Mercury Mcindoe 85594505

1) Coding practice [10 pts]

In this assignment, we will be working on the dataset of model price classification. The task is to classify the level of cost given the features of mobiles. In machine learning terms, it is a classification problem. The data are provided in the data folder. The training and testing data are in the different files. You can use numpy or pandas libraries to load the csv files.
The targets in the data have values:

- 0 (low cost)
- 1 (medium cost)
- 2 (high cost)
- 3 (very high cost)

which can be read from column `price_range`.

The features are the things like:

- battery power: Total energy a battery can store in one time measured in mAh
- blue: Has bluetooth or not
- clock speed: speed at which microprocessor executes instructions
- dual sim: Has dual sim support or not
- fc: Front Camera mega pixels
- four g: Has 4G or not
- . . .

which can be read from the columns after dropping column `price_range`. Before feeding the features in the machine learning models, you need to do zero-mean unit variance normalization: https://en.wikipedia.org/wiki/Feature scaling.

(a) Train a linear kernel SVM using `sklearn.svm.LinearSVC`. Please report the accuracy on testing data when choosing C in the range of $\left(10^{-5}, 10^{-4}, 10^{-3}, ..., 10^{5}\right)$ using `matplotlib.pyplot` where x-axis is C and y-axis is accuracy (range from 0 to 1). The accuracy can be calculated using `sklearn.metrics.accuracy_score`.

Using the code,

```
Cs = [10**(-5), 10**(-4), 10**(-3), 10**(-2), 10**(-1), 1, 10, 10**2, 10**3, 10**4,
10**5]
accuracies = []
for c in Cs:
  lsvc = LinearSVC(random_state = 7, C = c, dual=False)
  lsvc.fit(X_train_norm, y_train)
  y_pred = lsvc.predict(X_test_norm)
  accuracy = metrics.accuracy_score(y_test, y_pred)
  accuracies.append(accuracy)
```

The accuracies are shown as the follwing,

C	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10	10^{1}	10^{2}	10^{3}	10^{4}
Accurracy	0.588	0.592	0.604	0.7	0.792	0.824	0.836	0.846	0.84,	0.842	0.842

We can get the plot,

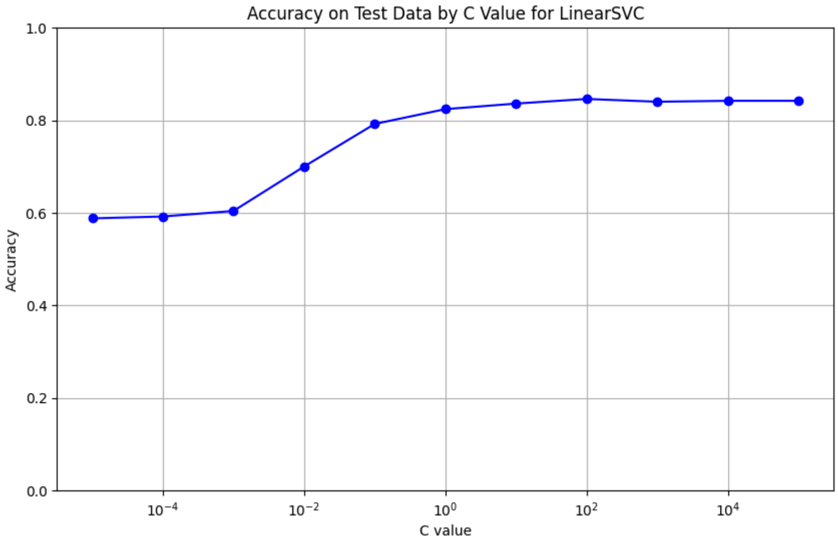

Figure 1: plot for tables

(b) Train a RBF (Gaussian) kernel SVM using `sklearn.svm.SVC`. Please report the accuracy on testing data when choosing γ in the range of $(10^{-1}, 10^0, ..., 10^4)$ using `matplotlib.pyplot` where x-axis is γ and y-axis is accuracy (range from 0 to 1). The accuracy can be calculated using `sklearn.metrics.accuracy_score`.

