Best score:"+str(knn_cv.best_score_))`
	`print("Best parameters: "+str(knn_cv.best_params_))`
	`Best score:0.9389632668581207`
	`Best parameters: {'n_neighbors': 5}`

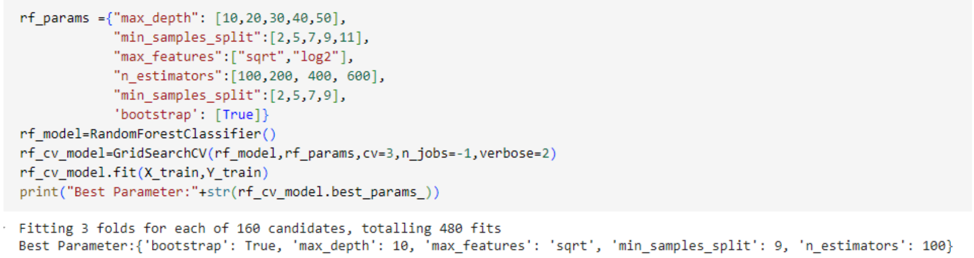
<table>
<tr>
<td>SVM(Linear) (with Hyper-parameterization Tuning)</td>
<td>



```
svc_params={"C":np.arange(1,8)}
svc=SVC(kernel="linear",class_weight='balanced', probability=True)
svc_cv_model=GridSearchCV(svc,svc_params,cv=3,n_jobs=1,verbose=2)
svc_cv_model.fit(X_train,Y_train)
print("Best Parameter: "+str(svc_cv_model.best_params_))
```

```
Fitting 3 folds for each of 7 candidates, totalling 21 fits
[CV] END ...........................................C=1; total time=  59.6s
[CV] END ...........................................C=1; total time=  58.7s
[CV] END ...........................................C=1; total time=  58.1s
[CV] END ...........................................C=2; total time= 1.3min
[CV] END ...........................................C=2; total time= 1.3min
[CV] END ...........................................C=2; total time= 1.3min
[CV] END ...........................................C=3; total time= 1.6min
[CV] END ...........................................C=3; total time= 1.7min
[CV] END ...........................................C=3; total time= 1.6min
[CV] END ...........................................C=4; total time= 1.9min
[CV] END ...........................................C=4; total time= 1.9min
[CV] END ...........................................C=4; total time= 1.9min
[CV] END ...........................................C=5; total time= 2.2min
[CV] END ...........................................C=5; total time= 2.3min
[CV] END ...........................................C=5; total time= 2.1min
[CV] END ...........................................C=6; total time= 2.5min
[CV] END ...........................................C=6; total time= 2.6min
[CV] END ...........................................C=6; total time= 2.4min
[CV] END ...........................................C=7; total time= 2.8min
[CV] END ...........................................C=7; total time= 2.9min
[CV] END ...........................................C=7; total time= 2.8min
Best Parameter: {'C': 6}
```

</td>
</tr>
<tr>
<td>Random Forest (With Hyper-parameterization Tuning)</td>
<td>



```
rf_params ={"max_depth": [10,20,30,40,50],
            "min_samples_split":[2,5,7,9,11],
            "max_features":["sqrt","log2"],
            "n_estimators":[100,200, 400, 600],
            "min_samples_split":[2,5,7,9],
            'bootstrap': [True]}
rf_model=RandomForestClassifier()
rf_cv_model=GridSearchCV(rf_model,rf_params,cv=3,n_jobs=-1,verbose=2)
rf_cv_model.fit(X_train,Y_train)
print("Best Parameter:"+str(rf_cv_model.best_params_))
```

```
Fitting 3 folds for each of 160 candidates, totalling 480 fits
Best Parameter:{'bootstrap': True, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_split': 9, 'n_estimators': 100}
```

</td>
</tr>
</table>

Final Model Selection Justification (2 Marks):

Final Model	Reasoning

KNN(with hyper-parameterisation Tuning)	