XG Boost

XG Boost Code

```
In [1]: import xgboost as xgb
In [2]: import pandas as pd
In [3]: import numpy as np
In [4]: from sklearn.model_selection import train_test_split
In [5]: class_data = pd.read_csv("classification_data.csv")
In [6]: X, y = class_data.iloc[:,:-1], class_data.iloc[:,-1]
In [7]: X_train, X_test, y_train, y_test= train_test_split(X, y, test_size=0.2, random_state=123)
In [8]: xg_cl = xgb.XGBClassifier(objective='binary:logistic', n_estimators=10, seed=123)
In [9]: xg_cl.fit(X_train, y_train)
In [10]: preds = xg_cl.predict(X_test)
In [11]: accuracy = float(np.sum(preds==y_test))/y_test.shape[0]
In [12]: print("accuracy: %f" % (accuracy))
accuracy: 0.78333
```

Decision Tree

```
In [1]: from sklearn.model_selection import train_test_split
In [2]: from sklearn.tree import DecisionTreeClassifier
In [3]: X_train, X_test, y_train, y_test= train_test_split(X, y, test_size=0.2, random_state=123)
In [4]: dt_clf_4 = DecisionTreeClassifier(max_depth = 4)
In [5]: dt_clf_4.fit(X_train,y_train)
In [6]: y_pred_4 = dt_clf_4.predict(X_test)
In [7]: accuracy = float(np.sum(y_pred_4==y_test))/y_test.shape[0]
In [8]: print("accuracy:", accuracy)
```

Cross-validation in XGBoost

```
In [1]: import xgboost as xgb
In [2]: import pandas as pd
In [3]: class_data = pd.read_csv("classification_data.csv")
In [4]: churn_dmatrix = xgb.DMatrix(data=X, label=y)
In [5]: params={"objective":"binary:logistic","max_depth":4}
In [6]: cv_results = xgb.cv(dtrain=churn_dmatrix, params=params, nfold=4,
num_boost_round=10, metrics="error", as_pandas=True , seed=123)
In [7]: print("Accuracy: %f" %((1-cv_results["test-error-mean"]).iloc[-1]))
Accuracy: 0.88315
```

AUC

```
In [1]: cv_results = xgb.cv(dtrain=churn_dmatrix, params=params, nfold=3, num_boost_round=5,
metrics="auc", as_pandas=True , seed=123)
In [2]: print(cv_results)
In [3]: print((cv_results["test-auc-mean"]).iloc[-1])
```