과목 명: 시스템프로그래밍

담당 교수 명: 박 운 상

<<Assignment 6>>

**서강대학교 컴퓨터공학과**

**[학번]20161665**

**[이름]황세현**

<목 차>

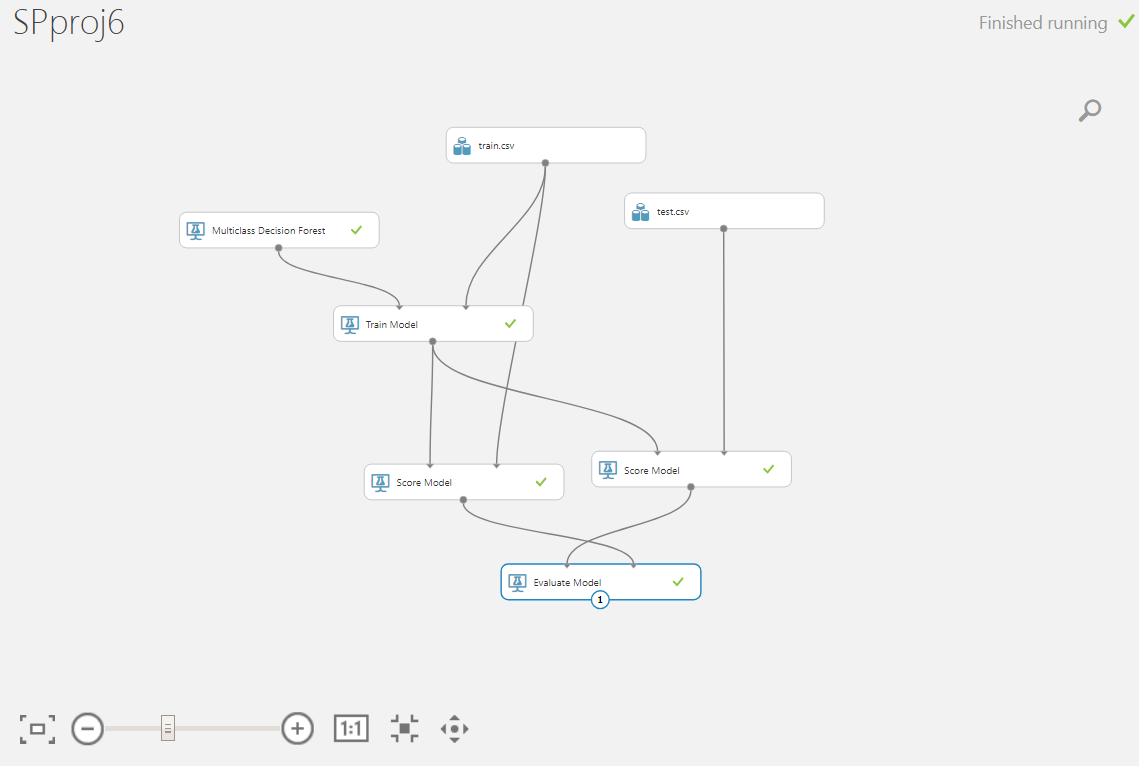
1. **프로그램 개요**
2. **프로그램 설명**
   1. **프로그램 흐름도**
3. **알고리즘 선택 과정**

# 프로그램 개요

Azure machine learning 플랫폼 환경을 통하여 사용자가 직접 데이터를 가지고 학습 및 서비스를 진행할 수 있는 Machine learning 모델을 구축한다.

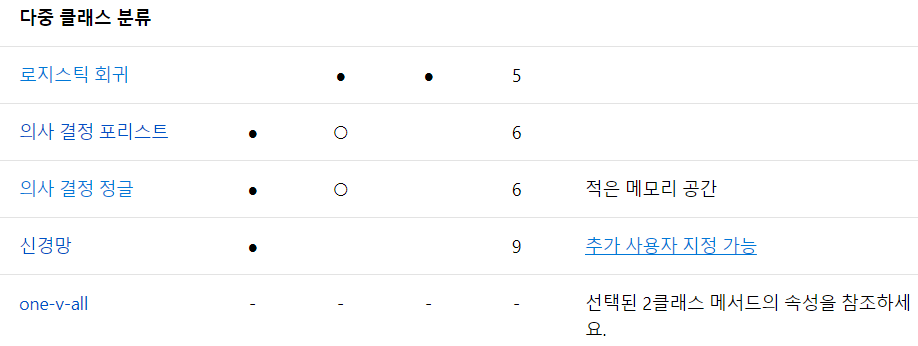
# 프로그램 설명

## 프로그램 흐름도

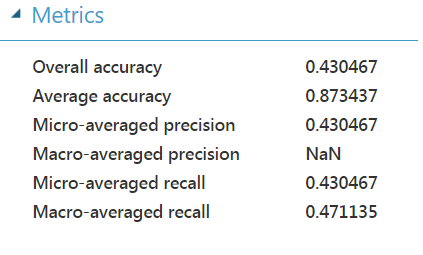
decision 알고리즘을 이용하여 주어진 데이터를 알맞은 클래스에 위치할 수 있도록 설계하는 프로그램이다. Evaluation model의 한 쪽에는 train데이터를 이용하여 학습한 결과를 test결과와 비교하였고, 다른 쪽에는 train 데이터를 이용하여 학습한 결과를 자기 자신과 비교하도록 하였다.

