Final_Report

October 6, 2022

ML_Assignment_1

Created by Khabibullin Salikh

s.khabibullin@innopolis.university

Maybe watched by Dji

g.dlamini@innopolis.university

gcinzoe04@gmail.com

Motivation :

In classification I must build model, which will predict class with minimum of Loss function (m

In regression I must build model, which will predict exactly target feature.

I created 2 separated .ipynb-files to these 2 tasks.

Also I must do this ML Task, that Dji will think, that Salikh is not a bad student.

Data :

In both tasks there are too many outlies.

In classification there is strong imbalance in data. So main goal is to take this moment into a

In regression given features help a little to predict a target. So I think PyTorch is more usef

Exploratory data analysis :

I calculate correlation matrix and drop some dependent features. (dropped_frames_max in classi

I also could built some categorical features instead of dropping them.

It's also important to choose right metrics, to validate and val(or test) data.

Task :

Classification :

I used metrics: "Roc_Auc", "Accuracy", "Precision", "Recall".
As I understood there are imbalance in data. Class "0" is 13.6 larger than Class "1". I chose u
Of course, I could use libraries like TSNE, but they help a little with real data.

I used LogisticRegression with different data.

Regression :

I didn't use features "bitrate_mean", "bitrate_std" because we predict bitrate))
I used metrics: "MAE", "R2", "RMSE".
I used LinearRegression, PolynomialFeatures, Lasso, Ridge, RandomForestRegressor, CatBoostRegr

Results :

In Classification task best model was LogisticRegression with class_weight = "balanced" and per
In Regression task best model was CatBoostRegressor (other models gave same results)

[12]:
```
from IPython.display import Image
Image("./Classification.jpeg")
```

[12]:

```
result
```

	Model	Class ratio	Removed outliers	Penalty	Roc_Auc	Accuracy	Precision	Recall
0	log_1	imbalanced	No	None	0.569401	0.941095	0.714514	0.142730
0	log_2	imbalanced	No	l1	0.563387	0.940500	0.707182	0.130496
0	log_2	imbalanced	No	l2	0.563387	0.940500	0.707182	0.130496
0	log_equal_1	balanced	No	None	0.709888	0.858040	0.236443	0.539824
0	log_equal_2	balanced	No	l1	0.709830	0.857986	0.236343	0.539760
0	log_equal_3	balanced	No	l2	0.709792	0.857859	0.236153	0.539824
0	log_last_1	balanced	Yes	None	0.710442	0.874674	0.262389	0.521919
0	log_last_2	balanced	Yes	l1	0.710449	0.874686	0.262414	0.521919
0	log_last_3	balanced	Yes	l2	0.710447	0.874682	0.262406	0.521919

[13]:
```
Image("./Regression.jpeg")
```

[13]:

```
# I want to know the weight of features of CatBoostModel (grad_model_2)
weights = list(grad_model_2.feature_importances_)
names = list(grad_model_2.feature_names_)

model_dict = dict(zip(names, weights))
model_dict = sorted(model_dict.items(), key=lambda x: x[1])[::-1]
```

```
model_dict
```

```
[('rtt_mean', 39.89420127579558),
 ('fps_mean', 31.695246077627022),
 ('rtt_std', 14.847658348542845),
 ('fps_std', 11.807615419516834),
 ('dropped_frames_mean', 1.7552788785177198)]
```

```
### I think you won't read to this place
### https://www.kaggle.com/salikh22
```

Conclusion :

I built different models and chose the best of them. If I know more about data I could built s