

마다브샤시 / 인간 활동 인식 - 스마트폰 센서 데이터 세트 (공공의)

Human activity recognition, is a challenging time series classification task. It involves predicting the movement of a person based on sensor data and traditionally involves deep domain expertise and methods from signal processing to correctly engineer features from the raw data in order to fit a machine learning model.

☆ 71 stars 🍴 23 forks 🌿 Branches 🏷 Tags 🔄 Activity

☆ Star

🔔 Notifications

<> 암호 ⌚ 문제 🔧 풀 리퀘스트 ▶ 행위 📁 프로젝트 ⚠ 보안 📈 통찰력

👤 주인 🏠 1 지점 🏷 0 태그 🌿 📁

🔍 파일로 이동

Go to file

암호

...

마다브샤시 하르 README.md ...

959a0df · 4 years ago 🕒

📄 .gitignore	초기 커밋	4 years ago
📄 1.인간활동인식_EDA.ipynb	프로젝트 소스 코드: 첫 번째 부분	4 years ago
📄 2. 고전적 ML 예측 모델 적용.ipynb	프로젝트 소스 코드: 두 번째 부분	4 years ago
📄 3. LSTM.ipynb 적용	프로젝트 소스 코드: 세 번째 부분	4 years ago
📄 4. 결론.ipynb	프로젝트 소스 코드: 네 번째 부분	4 years ago
📄 README.md	하르 README.md	4 years ago



Human Activity Recognition Using Smartphones Sensor Dataset

목차

- [개요](#)
- [출처/유용한 링크](#)
- [문제 진술](#)
- [해결책](#)
- [이것은 어떤 유형의 ML 문제입니까?](#)
- [이 문제에 대한 가장 적합한 성과 측정 기준은 무엇입니까?](#)
- [사업 목표 및 제약](#)
- [데이터 개요](#)
 - [1. 데이터가 기록된 방법](#)
 - [2. 데이터는 어떻게 사전 처리됩니까?](#)
 - [3. Y_LABELS\(인코딩됨\)](#)
 - [4. 데이터 디렉토리](#)
- [훈련 및 테스트 비율](#)
- [의제](#)
 - [1. 데이터 분석\(EDA\)](#)
 - [2. 머신 러닝 모델:](#)
 - [a. 로지스틱 회귀](#)
 - [b. 선형 SVC](#)
 - [c. 커널 SVM](#)

- [d. 의사결정 트리](#)
- [e. 랜덤 포레스트 분류기](#)
- [f. 그래디언트 부스트](#)
- [3. 딥러닝 모델:](#)
- [4. 결과 및 결론](#)
- [기술적 측면](#)
- [설치](#)
- [데이터 세트의 빠른 개요](#)

개요

스마트폰은 사용자에게 일상 활동에서 지능적인 지원을 제공하는 첨단 기술과 소통하는 데 있어 우리의 일상 생활에서 가장 유용한 도구가 되었습니다. 컴퓨팅 능력과 상호 연결성을 갖춘 휴대용 작업 프레임워크, 외부 도구와 애플리케이션을 실행하기 위한 애플리케이션 프로그래밍 인터페이스, 모바일 폰은 카메라, GPS, 웹 브라우저 등과 같은 하이라이트를 가지고 있으며, **가속도계**와 **자이로스코프**와 같은 이식된 센서는 클라이언트의 특정 영역, 움직임 및 맥락을 고려하여 애플리케이션을 개선할 수 있도록 합니다.

활동 인식 (AR)은 스마트폰을 사용하여 사람의 활력을 모니터링하는 것입니다. 스마트폰은 더 광범위한 방식으로 사용되고 있으며 스마트 모바일의 센서를 사용하여 인간의 환경 변화를 식별하는 방법 중 하나가 되었습니다. *스마트폰은 자이로스코프 및 가속도계와 같은 센서를 감지하도록 장착되어 있습니다.* 이 장치는 개인의 상태를 검사하는 것으로 시연되었습니다.

인간 활동 인식 (HAR) 프레임워크는 *센서에서 원시 데이터를 수집하고 다양한 딥 러닝 접근 방식을 사용하여 인간의 움직임을 관찰합니다.* 감지된 데이터를 사용하여 그럴듯한 높은 정확도로 인간의 움직임을 식별하기 위한 딥 러닝 모델이 제안됩니다.

UCI 데이터셋 저장소의 HAR 데이터셋을 활용했습니다. 이 데이터셋은 허리에 스마트폰을 두고 다양한 활동을 하는 30명(이 데이터셋에서는 피험자라고 함)에게서 수집했습니다. 데이터는 스마트폰의 센서(**가속도계** 및 **자이로스코프**)를 사용하여 기록합니다. 이 실험은 데이터에 수동으로 레이블을 지정하기 위해 비디오크로 녹화되었습니다.

이 프로젝트는 **걷기, 위로 걷기, 아래로 걷기, 앉기, 서기, 누워 있기**와 같은 **인간의 활동을 예측하는** 모델을 구축하는 것입니다.

소스/유용한 링크

- 블로그 1: <https://www.ijrte.org/wp-content/uploads/papers/v8i1/A1385058119.pdf>
- 블로그 2: <https://machinelearningmastery.com/how-to-model-human-activity-from-smartphone-data/>
- HAR 데이터 세트의 경우: <https://archive.ics.uci.edu/ml/datasets/Smartphone+Dataset+for+Human+Activity+Recognition+%28HAR%29+in+Ambient+Assisted+Living+%28AAL%29>

문제 진술

새로운 데이터 포인트가 주어지면 인간 활동을 예측해야 합니다.

해결책

우리는 " **걷기** ", " **위로 걷기** ", " **아래로 걷기** ", " **서기** ", " **앉기** " 및 " **누워 있기** " 로 레이블이 지정된 인간 활동 인식의 상당히 작은 데이터 집합을 가지고 있습니다. 우리는 UCI 데이터 집합 저장소에서 HAR 데이터 집합을 다운로드했으며 데이터 집합이 두 부분으로 정의되어 있음을 알고 있습니다. 첫 번째는 RAW 데이터 집합이고 두 번째는 도메인 또는 신호 전문가가 사전 설계한 것입니다. **따라서 먼저** 사전 설계된 데이터 집합을 클래식 머신 러닝(ML)으로 사용하여 데이터로부터 학습하고 인간 활동을 예측합니다. **두 번째** 로 RAW 데이터 집합을 딥 러닝 모델로 사용하여 데이터로부터 학습하고 인간 활동을 예측할 수 있습니다.

