DATA7703 Practical 11

2021 Semester 2

1. We compare several different methods for computing feature importance scores in this question.

(a) The iris dataset contains measurements of petal length, petal width, sepal length and sepal width for 150 iris flowers belonging to three different species.

Load the dataset from `sklearn.dataset`. For each feature, generate the boxplots of its values for each of the three different species. Based on the boxplots, comment on the expected importances of the features.

(b) Train a random forest classifiers on the entire dataset, and compute the impurity-based importances and permutation importances for the features. Repeat this 10 times, and report the means and standard errors of the importance scores. Compare the two different feature importance scores.

Read the documentation of `RandomForestClassifier` on how to compute the impurity-based importances, and read the documentation of `sklearn.inspection.permutation_importance` on how to compute the permutation importances.

(c) Train a logistic regression model on the iris dataset. Compute the permutation importance of the features using the logistic regression model. Compare the importance values with those in (b).

2. We study Gaussian process regression in this question.

(a) The following code fits and evaluates a linear regression model and a Gaussian process model with Matern kernel on the California Housing price dataset. Read and run the code to understand how it works. Does the Gaussian process assume that the observations are noisy? How does Gaussian process compare with linear regression?

```
from sklearn.datasets import fetch_california_housing
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import WhiteKernel, ConstantKernel, RBF,
    Matern
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X, y = fetch_california_housing(return_X_y=True)
X_tr, X_ts, y_tr, y_ts = train_test_split(X, y, test_size=0.8, random_state=42)

# train and test a linear regression kernel
ols = LinearRegression()
ols.fit(X_tr, y_tr)
print('Linear regression R2 - train: %.2f; test: %.2f' % (ols.score(X_tr, y_tr),
    ols.score(X_ts, y_ts)))

# train and test a GP with Matern kernel
```

```
kernel = Matern()
gpr = GaussianProcessRegressor(kernel=kernel, random_state=0)
gpr.fit(X_tr, y_tr)
print('Gaussian process R2 - train: %.2f; test: %.2f' % (grp.score(X_tr, y_tr),
    gpr.score(X_ts, y_ts)))
```

(b) Try each of the following modifications separately on the Gaussian process model. Based on the results, comment on the factors which are important for optimizing GP's performance.

(i) Replace the Matérn kernel with the RBF kernel

```
kernel = RBF()
```

(ii) Replace the Matérn kernel with a scaled Matérn kernel, and use a noisy observation model.

```
kernel = ConstantKernel()*Matern()+WhiteKernel()
```

(iii) Normalize each feature to have mean zero and unit variance.

```
scaler = StandardScaler()
X_tr_scale = scaler.fit_transform(X_tr)
X_ts_scale = scaler.transform(X_ts)
```

(iv) Combine (ii) and (iii).

(v) (iv) without observation noise.