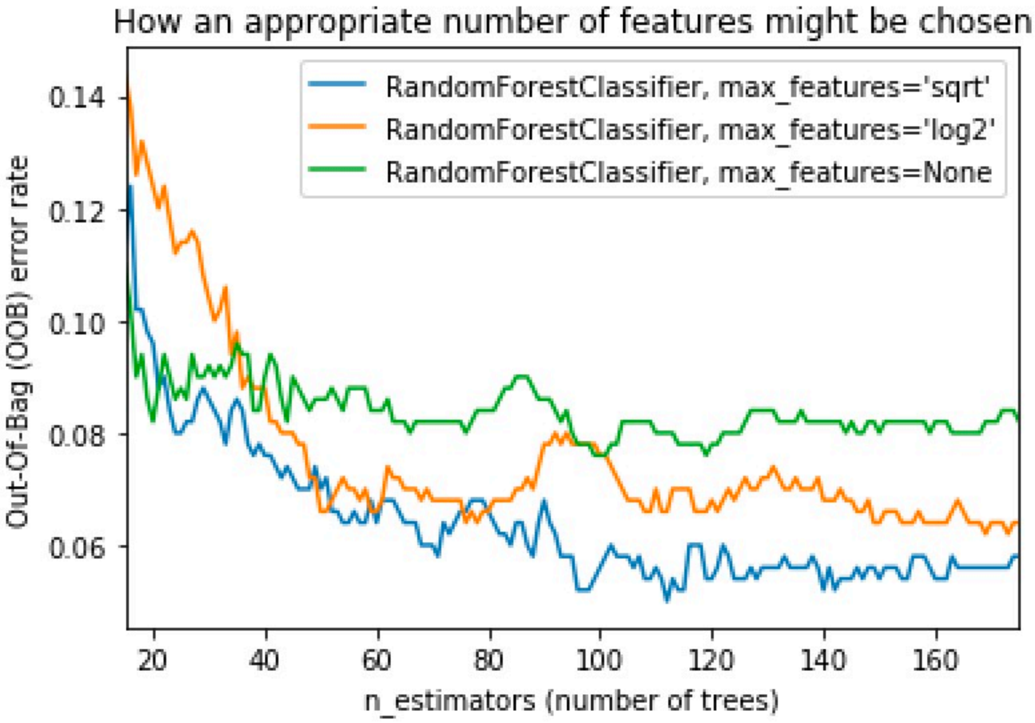
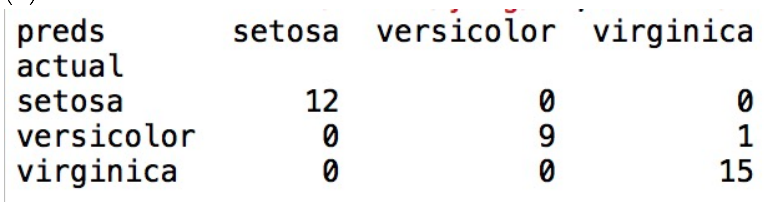
Random Forest

1.

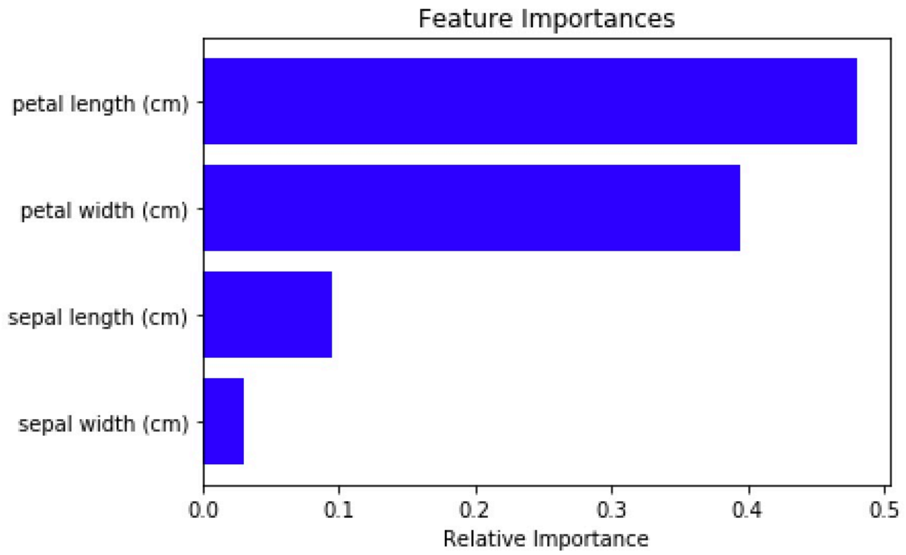


2.

