과목 명: 시스템프로그래밍

담당 교수 명: 박 운 상

<<Assignment 6>>

**서강대학교 컴퓨터학과**

**[20141603]**

**[황순]**

목 차

1. 프로그램 개요 3

2. 프로그램 설명 3

2.1 프로그램 흐름도 3

3. 모듈 정의 3

3.1 모듈 이름 : Multiclass Decision Forest 3

3.1.1 기능 3

3.1.2 사용 변수 3

3.2 모듈 이름: train model 3

3.2.1 기능 3

3.2.2 parameter 3

3.3 모듈이름: score model 4

3.3.1 기능 4

3.3.2 사용변수 4

3.4 모듈이름: evaluate model 4

3.4.1 기능 4

3.4.2 사용변수 4

4. 알고리즘 구성 이유 4

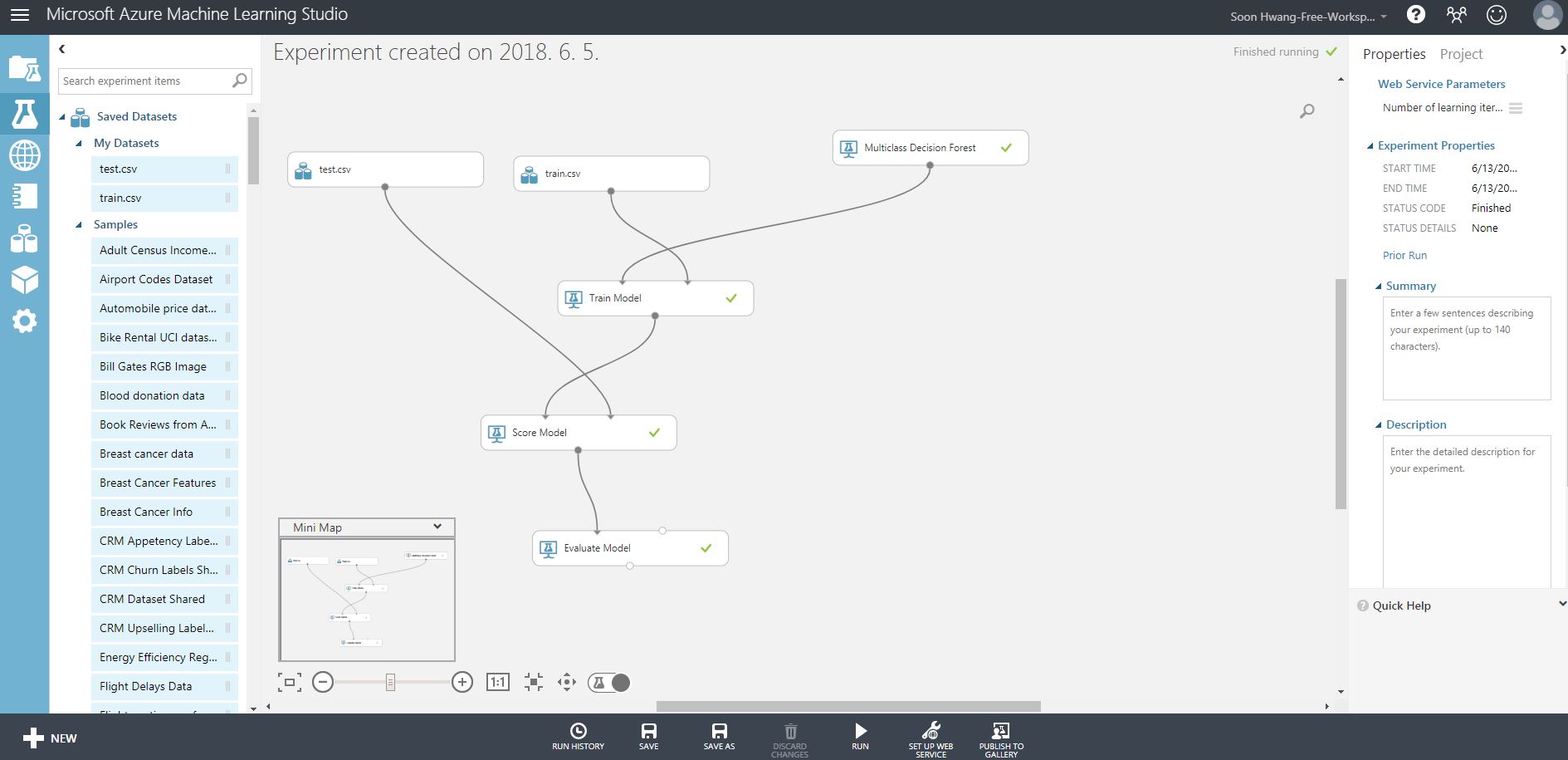
4.1 결과 4

# 프로그램 개요

본 프로젝트 에서는 사용자가 직접 데이터를 가지고 학습 및 서비스를 진행할 수 있는 Azure Machine Learning 서비스를 이용하여 ML 모델을 구축한다.

# 프로그램 설명

## 프로그램 흐름도



# 모듈 정의

## 모듈 이름 : Multiclass Decision Forest

### 기능

Train.csv에서 Class 가 9개, 93개의 feature를 가지고 class마다 702개의 데이터를 가지고 있으므로 두개의 데이터를 처리하는 binary가 아닌 Multiclass를 사용하였다. Decision tree를 만들어서

### 사용 변수

없음

## 모듈 이름: train model

### 기능

Classifaction 한 데이터를 train하는 모델이다. Tagged data가 있고 parameter를 설정하면 train을 할 수있다.

### parameter

target

## 모듈이름: score model

### 기능

Trained Classfication model 을 기반으로 예측 데이터를 만든다..