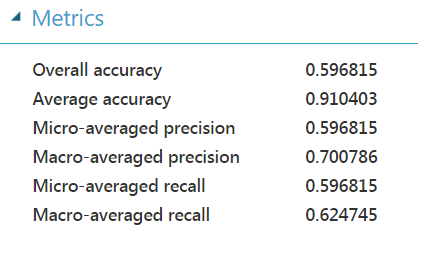
# 알고리즘 선택 과정

우선 주어진 조건을 만족시키기 위해서는 데이터를 총 9개의 클래스로 나눠야 하므로 classification 알고리즘들 중 two-class 알고리즘보다는 multiclass 알고리즘이 더 적합하다고 판단하였다. azure machine learning 서비스에서 제공하는 다중 클래스 분류 알고리즘에는 총 5개가 있는데 각각의 경우에 있어서 다음과 같은 특성을 갖고있었다. 항목은 차례대로 정확도-학습시간-선형성-매개변수 순이다.

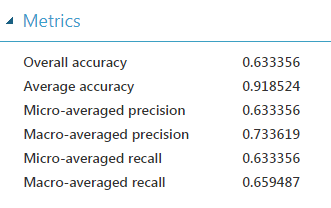


이 항목들 중에서 정확도에 가장 중점을 두었더니 의사 결정 포리스트와 의사 결정 정글로 알고리즘의 범위가 좁혀졌었다. 먼저 decision jungle 알고리즘을 선택하여 default파라미터를 주고 학습시킨 결과는 다음과 같았다.

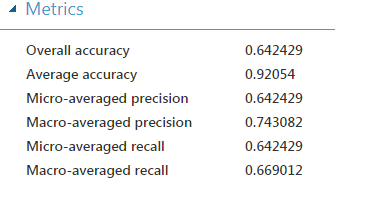


같은 방법으로 decision forest에 대해 default 파라미터를 주고 학습시킨 결과는 다음과 같이 나타났다. 

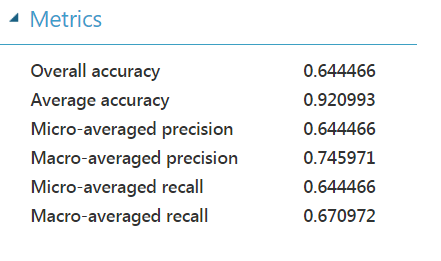
한 눈에 보기에도 decision forest알고리즘의 정확도가 훨씬 높았으므로 decision forest 알고리즘의 인자 값을 조정하면 정확도를 좀 더 높일 수 있을 것 같다고 판단하였다. 사용자가 조정할 수 있는 decision forest 알고리즘의 파라미터는 결정 트리의 개수, 결정 트리의 최대 깊이, 노드 당 랜덤 분할의 개수, 리프 노드당 샘플의 최소개수 등이 있었다. 처음 디폴스 세팅 값은 차례대로 8/32/128/1이었다. 결정 트리의 특성 상 특정 질문에 대한 응답을 따라가는 방식으로 계속해서 데이터를 분류해내기에 결정 트리의 개수가 늘어나면 보다 데이터를 세분화할 수 있을 것이라고 판단하였다. 또한 트리를 구성하는 노드당 랜덤한 분할이 많아질수록 데이터들이 편중되지 않고 잘 나눠질 거라고 생각하여 결정 트리의 개수는 8 🡪 16, 노드 당 랜덤 분할의 개수는 128 🡪 2048로 늘려보았다. 그 결과는 다음과 같았다.



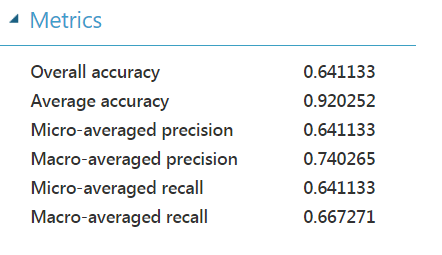
예상대로 전체적인 정확도가 어느정도 상승했음을 볼 수 있었다. 따라서 같은 생각에 근거하여 결정 트리의 개수를 다시 한번 늘려보았더니 (16 🡪 32)다음과 같은 결과가 나왔다.



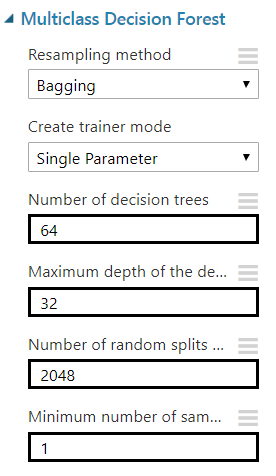
정확도의 상승 비율이 줄어들기는 했지만 여전히 소폭 상승했음을 확인할 수 있었다. 다시한번 트리의 개수를 늘려보았더니(32🡪64) 다음과 같은 결과를 확인할 수 있었다.



이 상태에서 노드 당 랜덤 분할의 수를 2배로 늘렸더니 다음과 같이 정확도의 비율이 소폭 감소하는 것을 확인할 수 있었다. 즉, 트리의 개수나 분할의 개수, 트리의 깊이 등 여러 인자들에 있어서 인자 값을 무조건 늘린다고 성능이 좋아지는게 아닌, 최적의 성능을 보이는 인자들이 존재하며 이 범위를 벗어나면 정확도가 줄어든다는 사실을 알 수 있었다.



이 이후에도 트리의 깊이, 리프 노드당 샘플의 최소 개수 등을 조절하며 정확도를 확인해보았으나 정확도가 감소하거나 그대로 유지될 뿐 더 이상 상승하지는 않았다. 따라서 여러 번 실행 끝에 가장 최적의 정확도(0.644466)를 보인 인자 값은 다음과 같다고 결론지을 수 있었다.



최종적인 결과 화면은 다음과 같다.