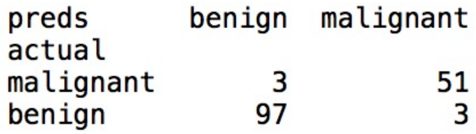
(a)



```
preds       setosa  versicolor  virginica
actual
setosa          12           0          0
versicolor       0           9          1
virginica        0           0         15
```



(b)

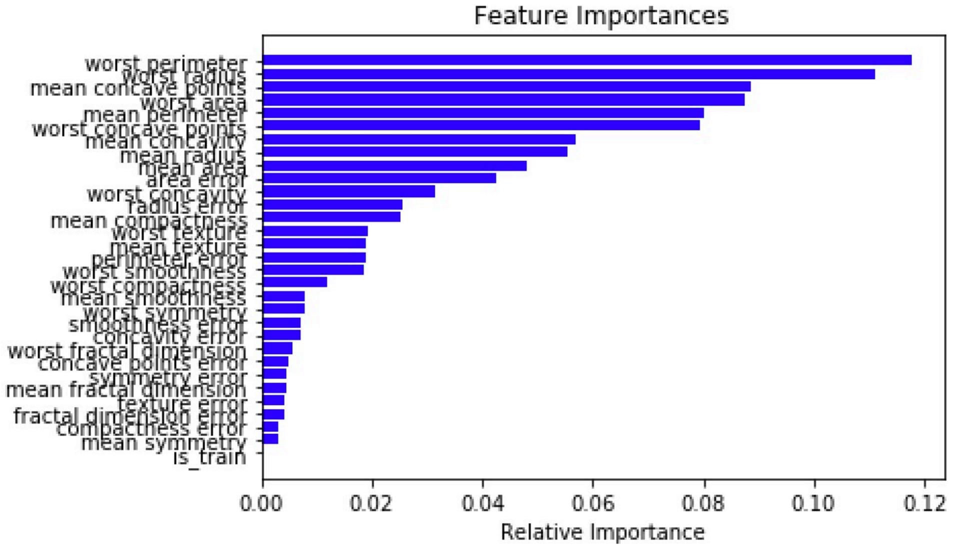


```
preds      benign  malignant
actual
malignant       3         51
benign         97          3
```

(i) breast cancer

(ii) 32

(iii)

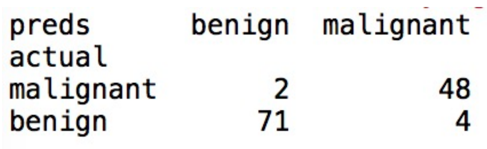


seeing the result, I think 6 features will be good. They are worst perimeter, worst radius, mean concave points , worst area, mean perimeter and worst concave points .

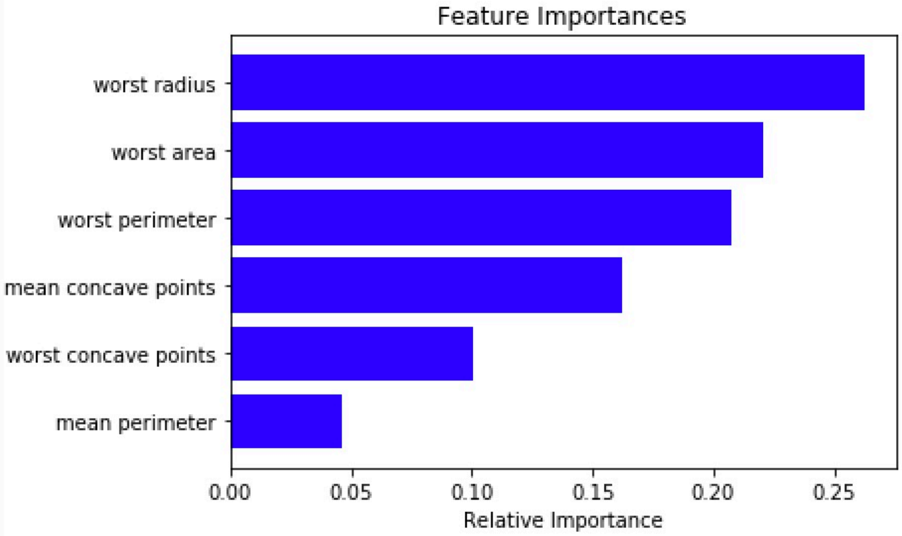
(iv)

```
from sklearn.ensemble import RandomForestClassifier as RFC
from sklearn.datasets import load_breast_cancer

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def getitems(a, *key):
    return type(a)(map(lambda i:a[i],key))

breast_cancer = load_breast_cancer()
df = pd.DataFrame(breast_cancer.data, columns=breast_cancer.feature_names)
df['is_train'] = np.random.uniform(0, 1, len(df)) <= .75
df['class'] = pd.Categorical.from_codes(breast_cancer.target,
breast_cancer.target_names)


train, test = df[df['is_train']==True], df[df['is_train']==False]
features = getitems(df.columns,2,7,20,22,23,27)

forest = RFC(n_jobs=2,n_estimators=50)
y, _ = pd.factorize(train['class'])
forest.fit(train[features], y)

preds = breast_cancer.target_names[forest.predict(test[features])]
print pd.crosstab(index=test['class'], columns=preds, rownames=['actual'],
colnames=['preds'])

importances = forest.feature_importances_
indices = np.argsort(importances)

plt.figure(1)
plt.title('Feature Importances')
plt.barh(range(len(indices)), importances[indices], color='b', align='center')
plt.yticks(range(len(indices)), features[indices])
plt.xlabel('Relative Importance')
```

(v)



```
preds      benign  malignant
actual
malignant       2         48
benign         71          4
```



Discussion: The model is better. The relative importance is much improved and less confusing factors, too. The graph looks more direct as well.