

# < TensorFlow 튜토리얼 >

## 1. 에스티메이터

- 사전 제작된 에스티메이터

(<https://www.tensorflow.org/tutorials/estimator/premade?hl=ko>)

이 튜토리얼은 Estimators를 사용해 Tensorflow의 Iris 분류 문제를 해결하는 예제. 에스티메이터는 tensorflow의 고수준 API이며 스케일링이 쉽고 비동기식 훈련을 위해 설계됨.

<https://www.tensorflow.org/tutorials/estimator/premade?hl=ko>

tensorflow의 Keras API는 estimator와 동일한 여러 작업을 수행 할 수 있으며, 좀 더 배우기 쉬운 API로 여겨짐.(즉, 처음 시작은 Keras가 좋음)

### ① 필요 라이브러리 설치

기본 설치(tensorflow, pandas)

### 참 고

#### 머신러닝 용어집

<https://developers.google.com/machine-learning/glossary/?hl=ko#feature>

#### numpy, pandas, matplotlib 비교?

numpy = 리스트, 배열, 매트릭스 연산(같은 데이터 형식)을 위한 라이브러리

pandas = 데이터 프레임(다른 데이터 형식)을 위한 라이브러리

matplotlib = 데이터 시각화 라이브러리(곡선, 원, 막대 등)

(참고 : <https://deancode-wsistory.com/13>, <https://software-creatoristory.com/22>)

#### Feature column(특성 열)이란?

모델에서 특정한 특성을 어떻게 해석해야 하는지 지정하는 함수. 특성 함수 호출에 반환된 출력을 수집하는 목록은 모든 estimator 생성자에게 필수 매개변수.

#### Estimator 인스턴스화를 위해 사전 정의된(pre-made) 종류

1) tf.estimator.DNNClassifier : 멀티클래스 분류를 수행하는 심층모델용

2) tf.estimator.DNNLinearCombinedClassifier : 넓고 깊은 모델

3) tf.estimator.LinearClassifier : 선형모델 기반의 경우

\* 참고사이트 = <https://excelsior-cjh.tistory.com/157>

## 2. 에스티메이터

### - 선형모델

(<https://www.tensorflow.org/tutorials/estimator/linear?hl=ko>)

이 튜토리얼은 tf.estimator API를 사용해 logistic regression model을 훈련하는 예제. 다른 더 복잡한 알고리즘의 기초로 사용되며, 타이타닉 dataset을 불러와 사용.

#### ① 필요 라이브러리 설치

기본(tensorflow, numpy, pandas, matplotlib, IPython)  
\$ pip3 install -q sklearn  
→ Scikit-learn(사이킷런)는 python에서 가장 많이 쓰이는 머신러닝 라이브러리 중 하나로 분류(classification), 회귀(regression), 군집화(clustering), 의사결정 트리(decision tree) 등 다양한 머신러닝 알고리즘을 적용할 수 있는 함수들을 제공.  
(<https://cyan91.tistory.com/38>)

#### hist 함수?

탐색적 데이터 분석 단계에서 변수의 분포, 중심, 퍼짐 등을 한 눈에 살펴볼 수 있는 시각화 종류로 히스토그램 사용.  
argument인 bins는 몇 개의 그래프를 그릴 것인지에 대한 변수  
(<https://rfriend.tistory.com/408>)

#### matplotlib plot()함수의 kind argument?

bar	세로막대	barh	가로막대
pie	원형그래프	ked	곡선
hist	히스토그램	box	네모박스

(<https://datascienceschool.net/view-notebook/372443a5cd90a46429c6459bba8b4342c/>)

#### 참 고

#### ROC란?

Receiver Operating Characteristic(수신자 동작 특성) 곡선으로 머신러닝 모델의 성능을 평가할 때 쓰임.

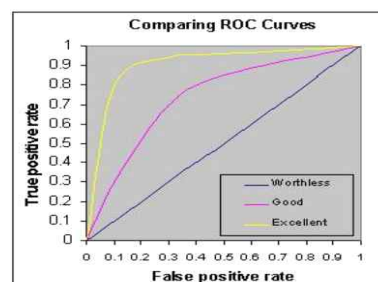
가장 이상적인 모델은 TPR=1이고 FPR=0인 경우

민감도(Sensitivity) = True Positive Rate(TPR)

실제 병 걸린 사람이 양성 판정 받는 경우

특이도(Specificity) = True Negative Rate(TNR)

실제 정상인 사람이 음성 판정 받는 경우



그래프 아래 면적이 넓을수록 그 모델의 성능이 좋다는 것을 의미.