이 ML 문제는 어떤 유형입니까?

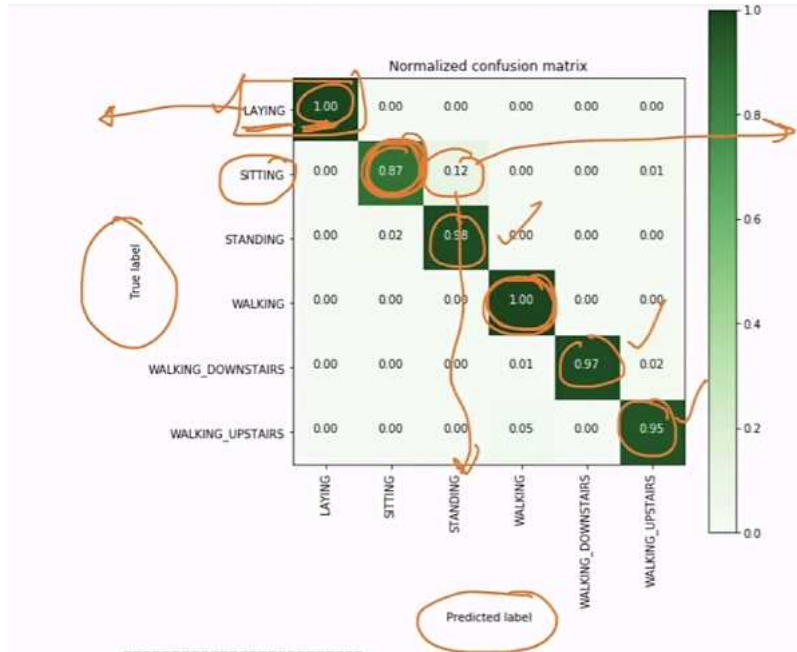
인간 활동 인식은 **까다로운 시계열 분류 작업**입니다. 센서 데이터를 기반으로 사람의 움직임을 예측하는 것이 포함되며 전통적으로 신호 처리에서 원시 데이터로부터 기능을 올바르게 엔지니어링하여 머신 러닝 모델에 맞추는 데 필요한 심층적인 도메인 전문 지식과 방법이 포함됩니다.

또는 다른 말로, **다중 클래스 분류 문제** 라고 부를 수 있습니다. 새로운 데이터 포인트가 주어지면 인간 활동을 예측해야 합니다. 그리고 **각 데이터 포인트는 6가지 활동 중 하나에 해당합니다.**

이 문제에 대한 가장 좋은 성과 지표는 무엇입니까?

- **정확도**: 모든 모델에 대해 이 간단한 "정확도" 측정 항목을 사용하여 전반적인 정확도를 인쇄했습니다.
- **혼동 행렬**: 혼동 행렬이 우리에게 어떤 종류의 오류와 어떤 종류의 혼동이 발생하는지 알려주는 매우 중요한 것입니다.

- 이 프로젝트 뷰에 대한 이 지표를 이해하기 위해 우리는 6개의 클래스 라벨이 있고 종종 모델이 앉아 있는지 서 있는지, 위층으로 걷는지 아래층으로 걷는지 혼동될 수 있다는 것을 알고 있습니다.
- 따라서 혼동 행렬은 알고리즘이나 ML 모델이 어떤 클래스에서 매우 잘 수행하고 있는지, 또는 알고리즘이나 ML 모델이 어떤 클래스에서 혼동을 겪고 있는지 이해하는 데 매우 중요한 방법입니다.



이 혼동 행렬 플롯에서 우리 모델은 누워서기와 걷기 수업에서는 매우 잘 동작하고 서기, 아래층 걷기, 위층 걷기 수업에서는 잘 동작하지만 앉아 있는 수업에서는 혼동을 겪고 있음을 분명히 알 수 있습니다.

- 다중 클래스 로그 손실** : 다중 클래스 로그 손실은 다중 클래스 ML 문제에 매우 중요한 지표라는 것을 알고 있습니다.