### 사용변수

없음

## 모듈이름: evaluate model

### 기능

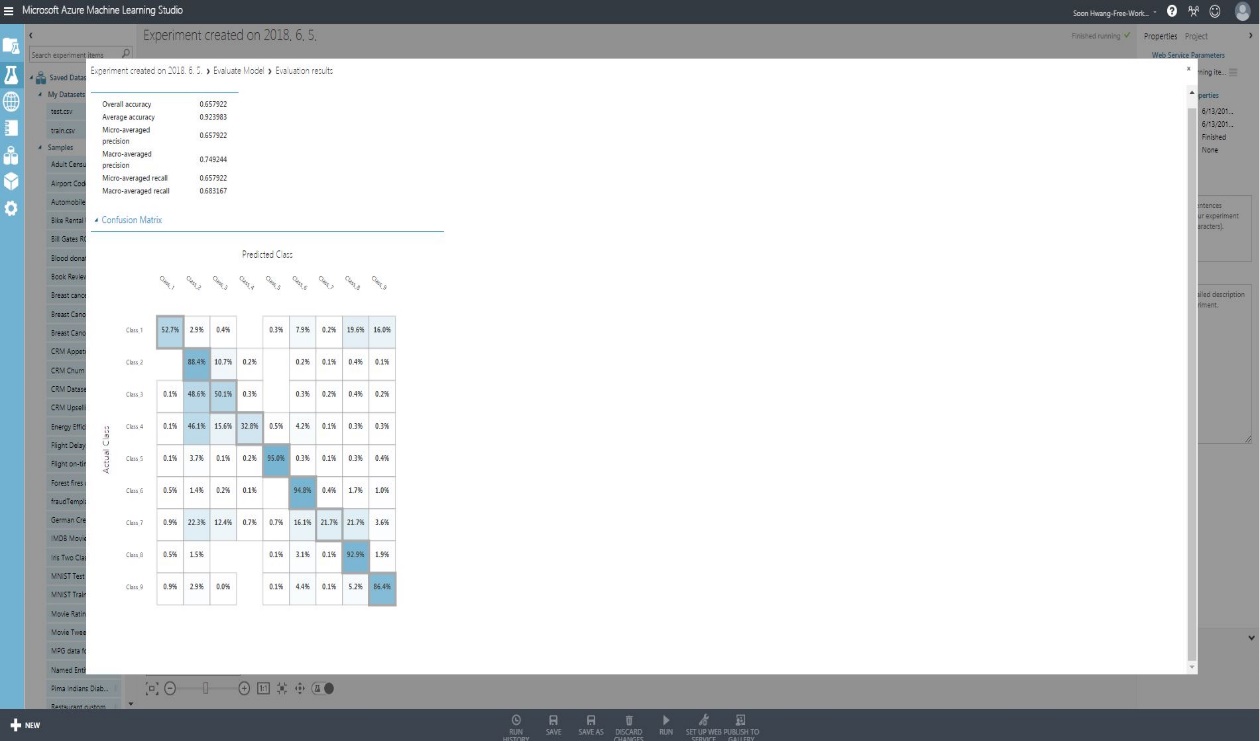
Trained model의 정확도를 측정한다. 모델로부터 만들어진 스코어를 포함한 dataset을 제공하면 evaluate 모델은 industry-standard evaluation metrics 의 set을 계산한다.

# 알고리즘 구성 이유

주어진 Train data와 test data를 이용하여 machine learning 을 수행하였다. Data가 두가지 class 나 feature를 갖는 것이 아니고, Train data에서 9개의 class가 존재하고 각 class는 93개의 feat를 가지는 703개의 data를 가지기 때문에 two class decision 이 아닌 multiclass decision을 선택하였다. Two class 는 두가지 경우의 수가 존재할 때만 쓰기 때문에 이번 프로젝트에는 적절하지 않다고 생각했다. 또한 모든 data가 같은 형식과 크기를 가지고 있으므로 bagging이 아닌 replicate를 사용하였다. Bagging은 주어진 훈련 데이터에서 중복을 허용하여 원 데이터와 같은 크기의 데이터를 만드는 과정을 말한다. 따라서 replicate가 더 적합하다고 판단하였다. Multiclass decision tree에서 최대허용 깊이는 하나의 트리에서 루트 노드부터 종단 노드까지 최대 몇개의 노드(테스트)를 거칠 것인지를 결정하는 매개변수이다. 최대 허용 깊이가 작으면 과소적합(underfitting)이 일어나고, 최대 허용 깊이가 크면 과대적합(overfitting)이 일어나기 때문에 적절한 값을 설정하는 것이 중요하다. 따라서 class의 데이터 수인 702를 넣었다. Number of decision trees는 총 포레스트를 몇 개의 트리로 구성할 지를 결정하는 매개변수이다. 포레스트가 작으면 트리들을 구성하고 테스트 하는데 걸리는 시간이 짧은 대신, 일반화 능력이 떨어져 임의의 입력 데이터 포인트에 대해 틀린 결과를 내놓을 확률이 높다. 반면에 포레스트의 크기가 크다면 훈련과 테스트 시간은 증가하지만, 포레스트의 결과값은 각 트리의 결과들에 평균을 취한 것으로 큰 포레스트의 결과값은 작은 포레스트보다 비교적 연속적이며 일반화 능력이 우수하다. 따라서 적절한 크기로 data의 개수인 702개로 설정하였다. Number of random split per node 도 역시 적절한 수를 찾다보니 702로 설정하였다. Train model을 거친 후 score model에 넣은 다음 test.csv의 정보와 함께 prediction을 만들고 evaluate에서 그것들을 기반으로 set of metrics를 만든다.

## 결과

Train.png



Test.png