By following,

```
gammas = [10**(-1), 1, 10, 10**2, 10**3, 10**4]
accuracies = []
for g in gammas:
  rsvc = SVC(random_state = 7, C=1.0, kernel='rbf', gamma = g)
  rsvc.fit(X_train_norm, y_train)
  y_pred = rsvc.predict(X_test_norm)
  accuracy = metrics.accuracy_score(y_test, y_pred)
  accuracies.append(accuracy)
```

The accuracies are shown as the follwing,

γ	10^{-1}	1	10	10^2	10^3	10^4
Accurracy	0.806	0.246	0.242	0.242	0.242	0.242

We can plot the accuracy as,

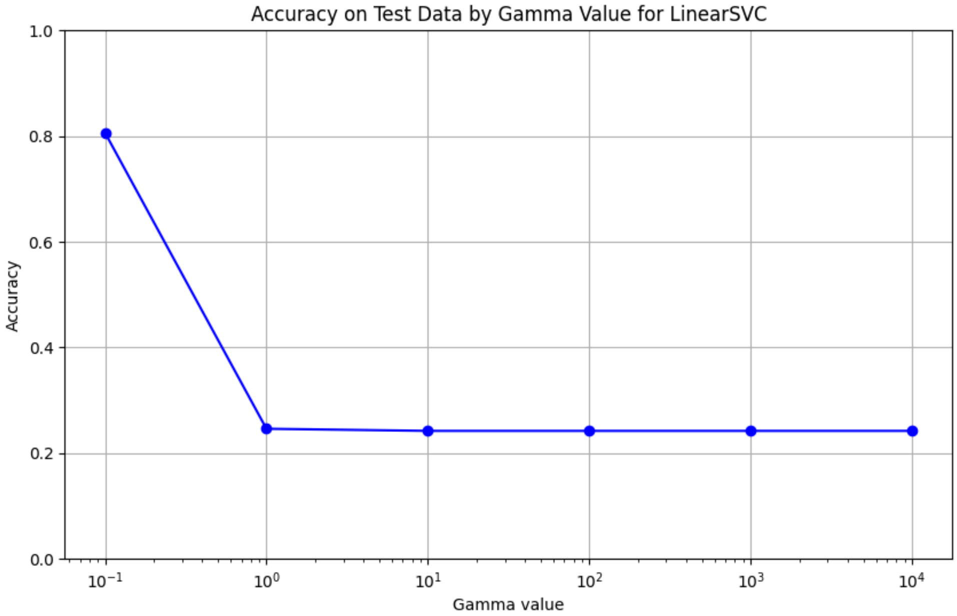

Figure 2: plot for tables

(c) Train a Random Forest Classifier using `RandomForestClassifier`. Please report the accuracy on testing data when choosing number of trees (`n_estimator`) in the range of (10, 100, 500, 1000) using `matplotlib.pyplot` where x-axis is n estimator and y-axis is accuracy (range from 0 to 1). The accuracy can be calculated using `sklearn.metrics.accuracy_score`.
Similar to the questions above, by following,

```
n_trees = [10, 100, 500, 1000]
accuracies = []
for n in n_trees:
  rf = RandomForestClassifier(random_state = 7, n_estimators=n)
  rf.fit(X_train_norm, y_train)
  y_pred = rf.predict(X_test_norm)
  accuracy = metrics.accuracy_score(y_test, y_pred)
  accuracies.append(accuracy)
```