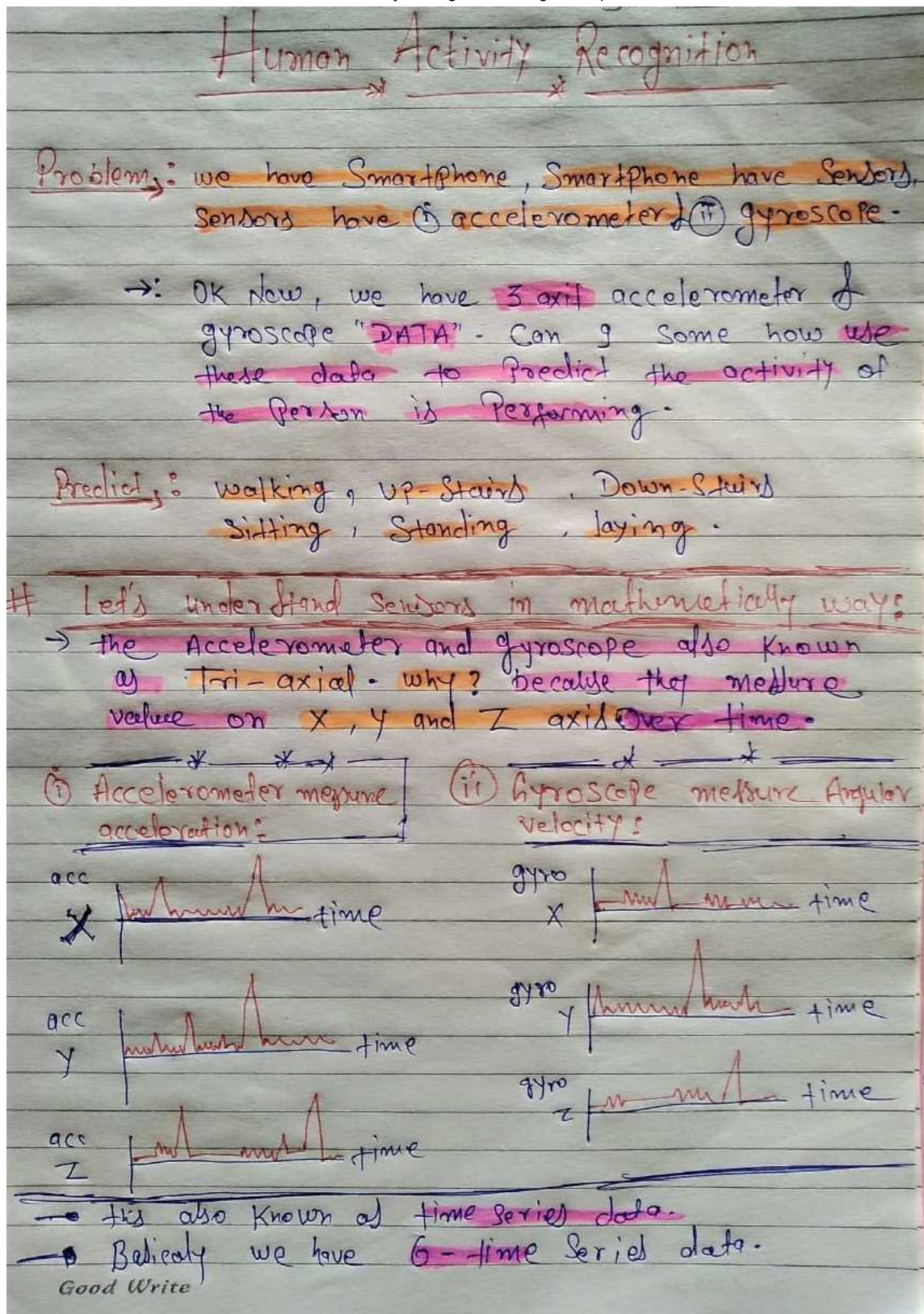
사업 목표 및 제약

- 요즘에는 스마트폰 외에도 Fitbit이나 Apple Watch와 같은 스마트워치를 사용하여 건강을 추적하는 데 도움이 됩니다. 하루 종일 모든 활동을 모니터링하여 소모한 칼로리를 확인합니다. 몇 시간을 잤는지도 확인합니다. 그러나 가속도계와 자이로스코프 외에도 심박수 데이터를 사용하여 활동을 모니터링합니다. 우리는 스마트폰 데이터만 가지고 있으므로 가속도계와 자이로스코프 데이터만 사용하여 사람의 활동을 모니터링할 수 있습니다. 이 소프트웨어는 스마트폰에 다운로드할 수 있는 앱으로 변환할 수 있습니다. 따라서 스마트폰이 있는 사람은 이 앱을 사용하여 건강을 모니터링할 수 있습니다.
- 잘못된 분류로 인한 비용은 매우 높을 수 있습니다 .
- 엄격한 지연 문제는 없습니다 .

데이터 개요

1. 데이터가 기록된 방법

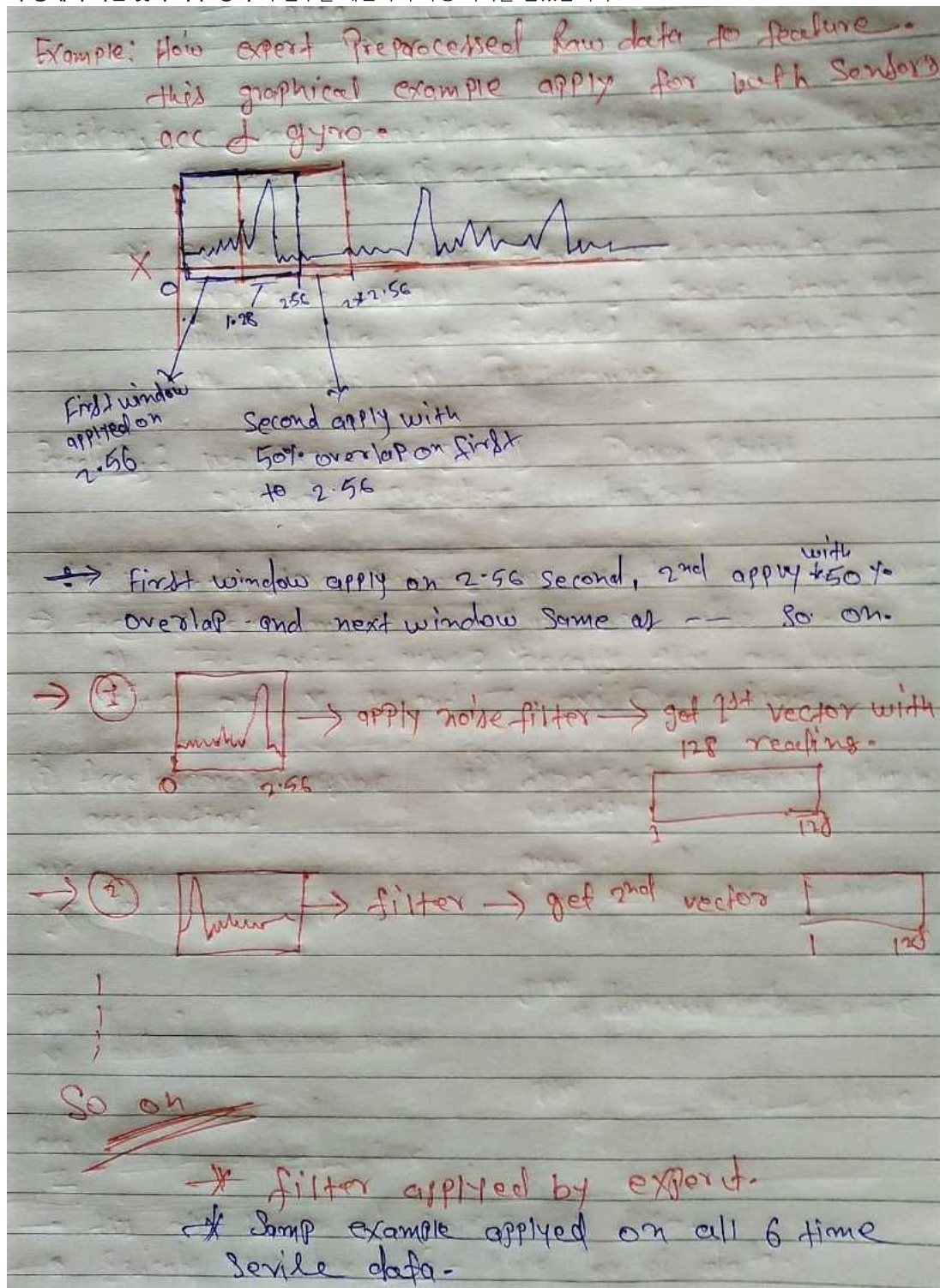
- 30명의 참가자(이 데이터 세트에서는 피험자라고 함) 가 허리에 장착한 스마트폰을 휴대한 채 일상 활동을 수행했습니다 . 이 휴대전화는 두 개의 구현된 센서(가속도계와 자이로스코프) 를 기록하도록 구성되었습니다 . 이 시계열의 경우 기본 연구 책임자가 기능 생성을 수행하고 시리즈에서 2.56초의 고정 너비 창을 이동하여 데이터 세트를 생성했습니다. 창이 50% 겹치므로 결과 포인트는 균일하게 간격이 지정됩니다.(1.28초) . 이 실험은 데이터에 수동으로 레이블을 지정하기 위해 비디오로 녹화되었습니다.
- 스마트폰의 센서(자이로스코프와 가속도계) 를 이용해 가속도계로부터 ' 3축 선형 가속도 '(tAcc-XYZ)를, 자이로스코프로부터 '3축 각속도 '(tGyro-XYZ)를 다양한 변형을 거쳐 수집했습니다 .
 - 이러한 측정 단위에서 접두사 't'는 시간을 나타냅니다.
 - 접미사 'XYZ'는 X, Y, Z 방향 의 3축 신호를 나타냅니다 .
 - 위의 정보를 아래와 같이 그래픽한 방식으로 이해해 보겠습니다.