참고: <https://datascienceschool.net/view-notebook/372443a5cd90a46429c6459bba8b4342c/>

(참고 : <https://nittaku.tistory.com/297>)

### 3. 에스티메이터

#### - 부스트된 트리

([https://www.tensorflow.org/tutorials/estimator/boosted\\_trees?hl=ko](https://www.tensorflow.org/tutorials/estimator/boosted_trees?hl=ko))

이 튜토리얼은 의사결정 트리를 사용해 Gradient Boosting model을 학습하는 예제. 회귀 및 분류에서 가장 널리 사용되고 효과적인 머신러닝 방법 중 하나로 여러 트리모델의 예측을 결합한 앙상블 기법 Boosted Tree 모델은 최소의 하이퍼 파라미터 튜닝만으로 고성능을 달성할 수 있어서 인기가 많음.

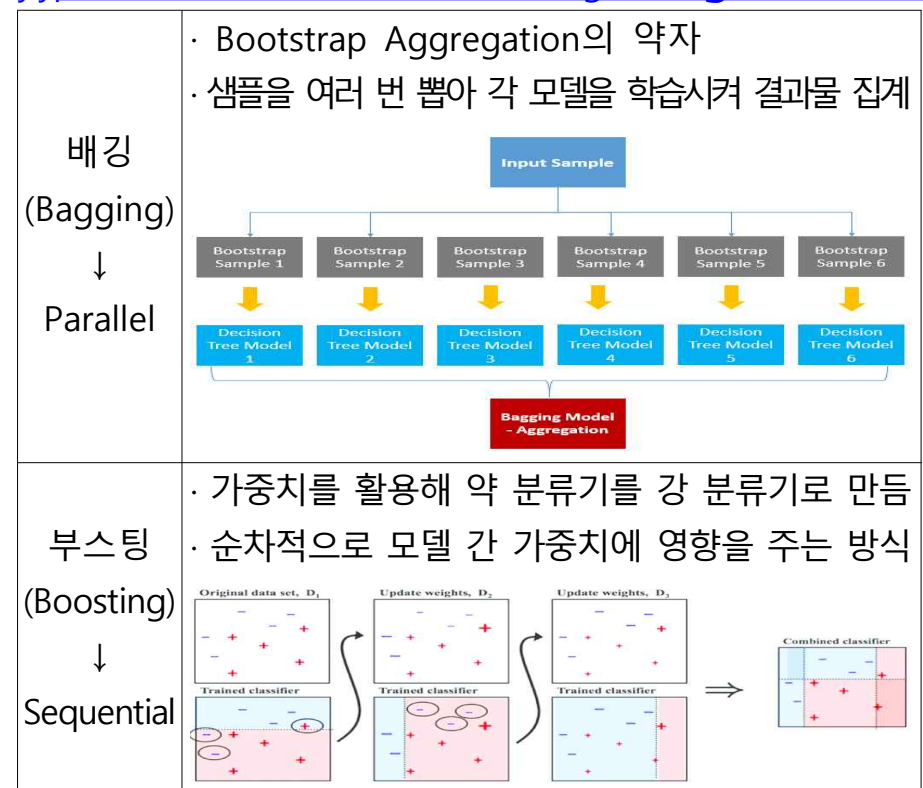
#### ① 필요 라이브러리 설치

기본(tensorflow, numpy, pandas, matplotlib, IPython)

#### 앙상블 기법이란?

Ensemble(조화, 통일) 기법은 여러 개의 데이터 모델을 조화롭게 학습시켜 더 정확한 예측값을 구하는 방법. 여러 개의 Decision Tree를 결합해 하나의 Decision Tree보다 좋은 성능을 내는 머신러닝 기법.(약+약=강) 학습법은 크게 두 가지로 분류.

