

BLM 4800 INTRODUCTION TO DATA MINING



COURSE PROJECT REPORT

Fatih ALTINCI

20011610

2023

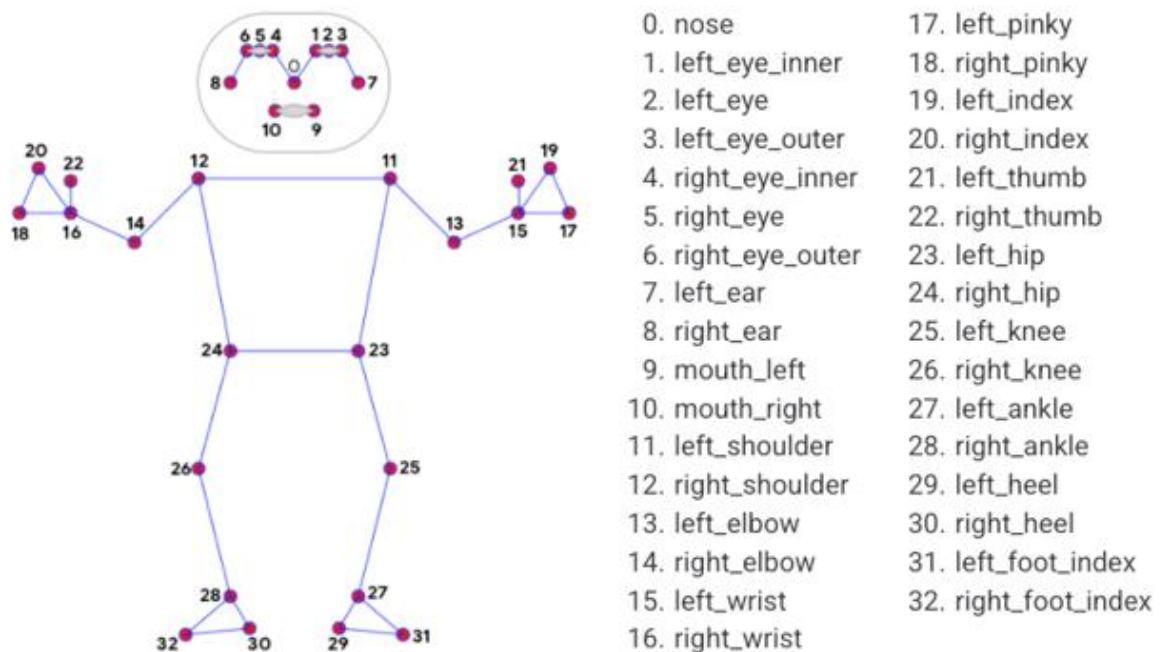
Table of Contents

Introduction.....	3
Data Analysis	3
Kaggle Competition Results.....	4
Results on Given Test Set	4
Discussion	6

Introduction

The dataset given in this project includes the movements of a person doing sports movements. The dataset was created based on the movements of people doing the exercises in about 500 videos. From every video, at least 2 frames are manually extracted. The extracted frames represent the terminal positions of the exercise. And in this project, we will estimate and classify which movements belong to the positions in the given test set.

Data Analysis



The landmark model in MediaPipe Pose predicts the location of 33 pose landmarks. Visit [Mediapipe Pose Classification](#) page for more details. Thereafter, the distances and angles between some important landmarks are calculated and represented in the dataset tables.

First, let's turn the given dataset into a dataframe with pandas and examine it.