2. 데이터는 어떻게 사전 처리되니까?

- 원시 센서 데이터를 얻은 후 전문가(도메인 전문가, 신호 엔지니어 전문가)는 이 데이터를 사전 처리하고 몇 가지 유용한 기능을 만듭니다. 저는 전문가가 아니지만 제가 이해하는 바에 따르면 이 데이터가 어떻게 사전 처리되는지 여기서 설명합니다.
- 이러한 센서 신호는 노이즈 필터를 적용하여 사전 처리된 다음 각각 50% 오버랩으로 2.56초씩 고정 너비 창(슬라이딩 창)으로 샘플링됩니다. 즉, 각 창에는 128개의 판독 값이 있습니다.

- 각 창에서 시간 및 주파수 영역의 변수를 계산하여 특징 벡터를 얻었습니다.



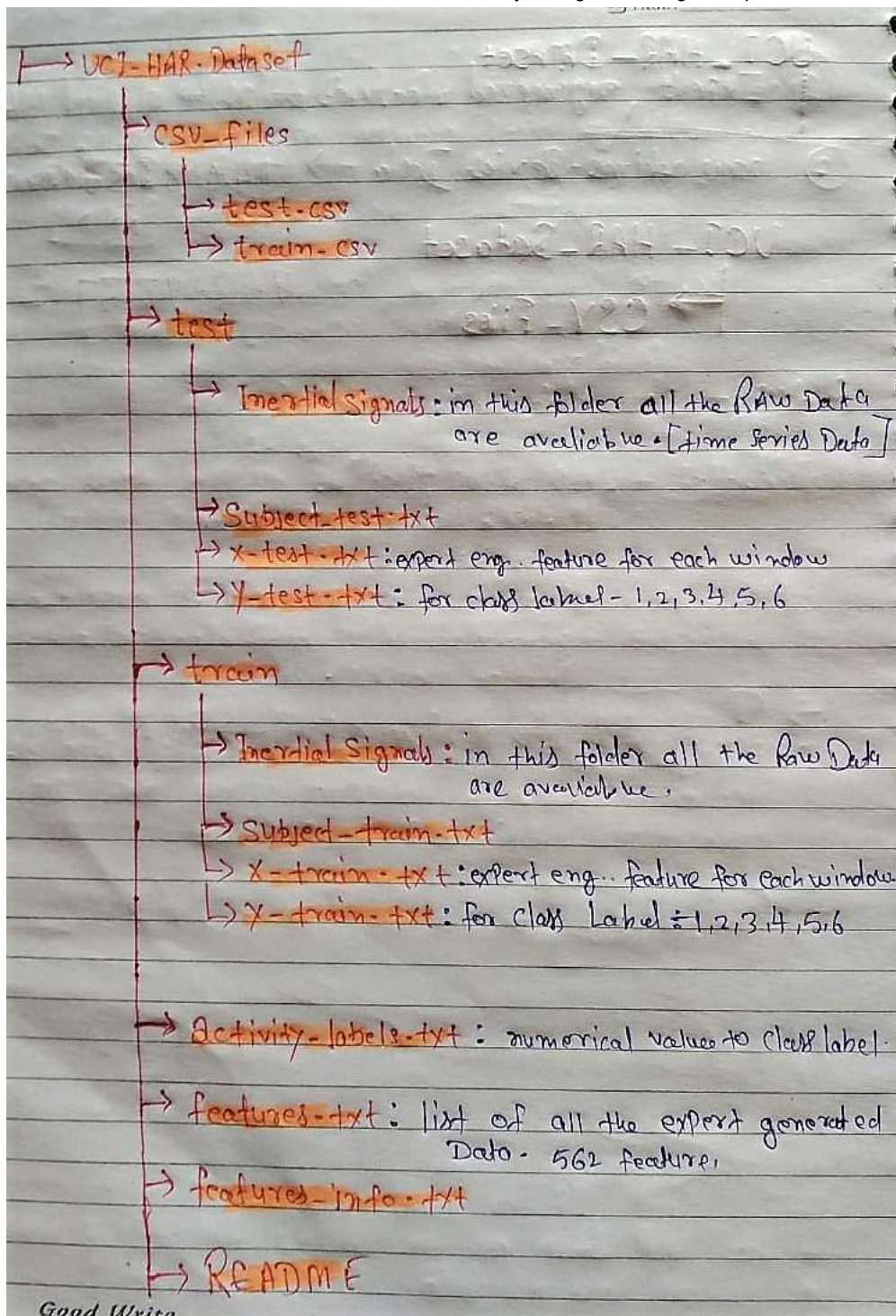
- The acceleration signal was separated into Body and Gravity acceleration signals (`tBodyAcc-XYZ` and `tGravityAcc-XYZ`) using some low pass filter with corner frequency of 0.3Hz.
- After that, the body linear acceleration and angular velocity were derived in time to obtain *jerk signals* (`tBodyAccJerk-XYZ` and `tBodyGyroJerk-XYZ`).
- The magnitude of these 3-dimensional signals were calculated using the Euclidian norm. These magnitudes are represented as features with names like `tBodyAccMag`, `tGravityAccMag`, `tBodyAccJerkMag`, `tBodyGyroMag` and `tBodyGyroJerkMag`.
- Finally, We've got frequency domain signals from some of the available signals by applying a FFT (Fast Fourier Transform). These signals obtained were labeled with prefix 'f' just like original signals with prefix 't'. These signals are labeled as `fBodyAcc-XYZ`, `fBodyGyroMag` etc.,.
- These are the signals that we got so far.
 - `tBodyAcc-XYZ`
 - `tGravityAcc-XYZ`

