Results:

```
SVM Results
Best SVM Parameters
{'decision_function_shape': 'ovo', 'gamma': 'scale', 'kernel': 'linear'}
SVM Classification Report
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        54
           1       0.97      0.99      0.98       117

    accuracy                           0.97       171
   macro avg       0.97      0.96      0.97       171
weighted avg       0.97      0.97      0.97       171

SVM Confusion Matrix
[[ 50   4]
 [  1 116]]
Random Forest Results
Best Random Forest Parameters
{'class_weight': 'balanced', 'criterion': 'entropy', 'max_features': 'log2'}
Random Forest Classification Report
              precision    recall  f1-score   support

           0       0.96      0.93      0.94        54
           1       0.97      0.98      0.97       117

    accuracy                           0.96       171
   macro avg       0.96      0.95      0.96       171
weighted avg       0.96      0.96      0.96       171

Random Forest Confusion Matrix
[[ 50   4]
 [  2 115]]
```

Explanation

The program uses two classifiers SVC and Random Forest; the latter is used as the ensemble algorithm while the first is used for the baseline classifier. As given by the instructions the dataset is split into a testing and training set with the given 30-70% split. A grid search algorithm is used to identify optimal parameters. The optimal parameters as shown in output was an One vs One, scaling gamma, linear kernel for SVC and log_2 features entropy based balanced weight for random forest. We have found very little difference between the two algorithms with SVC performing slightly better than random forest.

Code:

```
#Breast Cancer Dataset
from sklearn.datasets import load_breast_cancer

#Baseline Classifier
from sklearn.svm import SVC

#Ensemble Classifier
from sklearn.ensemble import RandomForestClassifier

#Grid Search for Parameter tuning
from sklearn.model_selection import train_test_split, GridSearchCV

#Classification report and confusion matrix for analysis
from sklearn.metrics import classification_report, confusion_matrix

'''
   Extracts the Data to use on model
'''

#Grabbing Data
dataset = load_breast_cancer()
target = dataset.target
data = dataset.data

#Splitting Data
X_train, X_test, y_train, y_test = train_test_split(data, target,
test_size=0.30)


'''
   Training and usage of baseline classification model
'''

print("SVM Results")

#Creates and trains best baseline model with best parameters
parameters = {'kernel':('linear', 'rbf','sigmoid'), 'gamma':('scale',
'auto'), 'decision_function_shape' : ('ovo', 'ovr')}
```

```
model = SVC()
model = GridSearchCV(model, parameters)

model.fit(X_train, y_train)
print("Best SVM Parameters")
print(model.best_params_)

# Tests model on data

y_pred = model.predict(X_test)

print("SVM Classification Report")
print(classification_report(y_test, y_pred))

print("SVM Confusion Matrix")

cm = confusion_matrix(y_test, y_pred)
print(cm)


'''
   Training and usage of ensemble classification model
'''

print("Random Forest Results")

#Creates and trains best baseline model with best parameters
parameters = {'criterion' : ('gini', 'entropy', 'log_loss'),
'max_features' : ('sqrt', 'log2'), 'class_weight' : ('balanced',
'balanced_subsample')}

model = RandomForestClassifier()
model = GridSearchCV(model, parameters)

model.fit(X_train, y_train)
print("Best Random Forest Parameters")
print(model.best_params_)

# Tests model on data
```

```
y_pred = model.predict(X_test)

print("Random Forest Classification Report")
print(classification_report(y_test, y_pred))

print("Random Forest Confusion Matrix")

cm = confusion_matrix(y_test, y_pred)
print(cm)
```