	pose	x_nose	y_nose	z_nose	x_left_eye_inner	y_left_eye_inner	z_left_eye_inner	x_left_eye	y_left_eye	z_left_eye	...
pose_id											
0	squats_up	-0.382815	-48.231250	-54.405792	0.137189	-50.040543	-51.997875	0.502047	-50.058890	-51.986694	...
1	situp_down	54.146880	-12.822491	5.564175	56.762527	-11.221117	-0.363063	56.795986	-10.608183	-0.378148	...
2	situp_down	9.891440	-54.147266	85.344970	12.784414	-55.229970	88.534775	14.006874	-54.291880	88.543910	...
3	jumping_jacks_up	0.904673	-51.350130	-33.606970	1.338871	-53.172337	-30.013737	1.743913	-53.050697	-30.007776	...
4	jumping_jacks_down	-3.153129	-55.255062	-17.745928	-2.046205	-57.477790	-18.198952	-1.506304	-57.428230	-18.204160	...
...
1092	situp_up	-25.679585	-47.380875	-5.901453	-25.139788	-51.002510	-10.440426	-24.879524	-51.218052	-10.440801	...
1093	jumping_jacks_up	-1.185803	-51.386070	-31.526268	-0.436185	-53.642360	-28.797546	0.175695	-53.750496	-28.796741	...
1094	pullups_down	-4.307419	-49.337822	7.097422	-4.982467	-51.214745	9.683287	-5.098945	-51.260090	9.667620	...
1095	situp_down	-41.915108	-1.429882	-64.905620	-43.944553	-5.871200	-69.808044	-43.605865	-7.228406	-69.804830	...
1096	jumping_jacks_up	-0.599986	-52.802720	-33.876865	0.146288	-54.501133	-30.398746	0.676095	-54.379950	-30.394870	...

1097 rows × 100 columns

We see that the pose status and the value of each point are given in columns.

Kaggle Competition Results

This result is for my first try:

6	Fatih Altinci		0.85818	5	10d
---	---------------	---	---------	---	-----

Stratified K-Fold Cross Validation for Sampling.

LightGBM model, which is a decision tree algorithm,

Optuna for hyperparameter optimization.

Results on Given Test Set

Stratified k-fold cross-validation is the same as just k-fold cross-validation, but Stratified k-fold cross-validation, it does stratified sampling instead of random sampling.

Strafided sampling: According to the given label, it performs the division process both in the given ratio and according to the percentages of the labels.

LightGBM is a histogram-based algorithm. It reduces the computational cost by making the variables with continuous values discrete. The training time of the decision trees is directly proportional to the calculation and therefore the number of divisions.

Optuna is a software framework developed to automate the hyperparameter optimization process, using Python and automatically searching for optimum values by trial and error for best performance.

```
### final model
param = {
    "objective": "multiclass",
    "metric": "multi_logloss",
    "boosting_type": "gbdt",
    'verbosity': -1,
    "random_state": 42,
    "num_classes": 10
}
```

	precision	recall	f1-score	support
0	0.93	0.90	0.92	30
1	0.88	0.72	0.79	29
2	0.76	0.76	0.76	25
3	0.62	0.59	0.60	22
4	0.94	1.00	0.97	16
5	0.95	0.91	0.93	23
6	0.79	0.88	0.83	17
7	0.61	0.88	0.72	16
8	0.85	0.85	0.85	20
9	0.70	0.64	0.67	22
accuracy			0.80	220
macro avg	0.80	0.81	0.80	220
weighted avg	0.81	0.80	0.80	220