- tBodyAccJerk-XYZ
 - tBodyGyro-XYZ
 - tBodyGyroJerk-XYZ
 - tBodyAccMag
 - tGravityAccMag
 - tBodyAccJerkMag
 - tBodyGyroMag
 - tBodyGyroJerkMag
 - fBodyAcc-XYZ
 - fBodyAccJerk-XYZ
 - fBodyGyro-XYZ
 - fBodyAccMag
 - fBodyAccJerkMag
 - fBodyGyroMag
 - fBodyGyroJerkMag
- We can estimate some set of variables from the above signals. ie., We will estimate the following properties on each and every signal that we recorded so far.
 - For better remember : we can see above image, EXPERTS apply some filter on each window and get 1st vector, 2nd vector and..... so on. On top of these vector they computed below listed function.
 - **mean()**: Mean value
 - **std()**: Standard deviation
 - **mad()**: Median absolute deviation
 - **max()**: Largest value in array
 - **min()**: Smallest value in array
 - **sma()**: Signal magnitude area
 - **energy()**: Energy measure. Sum of the squares divided by the number of values.
 - **iqr()**: Interquartile range
 - **entropy()**: Signal entropy
 - **arCoeff()**: Autoregression coefficients with Burg order equal to 4
 - **correlation()**: correlation coefficient between two signals
 - **maxInds()**: index of the frequency component with largest magnitude
 - **meanFreq()**: Weighted average of the frequency components to obtain a mean frequency
 - **skewness()**: skewness of the frequency domain signal
 - **kurtosis()**: kurtosis of the frequency domain signal
 - **bandsEnergy()**: Energy of a frequency interval within the 64 bins of the FFT of each window.
 - **angle()**: Angle between to vectors.
 - 우리는 단일 윈도우 샘플에서 신호의 평균을 취함으로써 다른 벡터를 얻을 수 있습니다. 이것들은 angle() 변수에서 사용됩니다.
 - 중력평균
 - t신체평균
 - t바디엑스저크평균
 - t바디자이로평균
 - t바디자이로저크평균

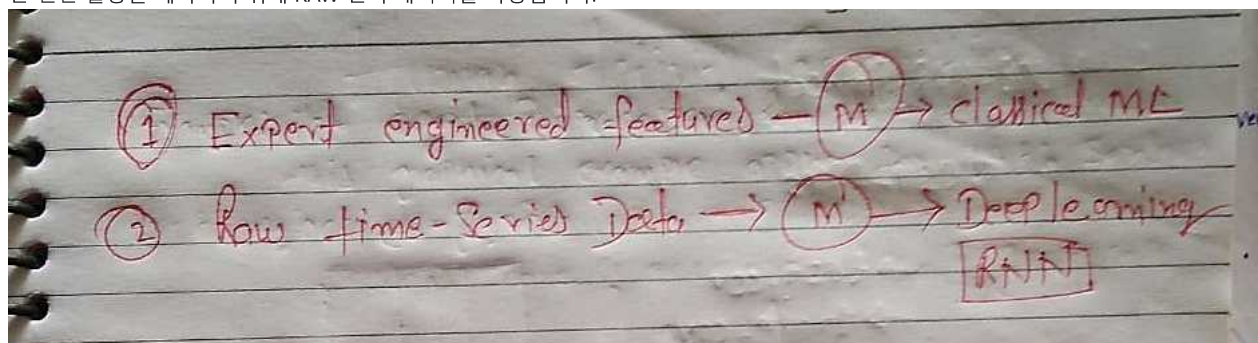
3. Y_Labels(인코딩됨)

- 데이터 세트에서 Y_레이블은 식별자로 1부터 6까지의 숫자로 표현됩니다.
 - 1 로 걷기
 - 2 로 WALKING_UPSTAIRS
 - 3 으로 계단을 걸어 내려가다
 - 4명 으로 앉기
 - 5 로 서있다
 - 6 으로 배치

4. 데이터 디렉토리



- **중요 참고 사항:** 머신 러닝 알고리즘을 적용할 때, 저는 이 전문가들이 만든 특징 데이터를 사용합니다. 딥 러닝 알고리즘을 적용할 때, 저는 인간 활동을 예측하기 위해 RAW 센서 데이터를 사용합니다.



- 데이터는 약 58메가바이트 크기의 단일 zip 파일로 제공됩니다. 이 다운로드의 직접 링크는 다음과 같습니다: [UCI HAR Dataset.zip](https://github.com/MadhavShashi/Human-Activity-Recognition-Using-Smartphones-Sensor-DataSet/blob/master/UCI%20HAR%20Dataset.zip)

훈련 및 테스트 비율

30명의 피험자(자원봉사자)의 데이터를 무작위로 나누어 70%의 자원 봉사자를 훈련 데이터 로 사용 하고 나머지 30%의 피험자의 녹음 내용을 테스트 데이터 로 사용했습니다 . 예를 들어, 21명의 피험자는 훈련 데이터로 사용하고 9명은 테스트 데이터로 사용합니다.

의제

1. 데이터 분석(EDA)

