Course Recommender System

with Supervised Machine Learning

Bo-Yan Huang
2024 1/16

Introduction - Background

Smartphones have become indispensable in our daily lives. We aim to develop an app capable of monitoring human activities using the inertial sensors embedded in a waist-mounted smartphone. This app will serve to track exercise levels, detect falls, and identify periods of unconsciousness.

Our goal is to classify participants' activities into six categories: walking, walking upstairs, walking downstairs, sitting, standing, and laying. To achieve this, we will utilize the Human Activity Recognition with Smartphones database. This dataset is compiled from recordings of study participants who carried smartphones with embedded inertial sensors while engaging in various activities of daily living (ADL).

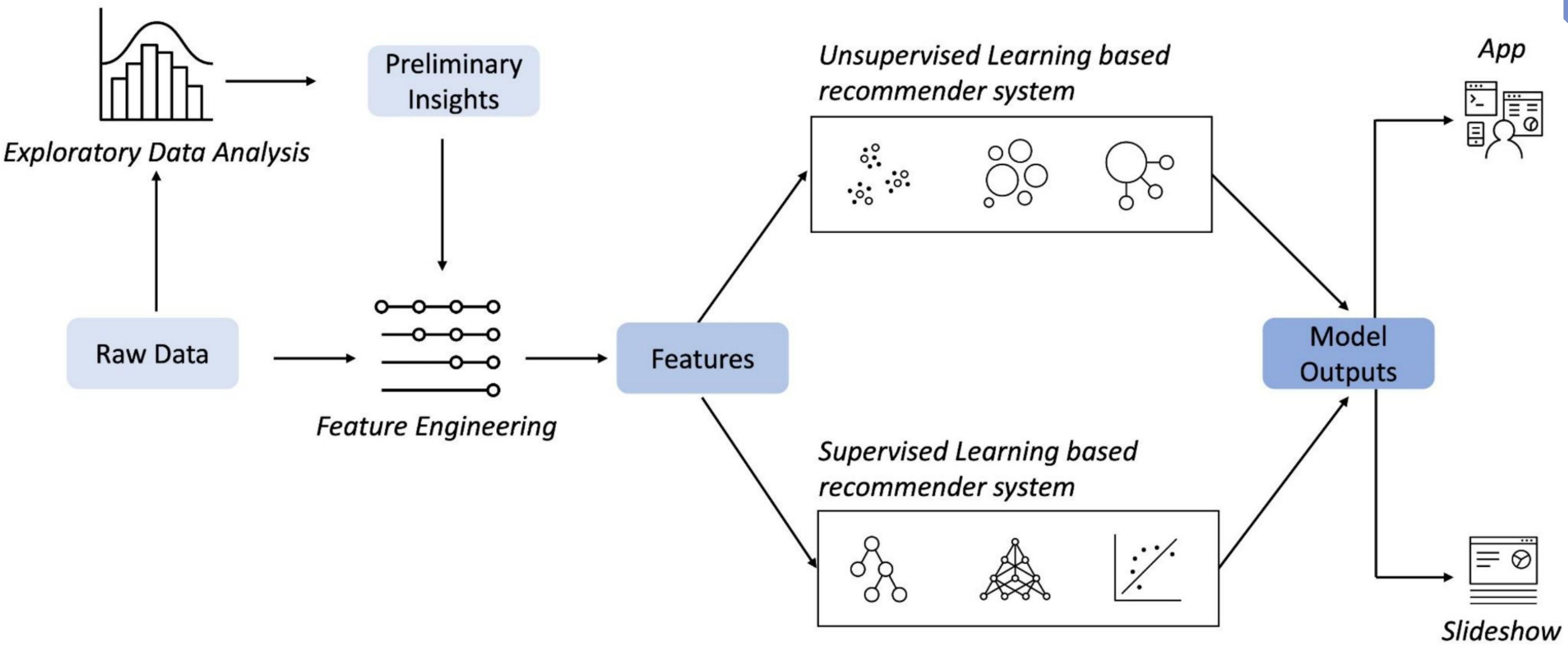
Introduction - Challenges

- **Data Quality and Variability:** The quality and variability of sensor data collected from wearable smartphones can significantly impact the model's performance. Variations in user behavior, device placement, and environmental conditions may introduce noise and affect the accuracy of motion classification.
- **Model Generalization Across Users:** Users have diverse walking patterns, body sizes, and device placements. Achieving a model that generalizes well across different users is challenging, as individual variations can lead to overfitting or underfitting issues.
- **Real-time Inference:** Real-time classification of human activities while managing power consumption on a wearable device is a critical challenge. The app needs to efficiently process sensor data, make predictions promptly, and operate within the constraints of limited battery resources.

