
Machine learning internship

# Disease prediction

Samuel MATIA

# Project presentation

The objective of this project is to develop a predictive model that can accurately classify individuals into
diseased or non-diseased categories based on their health attributes. By leveraging machine learning
algorithms, we aim to create a reliable tool that healthcare providers can use to assist in disease diagnosis and prognosis.

# Dataset overview

We have access to a dataset containing multiple health-related attributes such as cholesterol levels, blood cell counts, hormone levels, and other
physiological measurements. The dataset also includes information on whether the individual has been
diagnosed with a specific disease or not.

# Data shape

## Import data

```
train = pd.read_csv('Train_data.csv')
test= pd.read_csv('test_data.csv')

print(f'train data dims : {train.shape}')
print(f'test data dims : {test.shape}')
```

```
train data dims : (2351, 25)
test data dims : (486, 25)
```

# Preprocessing

- Label encoding
- Standardisation

# Label Encoding

```
disease = ['Diabetes', 'Thalasse', 'Anemia', 'Thromboc']
non_disease = ['Healthy']
```

```
train['Disease'] = np.where(train['Disease'].isin(non_disease), 0,1)
test['Disease'] = np.where(test['Disease'].isin(non_disease), 0,1)
```

# Standardisation

```
y_train = train.pop('Disease')
y_test = test.pop('Disease')
```

```
scaler = StandardScaler()
x_train = scaler.fit_transform(train)
x_test = scaler.fit_transform(test)
```

# Model

## Model

```
model = ExtraTreesClassifier(n_estimators=288, random_state = 0)
#the best n_estimators finded manually, GridSearchCV was inefficient in this case
```

```
model.fit(x_train, y_train)
model.score(x_train, y_train)
```

```
1.0
```

```
predictions = model.predict(x_test)
```

# Metrics

## Evaluation metrics

```
print(f'accuracy : {round(accuracy_score(predictions, y_test)*100, 2)} %')
print(f'precision : {round(precision_score(predictions, y_test)*100, 2)} %')
print(f'recall : {round(recall_score(predictions, y_test)*100, 2)} %')
print(f'f1_score : {round(f1_score(predictions, y_test)*100, 2)} %')
```

```
accuracy : 97.94 %
precision : 98.96 %
recall : 98.96 %
f1_score : 98.96 %
```


Machine learning internship

Samuel MATIA