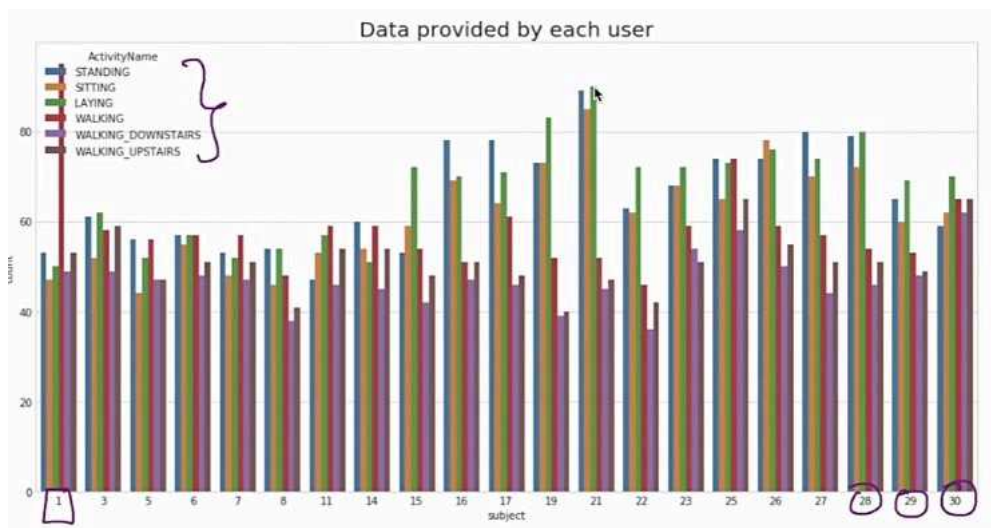
- 아래 데이터 세트에 대한 일부 분석:
- 여기서, 먼저 전문가가 생성한 데이터 세트에 EDA를 수행합니다. 데이터를 이해한 다음 그 위에 머신 러닝 모델을 만듭니다.
- feature.txt 파일을 로드하는 것부터 시작하여, 데이터와 테스트 데이터를 학습하고 이 데이터를 분석합니다.
- 훈련 및 테스트 데이터의 총 데이터 포인트 및 기능 수:

```
train = pd.read_csv('UCI_HAR_dataset/csv_files/train.csv')
test = pd.read_csv('UCI_HAR_dataset/csv_files/test.csv')
print(train.shape, test.shape)
```

Output: (7352, 564) (2947, 564)

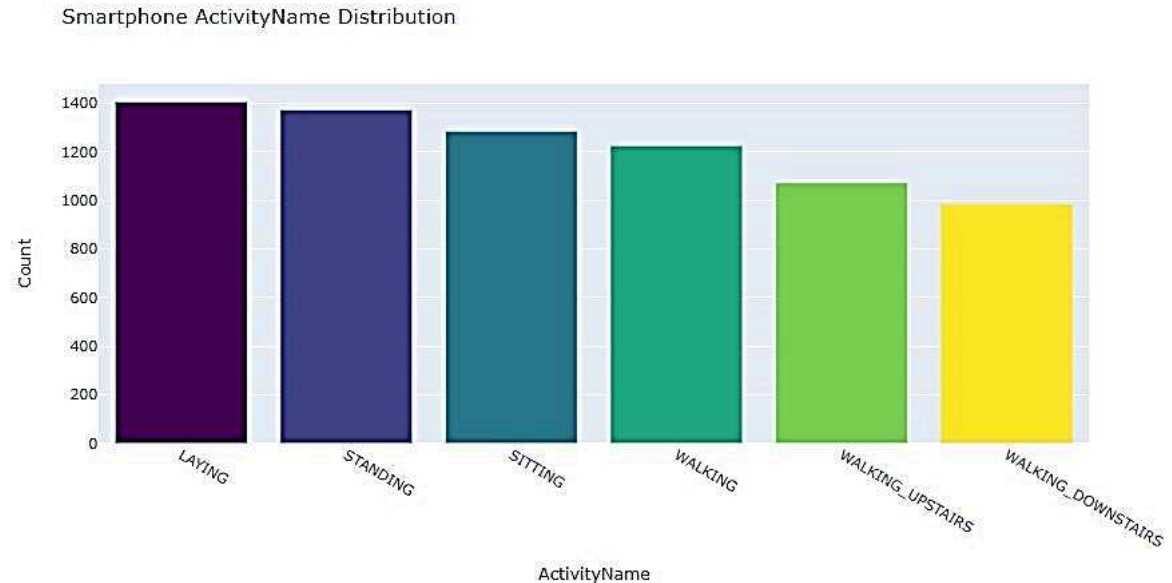
- 참가자 활동 기간을 조사합니다 . 데이터 세트는 과학적 환경에서 생성되었으므로 참가자에 대한 거의 동일한 전제 조건을 가정할 수 있습니다. 그들의 활동 기간을 조사해 보겠습니다.

```
sns.set_style('whitegrid')
plt.rcParams['font.family'] = 'Dejavu Sans'
plt.figure(figsize=(16,8))
plt.title('Data provided by each user', fontsize=20)
sns.countplot(x='subject',hue='ActivityName', data = train)
plt.show()
```



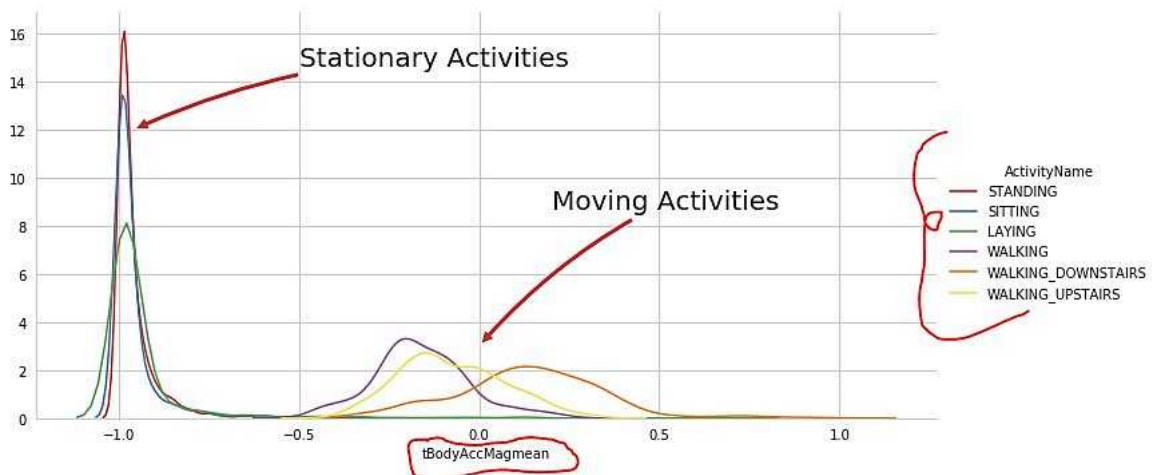
- 거의 모든 참가자는 아래층보다 위층으로 걷는 데 데이터가 더 많습니다. 위아래로 걷는 횟수가 같다고 가정하면 참가자는 위층으로 더 오래 걸어야 합니다.
- 우리는 6개의 클래스 분류가 있다는 것을 알고 있으므로, 큰 문제는 데이터에 불균형이 있는지 아는 것 또는 확인하는 것입니다. 그리고 위의 그래프를 그린 후에 데이터가 균형을 이룬다고 말할 수 있습니다.
- 우리는 모든 과목에서 거의 같은 양의 독서량을 얻었습니다.

- 다음 질문은 각 학급당 얼마나 많은 데이터 포인트를 가지고 있는가입니다.