참고: [https://www.tensorflow.org/tutorials/estimator/boosted\\_trees?hl=ko](https://www.tensorflow.org/tutorials/estimator/boosted_trees?hl=ko)



참 고

\* Bootstrap은 모수의 분포를 추정하는 파워풀한 방법 중 하나로 현재 있는 표본에서 추가적으로 표본을 복원 추출해 각 표본에 대한 통계량을 다시 계산하는 절차(<https://bshistory.com/entry/DATA-12?category=104279>)

#### Gradient Boost란?

부스팅의 대표적인 모델 중 하나로 하나의 leaf에서 시작해 추정값을 계산하는 방식

참고: <https://bshistory.com/entry/DATA-12?category=104279>

#### 편향(Bias)과 분산(Variance)

지도학습 error 처리에 매우 중요한 요소.

편향은 지나치게 단순한 모델로 인한 오류이고(과소적합), 분산은 지나치게 복잡한 모델로 인한 오류(과적합)

참고: <https://bshistory.com/entry/DATA-12?category=104279>

#### 4. 에스티메이터

- 부스트된 트리모델 이해

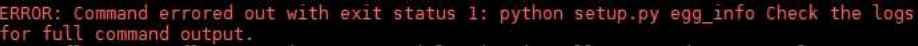
([https://www.tensorflow.org/tutorials/estimator/boosted\\_trees\\_model\\_understanding?hl=ko](https://www.tensorflow.org/tutorials/estimator/boosted_trees_model_understanding?hl=ko))

이 튜토리얼은 Boosted Tree 모델을 locally 및 globally하게 해석하는 방법과 dataset에 어떻게 적합한지 직감적으로 알 수 있는 예제.	
① 필요 라이브러리 설치	기본(tensorflow, numpy, pandas, matplotlib, IPython) \$ pip3 install seaborn → 데이터 시각화를 위한 라이브러리
참 고	<b>부스팅 기법의 이해</b> <a href="https://www.slideshare.net/freepsw/boosting-bagging-vs-boosting">https://www.slideshare.net/freepsw/boosting-bagging-vs-boosting</a>  <b>Boosted Tree 해석력</b> Locally => 단일 데이터 포인트나 분포의 작은 영역 이해 Globally => 모델 전체에 대한 이해 <a href="https://www.tensorflow.org/tutorials/estimator/boosted_trees_model_understanding?hl=ko">https://www.tensorflow.org/tutorials/estimator/boosted_trees_model_understanding?hl=ko</a>  <b>Directional Feature Contribution(방향성 기능 기여, DFC)란?</b> 뭔까요.....  <b>모델 피팅 시각화</b> 아래 공식을 사용해 교육 데이터를 시뮬레이션 및 생성 할 수 있음 z는 종속변수이고(결과), x, y는 feature(독립변수, input variable) $z = x * e^{-x^2 - y^2}$

#### 5. 에스티메이터

- 에스티메이터로 Keras 모델링

([https://www.tensorflow.org/tutorials/estimator/keras\\_model\\_to\\_estimator?hl=ko](https://www.tensorflow.org/tutorials/estimator/keras_model_to_estimator?hl=ko))

이 튜토리얼은 tf.keras 모델에서 Estimator를 생성하는 프로세스의 전체에 대한 예제.	
① 필요 라이브러리 설치	기본(tensorflow, numpy, tensorflow_datasets)  python 모듈 설치 간 egg_info 오류가 날 경우, \$ pip3 install --upgrade setuptools 후 명령어 치면 해결!
참 고	<b>ipywidgets란?</b> 데이터를 분석하거나 분석된 결과를 확인할 때 좀 더 효율적인 작업이 되도록 도와주는 모듈. (참고 : <a href="https://junpyopark.github.io/interactive_jupyter/">https://junpyopark.github.io/interactive_jupyter/</a> )  <b>Keras와 Estimator의 차이?</b> Keras = Tensorflow의 주력 딥러닝 라이브러리(신경망) Estimator = 복잡한 딥러닝 모델을 쉽게 작성할 수 있도록 도와주는 라이브러리( <a href="https://excelsior-gh.tistory.com/176">https://excelsior-gh.tistory.com/176</a> ) (참고 : <a href="http://www.slideshare.net/dupintalk/two-different-estimators-for-tensorflow-2049">http://www.slideshare.net/dupintalk/two-different-estimators-for-tensorflow-2049</a> )