

SPRINT 2 – MODEL BUILDING

Model building

```
from sklearn.preprocessing import StandardScaler  
  
sc = StandardScaler()  
  
X_train_final = sc.fit_transform(X_train)  
  
X_test_final = sc.transform(X_test)  
  
from sklearn.metrics import confusion_matrix, classification_report,  
accuracy_score
```

Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier  
  
rf_classifier = RandomForestClassifier(n_estimators = 20, criterion =  
'entropy', class_weight = "balanced_subsample", random_state = 51)  
  
rf_classifier.fit(X_train_final, y_train)  
  
y_pred = rf_classifier.predict(X_test_final)  
  
accuracy_score(y_test, y_pred)  
  
print(classification_report(y_test, y_pred))
```

XGBoost Classifier

```
from xgboost import XGBClassifier  
  
xgb_classifier = XGBClassifier(random_state=0)  
  
xgb_classifier.fit(X_train_final, y_train)  
  
y_pred_xgb = xgb_classifier.predict(X_test_final)  
  
accuracy_score(y_test, y_pred_xgb)
```

```
print(classification_report(y_test, y_pred_xgb))
```

Support vector Machine

Support vector classifier

```
from sklearn.svm import SVC
```

```
svc_classifier = SVC(class_weight = "balanced" )
```

```
svc_classifier.fit(X_train_final, y_train)
```

```
y_pred_scv = svc_classifier.predict(X_test_final)
```

```
accuracy_score(y_test, y_pred_scv)
```

```
print(classification_report(y_test, y_pred_scv))
```

```
cm = confusion_matrix(y_test, y_pred_scv)
```

```
plt.title('Heatmap of Confusion Matrix', fontsize = 12)
```

```
sns.heatmap(cm, annot = True, fmt = "d")
```

```
plt.show()
```

defining parameter range

```
param_grid = {'C': [0.1, 1, 10, 100, 200 , 400 , 600 , 800],
```

```
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
```

```
              'kernel': ['rbf']}]
```

```
from sklearn.model_selection import GridSearchCV
```

```
grid = GridSearchCV(SVC(), param_grid, refit = True, verbose = 3)
```

fitting the model for grid search

```
grid.fit(X_train_final, y_train)
```

print best parameter after tuning

```
print(grid.best_params_)
```

print how our model looks after hyper-parameter tuning

```
print(grid.best_estimator_)
```

Support vector classifier

```
from sklearn.svm import SVC
```

```
svc_classifier = SVC(class_weight = "balanced" , C=100, gamma=0.01)
```

```
svc_classifier.fit(X_train_final, y_train)
```

```
y_pred_scv = svc_classifier.predict(X_test_final)
```

```
accuracy_score(y_test, y_pred_scv)
```

```
print(classification_report(y_test, y_pred_xgb))
```

```
cm = confusion_matrix(y_test, y_pred_scv)
```

```
plt.title('Heatmap of Confusion Matrix', fontsize = 12)
```

```
sns.heatmap(cm, annot = True , fmt = "d")
```

```
plt.show()
```

Pickle

```
from sklearn.svm import SVC
```

```
import pickle
```

save model

```
pickle.dump(svc_classifier, open('model.pkl', 'wb'))
```

```
# load model
```

```
water_quality_model = pickle.load(open('model.pkl', 'rb'))
```

```
# predict the output
```

```
y_pred = water_quality_model.predict(X_test_final)
```

```
# confusion matrix
```

```
print('Confusion matrix of Support vector Machine :  
\n', confusion_matrix(y_test, y_pred), '\n')
```