Introduction - Transfer Learning

- **Data Quality and Variability:** The quality and variability of sensor data collected from wearable smartphones can significantly impact the model's performance. Variations in user behavior, device placement, and environmental conditions may introduce noise and affect the accuracy of motion classification.
- **Model Generalization Across Users:** Users have diverse walking patterns, body sizes, and device placements. Achieving a model that generalizes well across different users is challenging, as individual variations can lead to overfitting or underfitting issues.
- **Real-time Inference:** Real-time classification of human activities while managing power consumption on a wearable device is a critical challenge. The app needs to efficiently process sensor data, make predictions promptly, and operate within the constraints of limited battery resources.

Machine Learning Workflow



Exploratory Data Analysis

- Statistical Overview
- Key feature Identification
- Find Popular Activities
- Summary Statistics and Visualizations for the data

Columns/Features of the Data:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	...	angle(Y,gravityMean)	angle(Z,gravityMean)	Activity
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	...	0.179941	-0.058627	STANDING
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	...	0.180289	-0.054317	STANDING
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	...	0.180637	-0.049118	STANDING
3	0.279174	-0.026201	-0.123283	-0.996091	-0.983403	-0.990675	...	0.181935	-0.047663	STANDING
4	0.276629	-0.016570	-0.115362	-0.998139	-0.980817	-0.990482	...	0.185151	-0.043892	STANDING
...	...	...	...	...	...	...	...	...	...	...
10294	0.310155	-0.053391	-0.099109	-0.287866	-0.140589	-0.215088	...	0.274627	0.184784	WALKING_UPSTAIRS
10295	0.363385	-0.039214	-0.105915	-0.305388	0.028148	-0.196373	...	0.273578	0.182412	WALKING_UPSTAIRS
10296	0.349966	0.030077	-0.115788	-0.329638	-0.042143	-0.250181	...	0.274479	0.181184	WALKING_UPSTAIRS
10297	0.237594	0.018467	-0.096499	-0.323114	-0.229775	-0.207574	...	0.264782	0.187563	WALKING_UPSTAIRS
10298	0.153627	-0.018437	-0.137018	-0.330046	-0.195253	-0.164339	...	0.263936	0.188103	WALKING_UPSTAIRS

10299 rows × 562 columns

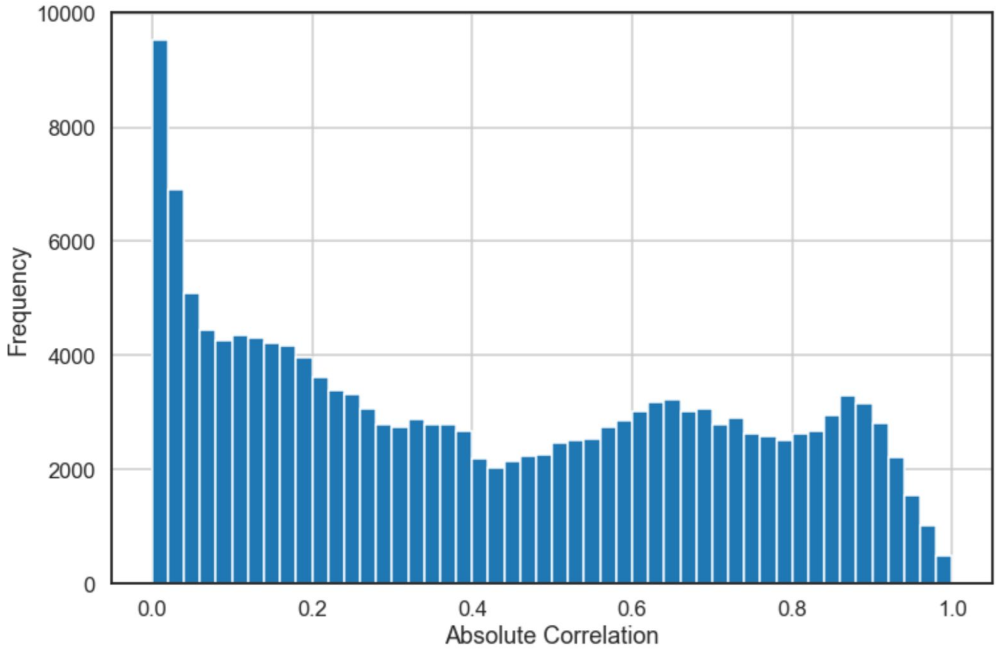
Examine the breakdown of activities

```
data.Activity.value_counts()
```

```
LAYING                1944
STANDING              1906
SITTING               1777
WALKING               1722
WALKING_UPSTAIRS      1544
WALKING_DOWNSTAIRS    1406
Name: Activity, dtype: int64
```

The activity labels are relatively balanced

Correlations



Plotting a correlation matrix is impractical due to the extensive number of features, exceeding 500. However, we can identify those that are most correlated.