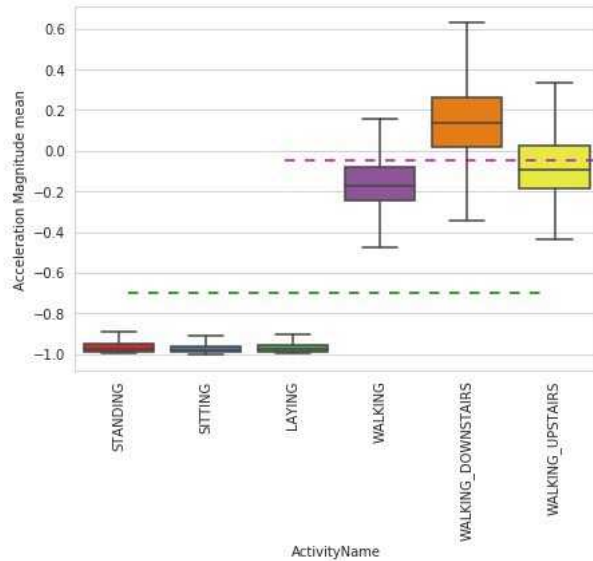
- 데이터는 거의 균형을 이루었습니다.
 - 라벨 수에는 변동이 있지만 라벨 분포는 꽤 균등합니다.
 - 참가자가 위아래로 같은 수의 계단을 걸어야 하고, 스마트폰의 샘플링 속도가 일정하다고 가정하면, 위아래로 걷는 데 필요한 데이터 포인트의 양도 같아야 합니다.
 - 잘못된 데이터가 있을 가능성을 무시하고도 참가자들은 아래쪽으로 약 10% 더 빨리 걷는 것으로 나타났습니다.
- 이제 인간의 정적 및 동적 활동에 대해 알아볼 시간입니다 .

- 정적인 활동(앉기, 서기, 눕기)에서는 동작 정보가 별로 유용하지 않습니다.
- 동적인 활동(걷기, 걷기위로, 걷기아래로)에서는 동작 정보가 중요합니다.
- 여기서 우리는 인간의 정적 및 동적 활동을 더 잘 이해하기 위해 그래프를 그리기 위해 " tBodyAccMagmean "(tBody 가속도 크기 특징 평균 값) 함수를 사용합니다 .

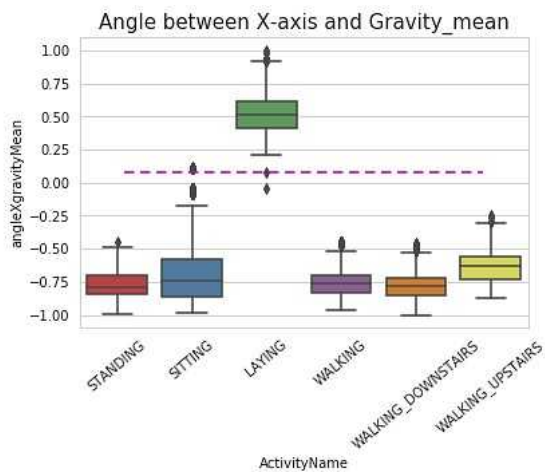


- 우리는 tbodyAccMagmean 특징이 인간의 정적 활동과 동적 활동을 아주 잘 분리한다는 것을 볼 수 있습니다 .

- 이제 우리는 이해를 위해 상자 그림에 인간의 정적 및 동적 활동을 표시합니다.

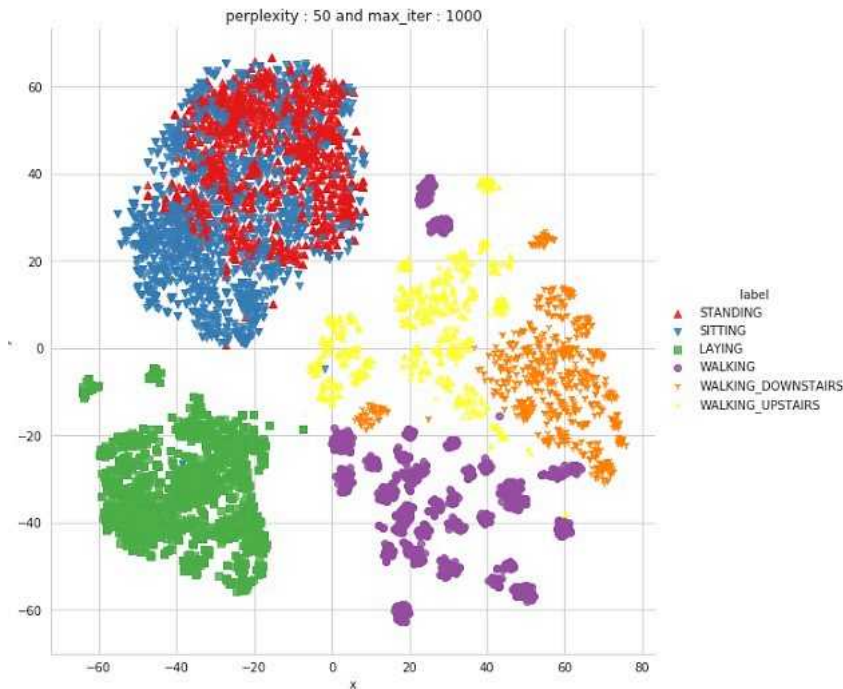


- tAccMean 이 < -0.8 이면 활동은 서 있거나 앉아 있거나 누워 있는 것입니다.
 - tAccMean 이 > -0.6 이면 활동은 Walking , WalkingDownstairs 또는 WalkingUpstairs 입니다.
 - tAccMean > 0.0 이면 활동은 WalkingDownstairs 입니다.
 - 우리는 약간의 오류가 있긴 하지만 활동 라벨의 75%를 분류할 수 있습니다.
- GravityAccelerationComponents 의 위치 도 중요합니다.



- angleX, gravityMean > 0 이면 Activity는 Laying 입니다.
 - 단 하나의 if else 문으로 Laying 활동에 속하는 모든 데이터 포인트를 분류할 수 있습니다.

- 데이터에 t-sne를 적용하여 **활동이 얼마나 분리 가능한지 알아보세요**. 561차원 전문가가 엔지니어링한 피처가 있다는 것을 알고 있습니다. 이제 이러한 피처에 TSNE를 적용하여 이러한 피처가 얼마나 도움이 되는지 확인하세요.



