# project1

September 23, 2020

[1]:
```
import numpy as np
import pandas as pd
```

[2]:
```
class LinearRegression:

    def __init__(self, lr=0.001, n_iters=1000):
        self.lr = lr
        self.n_iters = n_iters
        self.weights, self.bias = None, None

    def fit(self, X, y):
        # First, randomly initialize the weights
        n_samples, n_features = X.shape
        self.weights = np.random.rand(n_features)
        # I'm setting the bias (intercept) to zero
        self.bias = 0.0

        for _ in range(self.n_iters):
            y_approx = np.dot(X, self.weights) + self.bias

            # Gradient calculations w.r.t "w" and "b"
            dw = float(1/n_samples) * np.dot(X.T, (y_approx - y))
            db = float(1/n_samples) * np.sum(y_approx - y)

            # Update the params
            self.weights -= self.lr * dw
            self.bias -= self.lr * db

    def predict(self, X):
        y_pred = np.dot(X, self.weights) + self.bias
        return y_pred
```

[3]:
```
class Dataset:

    def __init__(self):
        self.data = self.read_data()
```

```
    # Read data and make dataset
    def read_data(self):
        data = pd.read_csv('data/iris.data', names=['f1', 'f2', 'f3', 'f4',␣
↪'species'])
        # Replace class strings with numbers to apply linear regression
        data.loc[(data.species == 'Iris-setosa'), 'species'] = -1.0
        data.loc[(data.species == 'Iris-versicolor'), 'species'] = 0.0
        data.loc[(data.species == 'Iris-virginica'), 'species'] = 1.0
        return data

    def split_data(self, test_pct=0.2):
        train_df = self.data.sample(frac=1-test_pct)
        test_df = self.data.drop(train_df.index)
        X_train = train_df[['f1', 'f2', 'f3', 'f4',]].to_numpy().
↪astype('float64')
        y_train = train_df[['species']].to_numpy().astype('float64').ravel()
        X_test = test_df[['f1', 'f2', 'f3', 'f4',]].to_numpy().astype('float64')
        y_test = test_df[['species']].to_numpy().astype('float64').ravel()
        return X_train, y_train, X_test, y_test
```

[4]:
```
dataset = Dataset()
```

[5]:
```
def round_class(prediction):
    if prediction <= -0.33:
        return -1
    elif prediction > -0.33 and prediction < 0.33:
        return 0
    else:
        return 1
```

[6]:
```
def test_model(regressor, X_test, y_test):
    total, correct = 0, 0
    mse = 0
    for i, x in enumerate(X_test):
        y_pred = regressor.predict(x)
        y_true = y_test[i]
        y_error = y_pred - y_true
        mse += y_error**2
        if round_class(y_pred) == int(y_true):
            correct += 1
        total += 1

    return correct/total, mse/len(X_test)
```

[14]:
```python
def run_k_fold(dataset, k=5):
    accs = []
    mses = []
    for i in range(k):
        X_train, y_train, X_test, y_test = dataset.split_data(test_pct=0.2)
        regressor = LinearRegression(lr=0.001, n_iters=1000)
        regressor.fit(X_train, y_train)
        acc, mse = test_model(regressor, X_test, y_test)
        print(f'{i}. Accuracy: {acc:.4f} || MSE: {mse:.4f}')
        accs.append(round(acc, 4))
        mses.append(round(mse, 4))
    print(f'\nAccuracy every fold: {accs}')
    print(f'Average classification accuracy: {sum(accs)/len(accs):.4f}')
    print(f'\nMSE every fold: {mses}')
    print(f'Average MSE: {sum(mses)/len(mses):.4f}')
```

[22]:
```python
run_k_fold(dataset, k=5)
```

```
0. Accuracy: 1.0000 || MSE: 0.0410
1. Accuracy: 0.9667 || MSE: 0.0674
2. Accuracy: 0.9667 || MSE: 0.0539
3. Accuracy: 0.9333 || MSE: 0.0651
4. Accuracy: 0.9667 || MSE: 0.0606

Accuracy every fold: [1.0, 0.9667, 0.9667, 0.9333, 0.9667]
Average classification accuracy: 0.9667

MSE every fold: [0.041, 0.0674, 0.0539, 0.0651, 0.0606]
Average MSE: 0.0576
```

[ ]:
```python

```

## 0.1 STEPS TO RUN PROJECT 1

### 0.1.1 Set up a virtual environment on Python3 (>3.6)

1. Create a virtualenv
   - `python3 -m venv env`
2. Activate the virtualenv
   - `source env/bin/activate`

### 0.1.2 Install required packages for the program

- `pip install numpy pandas`

OR

- `pip install -r requirements.txt`

### 0.1.3 Run the code

- `python project1.py`

**NOTE: I have used f-string to print values in my code. Please try to use the latest version of Python3 (>=3.6) that supports f-string.**

[ ]: