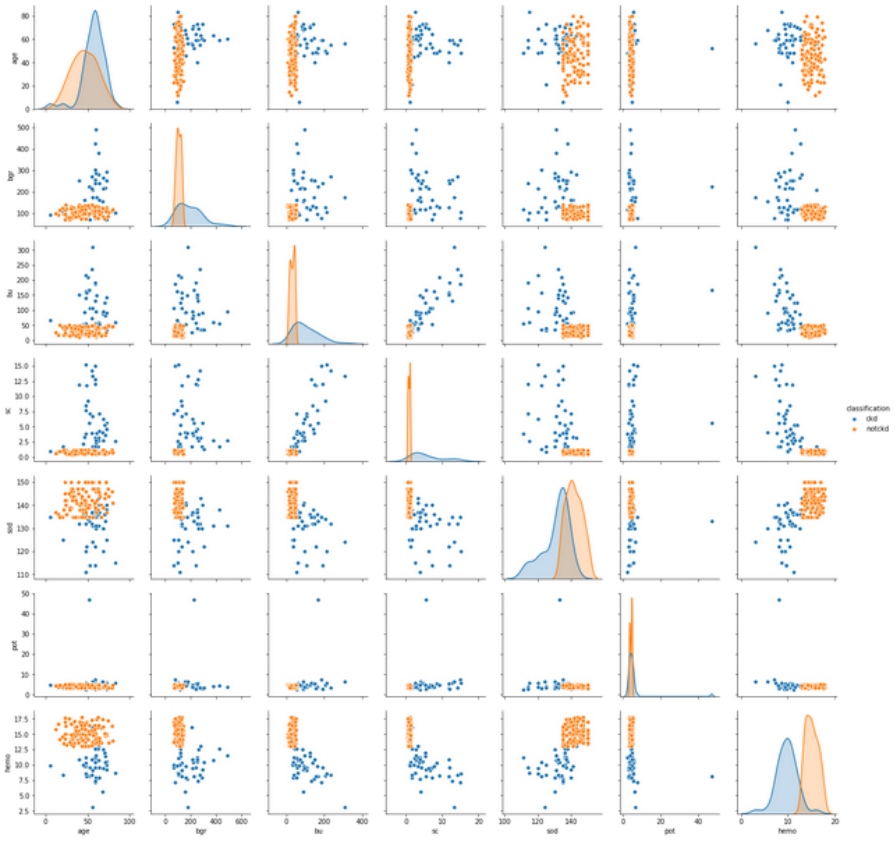
Ensemble Learning

Ishita Jaju
16BCE1059

Dataset:



1 Accuracy with SVM/MLP

MLP and SVM gives 73% accuracy

```
#SVM
from sklearn import preprocessing, cross_validation, neighbors, svm
clf = svm.SVC()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

```
0.7291666666666666
```

```
#MLP
from sklearn.neural_network import MLPClassifier
clf = MLPClassifier(solver='lbfgs', alpha=1e-5,hidden_layer_sizes=(5, 2), random_state=1)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

```
accuracy_score(y_test,y_pred)
```

```
0.7291666666666666
```

2 Bagging

Using the Random Forest Algorithm It gives 46% accuracy

```
#Bagging - Random Forest
y_train_classes = np.where(y_train=="ckd",1,-1)
y_test_classes = np.where(y_test=="ckd",1,-1)

from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(n_estimators = 1000, random_state = 0)
rf.fit(X_train, y_train_classes)
y_pred = rf.predict(X_test)
```

```
accuracy_score(y_test_classes,y_pred.round(), normalize=False)
```

```
46
```

3 Boosting

3.1 Adaboost Algorithm

Accuracy 100%

```
#Boosting - Adaboost
from sklearn.ensemble import AdaBoostClassifier
clf = AdaBoostClassifier(n_estimators=50,
                         learning_rate=1,
                         random_state=0)
clf.fit(X_train, y_train)
y_pred=clf.predict(X_test)
```

```
accuracy_score(y_test,y_pred)
```

```
1.0
```

3.2 XGBoost Algorithm

```
#Boosting - XGBoost
import xgboost as xgb
gbm = xgb.XGBClassifier(max_depth=3, n_estimators=300, learning_rate=0.05).fit(X_train,y_train)
y_pred = gbm.predict(X_test)
```

```
/usr/local/lib/python3.5/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The tru
pty array is ambiguous. Returning False, but in future this will result in an error. Use `array.size >
an array is not empty.
  if diff:
```

```
accuracy_score(y_test,y_pred)
```

```
1.0
```

Accuracy 100%

4 Why bagging and boosting improve the performance

Ensemble methods like bagging and boosting are algorithms that combine several machine learning techniques into one predictive model. This is done in order to

- decrease variance (bagging),
- bias (boosting)

Hence, they end up performing better than algorithms like SVM and MLP

5 Why XGBoost helps in winning competitions

Tree boosting is much more efficient for predictive mining for both classification and regression. Trees do have limited predictive power, but when multiple tree models are combined together they become a powerful predictive model. XGBoost stands for eXtreme Gradient Boosting. Three main forms of gradient boosting are supported in this model:

- Gradient Boosting algorithm also called gradient boosting machine including the learning rate.
- Stochastic Gradient Boosting with sub-sampling at the row, column and column per split levels.
- Regularized Gradient Boosting with both L1 and L2 regularization.

So XGBoost is used because of two of its properties: execution speed and model performance.