The accuracies are shown as the follwing,

n_estimator	10	100	500	1000
Accurracy	0.776	0.858	0.866	0.87

The accuracy is plotted as the following,

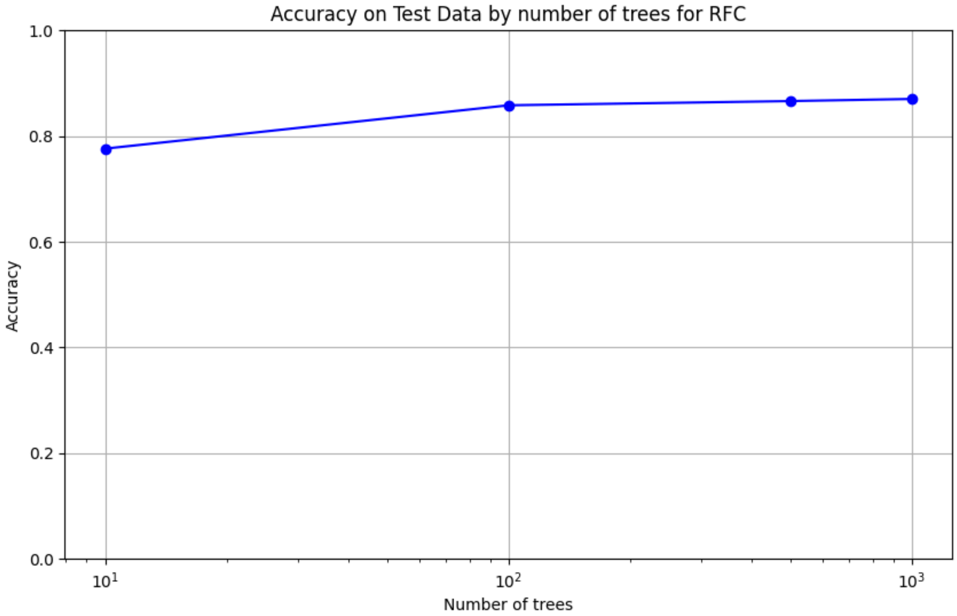

Figure 3: plot for tables

(d) Please implement 5-fold cross-validation for hyper-parameter selection using `sklearn` on the training set and select the best parameters for problem (a)-(c) separately and report the corresponding testing accuracy.

In order to implement a 5-fold cross-validation, we will use the following code.

```
n_folds = 5
cv = KFold(n_splits=n_folds, shuffle=True, random_state=1)

space_a = dict()
space_b = dict()
space_c = dict()

space_a["C"] = Cs
space_b["gamma"] = gammas
space_c["n_estimators"] = n_trees

lsvc = LinearSVC(random_state = 7, dual=False)
rsvc = SVC(random_state = 7, C=1.0, kernel='rbf')
rf = RandomForestClassifier(random_state = 7)

lsvc_search = GridSearchCV(lsvc, space_a, scoring = 'accuracy', n_jobs = -1, cv = cv)
rsvc_search = GridSearchCV(rsvc, space_b, scoring = 'accuracy', n_jobs = -1, cv = cv)
rf_search = GridSearchCV(rf, space_c, scoring = 'accuracy', n_jobs = -1, cv = cv)

lsvc_fit = lsvc_search.fit(X_train_norm, y_train)
rsvc_fit = rsvc_search.fit(X_train_norm, y_train)
rf_fit = rf_search.fit(X_train_norm, y_train)
best_lsvc = lsvc_fit.best_estimator_.C
best_rsvc = rsvc_fit.best_estimator_.gamma
best_rf = rf_fit.best_estimator_.n_estimators;

y_pred_lsvc = lsvc_fit.predict(X_test_norm)
y_pred_rsvc = rsvc_fit.predict(X_test_norm)
y_pred_rf = rf_fit.predict(X_test_norm)
```

```
acc_lsvc = metrics.accuracy_score(y_pred_lsvc, y_test)
acc_rsvc = metrics.accuracy_score(y_pred_rsvc, y_test)
acc_rf = metrics.accuracy_score(y_pred_rf, y_test)
```

After running the code, we get these optimal values (best parameters).

lsvc	rsvc	rf
$C = 100$	$\gamma = 0.1$	n_estimator = 500

With the given best predictors, we can get the accuracies as,

lsvc	rsvc	rf
0.846	0.806	0.866

2) Machine Learning Question [2 pts]

Some questions have Multiple Choices

2. a) Question 1: What is the difference between Linear Regression and Support Vector Regression (SVR)

(A) Linear Regression can only model linear relationships, while SVR can model both linear and non-linear relationships.
(B) Linear Regression is a classification technique, while SVR is a regression technique.
(C) Linear Regression does not use a margin for error, while SVR incorporates an error margin in the regression.
(D) Linear Regression is computationally more complex than SVR.

2. b) Question 2: Choose the disadvantage(s) of Decision Trees.

(A) Decision Trees are robust to outliers.
(B) Factor analysis.
(C) Decision Trees are prone to overfit.
(D) All of the above.