This result is my fifth try:

```
In [44]: score
Out[44]: 0.8047427989901914
```

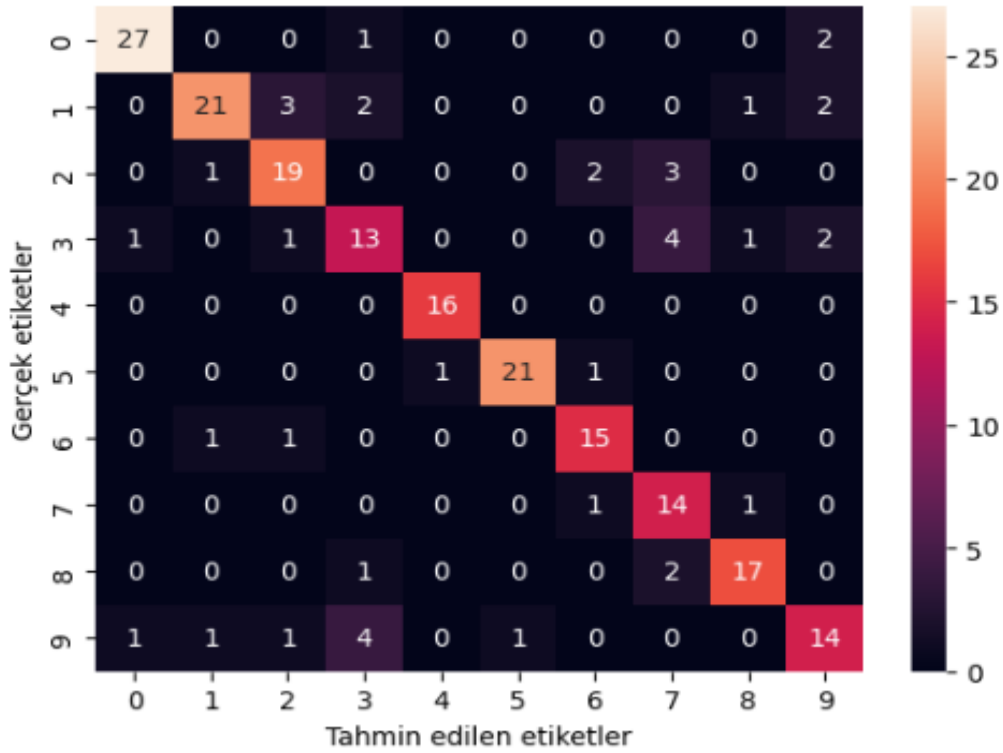
This is the result between my fifth attempt and the solution csv given:

```
In [39]: # solution.csv dosyasını oku
solution_df = pd.read_csv("solution.csv")

# my_results.csv dosyasını oku
my_results_df = pd.read_csv("besinci_deneme.csv")

# Başarı oranını hesapla
accuracy = (my_results_df["pose"] == solution_df["pose"]).mean()
print(accuracy)

0.8290909090909091
```



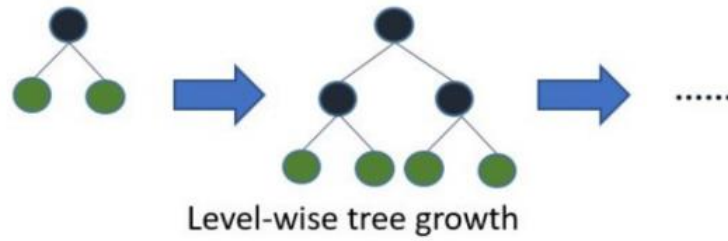
Discussion

Compared to other boosting algorithms, it has advantages such as high processing speed, large data processing, less resource (RAM) usage, high prediction rate, parallel learning and GPU learning support.

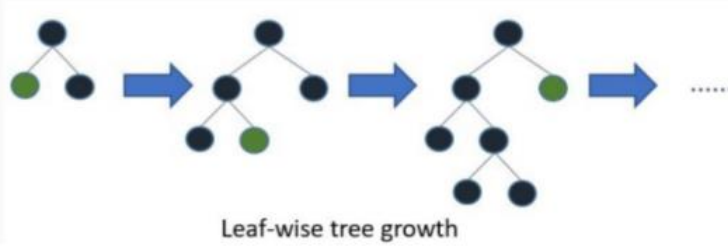
Two strategies, level-wise or depth-wise, or leaf-wise, can be used in learning decision trees. In the level-oriented strategy, the tree's balance is maintained as the tree grows. In the leaf-focused strategy, on the other hand, division from leaves, which reduces loss, continues.

Thanks to this feature, LightGBM differs from other boosting algorithms. The model has less error rate and learns faster with the leaf-oriented strategy.

XGBoost:



LightGBM:



In Gradient Boosting, the first leaf is created first. Afterwards, new trees are created by taking into account the estimation errors. This situation continues until the number of trees decided or no further improvement can be made from the model.

Classes 3 and 9 have lower accuracy (F1) than other classes.