Correlations

```
# The most highly correlated values
corr_values.sort_values('correlation', ascending=False).query('abs_correlation>0.8')
```

	feature1	feature2	correlation	abs_correlation
156894	fBodyBodyGyroJerkMag-mean()	fBodyBodyGyroJerkMag-sma()	1.000000	1.000000
93902	tBodyAccMag-sma()	tGravityAccMag-sma()	1.000000	1.000000
101139	tBodyAccJerkMag-mean()	tBodyAccJerkMag-sma()	1.000000	1.000000
96706	tGravityAccMag-mean()	tGravityAccMag-sma()	1.000000	1.000000
94257	tBodyAccMag-energy()	tGravityAccMag-energy()	1.000000	1.000000
...	...	...	...	...
22657	tGravityAcc-mean()-Y	angle(Y,gravityMean)	-0.993425	0.993425
39225	tGravityAcc-arCoeff()-Z,3	tGravityAcc-arCoeff()-Z,4	-0.994267	0.994267
38739	tGravityAcc-arCoeff()-Z,2	tGravityAcc-arCoeff()-Z,3	-0.994628	0.994628
23176	tGravityAcc-mean()-Z	angle(Z,gravityMean)	-0.994764	0.994764
38252	tGravityAcc-arCoeff()-Z,1	tGravityAcc-arCoeff()-Z,2	-0.995195	0.995195

22815 rows × 4 columns

Certain features exhibit identical characteristics but are denoted by different labels, leading to a correlation coefficient of 1. It is safe to exclude these redundant features.

Flowchart of clustering-based recommender system

Raw data → User/Item feature engineering → Model training → Evaluation of the models

Raw data

A 561-feature vector with time and frequency domain variables including Triaxial acceleration, the estimated body acceleration, Triaxial acceleration, Triaxial Angular velocity from the gyroscope and Activity label

User/Item feature engineering

Normalization

Handling Missing Data and Outliers

Apply PCA on feature vectors to reduce dimensions

Model training

Choose an appropriate supervised machine learning algorithm and train the model.

Evaluation of the models

Assess the model to determine if it meets our requirements. If not, revisit the previous step for further refinement.

Model Comparison

Algorithm	Logistic Regression	KNN	Gradient Boosted Trees	SVC
Interpretability	Provides coefficients that indicate the impact of features.	No explicit model, making it less interpretable.	Less interpretable due to complex ensemble structures.	Can be less interpretable, especially in high-dimensional spaces.
Handling Non-Linearity	Assumes a linear relationship between features and the log-odds.	Suitable for capturing non-linear patterns.	Can capture non-linear relationships effectively.	Can handle non-linear decision boundaries with kernel functions.
Scalability	Highly scalable.	Not very scalable, especially during inference.	Less scalable due to ensemble complexity.	Can be less scalable on large
Parameter Sensitivity	Limited hyperparameters, often robust.	Sensitive to the choice of k.	Sensitive to hyperparameters, requires tuning.	Sensitive to the choice of kernel and regularization parameters.
Multiclass	Available	Available	Available	Need to extend SVMs, may not generalize well.
Training Speed	Fast training on large datasets.	Lazy learner, slow during inference.	Slower training, especially with deep trees.	Can be slow on large datasets, especially with non-linear kernels.

Evaluation results

DummyClassifier

	precision	recall	f1-score	support
0	0.20	0.20	0.20	583
1	0.14	0.13	0.13	533
2	0.16	0.16	0.16	572
3	0.17	0.18	0.18	517
4	0.15	0.15	0.15	422
5	0.14	0.14	0.14	463
accuracy			0.16	3090
macro avg	0.16	0.16	0.16	3090
weighted avg	0.16	0.16	0.16	3090

Accuracy score: 0.16

KNeighborClassifier

	precision	recall	f1-score	support
0	1.00	1.00	1.00	583
1	0.91	0.92	0.91	533
2	0.93	0.92	0.92	572
3	0.99	1.00	0.99	517
4	1.00	0.98	0.99	422
5	0.99	1.00	0.99	463
accuracy			0.97	3090
macro avg	0.97	0.97	0.97	3090
weighted avg	0.97	0.97	0.97	3090

Accuracy score: 0.97

Evaluation results

SVC

	precision	recall	f1-score	support
0	0.19	1.00	0.32	583
1	0.00	0.00	0.00	533
2	0.00	0.00	0.00	572
3	0.00	0.00	0.00	517
4	0.00	0.00	0.00	422
5	0.00	0.00	0.00	463
accuracy			0.19	3090
macro avg	0.03	0.17	0.05	3090
weighted avg	0.04	0.19	0.06	3090

