Part B

```
#Nikolay Valev
from sklearn.datasets import fetch_openml
from sklearn.metrics import accuracy_score
mnist = fetch_openml('mnist_784', version=1)
```

```
X, y = mnist["data"], mnist["target"]
X.shape
```

```
(70000, 784)
```

Split the dataset into a training set, a validation set, and a test set using the ratio 5 : 1 : 1.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/7, random_state=1)

X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=1/6, random_sta
```

```
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
print(X_val.shape)
print(y_val.shape)
```

```
(50000, 784)
(50000,)
(10000, 784)
(10000,)
(10000, 784)
(10000,)
```

a. Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_predict
forest_clf = RandomForestClassifier(n_estimators=500,random_state=42)
forest_clf.fit(X_train, y_train)
y_pred_rf = forest_clf.predict(X_test)
print(accuracy_score(y_test, y_pred_rf))
```

```
0.9689
```

b. Bagging Classifier

```
from sklearn import model_selection
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
import pandas as pd

# bagging classifier
bagging_clf = BaggingClassifier(n_estimators=500,
                           max_samples=100, bootstrap=True, random_state=42)
bagging_clf.fit(X_train, y_train)
Bagging_pred = bagging_clf.predict(X_test)
print(accuracy_score(y_test, Bagging_pred))
```

```
0.842
```

Decision Tree Classifier

```
dtree_clf = DecisionTreeClassifier(random_state=42)
dtree_clf.fit(X_train, y_train)
y_pred_tree = dtree_clf.predict(X_test)
print(accuracy_score(y_test, y_pred_tree))
```

```
0.8658
```

Combine the classifiers into an ensemble on the validation set using hard voting.

```
from sklearn.ensemble import VotingClassifier
voting_clf = VotingClassifier(
    estimators=[('bg', bagging_clf), ('rf', forest_clf), ('dt', dtree_clf)],
    voting='hard')  # hard voting
voting_clf.fit(X_train, y_train)
y_pred = voting_clf.predict(X_test)
print(accuracy_score(y_test, y_pred))
```

```
0.9331
```

4. Does the ensemble outperform the individual classifiers? The voting classifier which has accuracy of 0.9457 performs better than all the individual classifiers except random forest

classifier which has accuracy of 0.97

5. Next remove the individual classifier with the smallest accuracy score. Bagging Classifier has accuracy of 0.842 making it the worst performing one.

```
from sklearn.ensemble import VotingClassifier
voting_clf = VotingClassifier(
    estimators=[('dt', dtree_clf), ('rf', forest_clf)],
voting='hard')
voting_clf.fit(X_train, y_train)
```

```
VotingClassifier(estimators=[('dt', DecisionTreeClassifier(random_state=42)),
                             ('rf',
                              RandomForestClassifier(n_estimators=500,
                                                     random_state=42))])
```

6. Now combine the classifiers into an ensemble on the test data using hard voting.

```
y_pred = voting_clf.predict(X_test)
print(accuracy_score(y_test, y_pred))
```

```
0.9208
```

7. How much better does it perform compared to the individual classifiers? Comment on your results

When the bagging classifier is removed from the ensemble, the voting classifier outperforms the decision tree classifier (0.8658) but not the random forest classifier (0.9689).The voting classifier accuracy score decreased to 0.9208 from 0.9331