- TSNE 클러스터를 명확히 볼 수 있습니다. "서 있기"와 "앉기"를 제외한 모든 활동이 명확하게 분리되어 있습니다.

2. 머신 러닝 모델:

- **중요 참고 사항**: 561개의 전문가가 설계한 기능을 사용했으며, 이를 기반으로 **고전적인 머신 러닝 모델을 적용 할 것입니다**.
- 제가 적용한 머신 러닝 모델은 다음과 같습니다.

a. 로지스틱 회귀

- 로지스틱 회귀는 분류를 위한 선형 모델입니다. 이 모델에서 단일 시행의 가능한 결과를 설명하는 확률은 로지스틱 함수를 사용하여 모델링 됩니다. 로지스틱 함수는 실수 입력을 받아 0과 1 사이의 값을 출력하는 시그모이드 함수이므로 분류에 이상적입니다.

모델이 훈련 데이터를 너무 면밀하게 학습하면 새로운 데이터에 적합하거나 보이지 않는 관찰을 신뢰할 수 있게 예측하지 못합니다. 이러한 조건을 과적합이라고 하며, 여러 가지 방법 중 하나로 린지(L2) 정규화로 대응합니다. 린지 정규화는 모델 예측 변수가 너무 큰 경우 패널티를 부과하여 작게 만듭니다. 이렇게 하면 모델 분산이 줄어들고 과적합이 방지됩니다.

- **하이퍼파라미터 튜닝**: 교차 검증은 정규화 인자와 중지 기준에 대한 허용 오차(훈련을 언제 중지할지 결정하기 위한)와 같은 모델 매개변수를 튜닝하는 좋은 기술입니다. 여기서 검증 세트는 각 실행(폴드라고 함)에 대한 훈련 데이터에서 보류되고, 모델은 나머지 훈련 데이터에서 훈련된 다음 검증 세트에서 평가됩니다. 이것은 총 폴드 수(예: 5 또는 10)에 대해 반복되고 평가 점수가 가장 좋은 폴드의 매개변수는 모델의 최적 매개변수로 사용됩니다.

b. 선형 SVC

- **선형 SVC** (지원 벡터 분류기)의 목적은 귀하가 제공한 데이터에 맞춰 데이터를 나누거나 분류하는 "최적 적합" 초평면을 반환하는 것입니다. 그런 다음 초평면을 얻은 후 분류기에 일부 기능을 제공하여 "예측된" 클래스가 무엇인지 확인할 수 있습니다.

c. 커널 SVM

- **SVM 알고리즘은 커널**로 정의된 수학 함수 집합을 사용합니다. **커널**의 기능은 데이터를 입력으로 받아서 필요한 형태로 변환하는 것입니다. 다른 SVM 알고리즘은 다른 유형의 커널 함수를 사용합니다. 이러한 함수는 다른 유형일 수 있습니다.

d. 의사결정 트리

- 의사결정 트리는 분류 및 회귀 트리라고도 알려진 계층적 모델입니다. 이들은 데이터로부터 응답을 예측하는 속성을 가지고 있습니다. 의사결정 트리의 속성은 노드로 매핑됩니다. 트리의 모서리는 가능한 출력 값을 나타냅니다. 트리의 각 가지는 루트에서 리프 노드까지 분류 규칙을 나타냅니다.
- 이 방법은 패턴 인식 및 머신 러닝 분야에서 예측 모델로 여러 작업에 사용되었습니다. 주요 목표는 여러 입력 변수가 주어졌을 때 다음 값을 예측하는 것입니다.

e. 랜덤 포레스트 분류기

- 랜덤 포레스트는 가지치기되지 않은 수요의 복장이거나 다양한 의사결정 트리를 가진 부트스트래핑 알고리즘처럼 내려갑니다. 각 트리는 예측 불가능하고 독립적으로 선택된 벡터의 추정치에 따라 달라집니다. 랜덤 포레스트는 단일 트리 분류기보다 믿을 수 있게 엄청난 개선을 제

f. 그래디언트 부스팅

- 그래디언트 부스팅은 재발 및 주문 문제에 대한 AI 방법으로, 일반적으로 선택 트리인 무력한 예측 모델 그룹으로 기대 모델을 생성합니다. 모든 지시 학습 알고리즘의 목표는 불행한 작업을 특성화하고 제한하는 것입니다. 그래디언트 부스팅 머신은 선택 트리 앙상블에 기반하여 여러 약한 학습자 트리를 그룹으로 혼합하여 단일 트리보다 선호하는 예측을 제공합니다. 부스팅은 타의 추종을 불허하는 정규화와 누락된 품질에 대한 더 나은 처리, 그리고 크게 향상된 효율성을 제공합니다.
- **참고사항** : "GridSearchCV"로 "GradientBoostingClassifier()"를 실행하려고 하는데, 제 시스템은 이 코드를 지원하지 않습니다.

3. 딥러닝 모델:

- 이제 원시 시계열 데이터를 기반으로 LSTM 기반 딥러닝 모델을 만들었습니다 .
- HAR은 시계열 분류 문제 중 하나입니다. 이 프로젝트에서는 다양한 머신 러닝 및 딥 러닝 모델이 최상의 최종 결과를 얻기 위해 개발되었습니다. 같은 순서로, 우리는 순환 신경망(RNN)의 LSTM(장단기 메모리) 모델을 사용하여 서 있기, 위아래로 오르기 등과 같은 인간의 다양한 활동을 인식할 수 있습니다.
- LSTM 모델 은 시퀀스 예측 문제에서 순서 의존성을 학습할 수 있는 순환 신경망의 한 유형입니다. 이 모델은 임의의 간격에 걸쳐 값을 기억하는 데 도움이 되므로 사용됩니다.
- LSTM을 다음과 같이 적용했습니다.
 - 1-LSTM 레이어
 - 더 많은 하이퍼파라미터 튜닝을 갖춘 2계층 LSTM

4. 결과 및 결론

- 아래 표에서는 모든 ML 모델 정확도 점수를 비교하고 있습니다 .

모델명	특징	하이퍼파라미터 튜닝	정확도 점수
로지스틱 회귀	Expert generated Feature	완료	95.83%
선형 SVC	Expert generated Feature	완료	96.47%
RBF SVM 분류기	Expert generated Feature	완료	96.27%
의사결정 트리	Expert generated Feature	완료	86.46%
랜덤 포레스트	Expert generated Feature	완료	92.06%

출시

게시된 릴리스가 없습니다

패키지

게시된 패키지가 없습니다.

언어

- 주피터 노트북 100.0%