Accuracy score: 0.19

GradientBoostingClassifier

	precision	recall	f1-score	support
0	1.00	1.00	1.00	583
1	0.98	0.97	0.98	533
2	0.97	0.98	0.98	572
3	1.00	0.99	1.00	517
4	1.00	0.99	1.00	422
5	0.99	1.00	0.99	463
accuracy			0.99	3090
macro avg	0.99	0.99	0.99	3090
weighted avg	0.99	0.99	0.99	3090

Accuracy score: 0.99

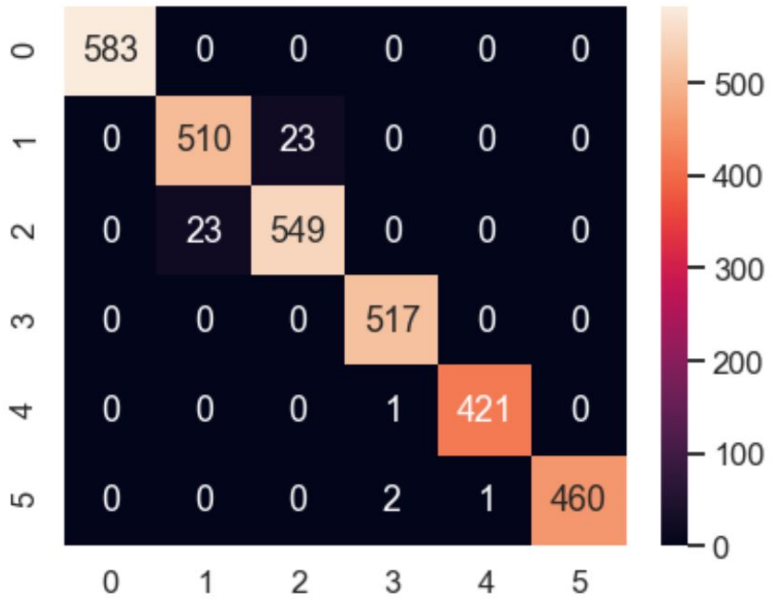
Combining models

VotingClassifier

	precision	recall	f1-score	support
0	1.00	1.00	1.00	583
1	0.97	0.97	0.97	533
2	0.97	0.98	0.97	572
3	1.00	1.00	1.00	517
4	1.00	1.00	1.00	422
5	0.99	1.00	1.00	463
accuracy			0.99	3090
macro avg	0.99	0.99	0.99	3090
weighted avg	0.99	0.99	0.99	3090

We fused logistic regression and gradient boosted trees to create a voting classifier. This new ensemble model enhanced precision and recall for activities 3 and 4, albeit resulting in a slight reduction in classification performance for activities 1 and 2.

Confusion Matrix



We fused logistic regression and gradient boosted trees to create a voting classifier. This new ensemble model enhanced precision and recall for activities 3 and 4, albeit resulting in a slight reduction in classification performance for activities 1 and 2.

Conclusions

- We can group users based on the genre in their profiles and suggest courses that are popular within the same cluster.
- Applying PCA to user profile feature vectors can decrease dimensions, consequently reducing computational power requirements
- The K-means method is preferred for our project.
- We have the flexibility to fine-tune the number of recommended courses by adjusting the popular ratio parameter.

Outlook

Possible issues:

- **Lower accuracy/recall for activity 2, 3 prediction:**
  The accuracy and recall are almost one, except for predictions related to activity 2 and 3, where a decrease in performance is observed.
- **Sensor functionality:** The collected data exclusively originates from operational sensors, and data from malfunctioning sensors has been excluded.

Outlook

Possible solutions:

- **Lower accuracy/recall for activity 2, 3 prediction:**
The data can be partitioned into two different dataset: one for instances with labels 2 and 3, and another for the remaining labels. Subsequently, we can train a dedicated model for predicting activities 2 and 3 and a separate model for the remaining activities. Combining the outputs of these specialized models is expected to yield improved results.

- **Sensor functionality:** We should introduce a new label, such as 'Not Available,' to signify data originating from malfunctioning sensors. Subsequently, we can train a model with the capability to detect whether the sensors are operational or broken.

Appendix

Data source:
https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-ML241EN-SkillsNetwork/labs/datasets/Human_Activity_Recognition_Using_Smartphones_Data.csv

Courses:
https://www.coursera.org/learn/supervised-machine-learning-classification/home

Jupyter notebook:
https://github.com/r95222023/IBM-Machine-Learning-Professional-Certificate/tree/main/Supervised%20Machine%20Learning%20-%20Classification