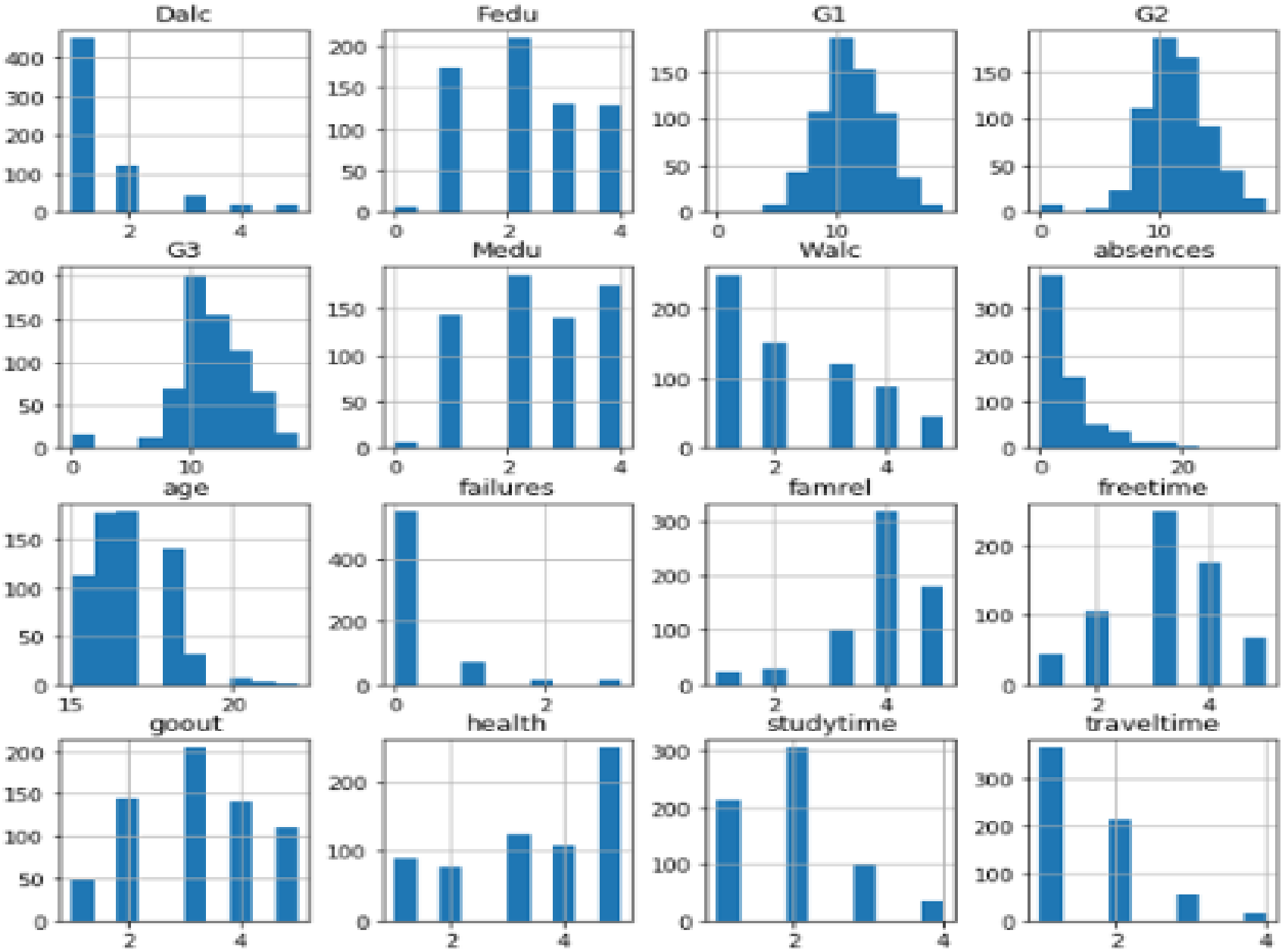
ML PROJECT

BY LUKAS OŠKELIS

- I chose a dataset which contains a lot of interesting social, gender and study information about students. There are 33 features. From these features I will try to predict final grade (G3).


Dalc
Fedu
G1
G2
G3
Medu
Walc
absences
age
failures
famrel
freetime
goout
health
studytime
traveltime

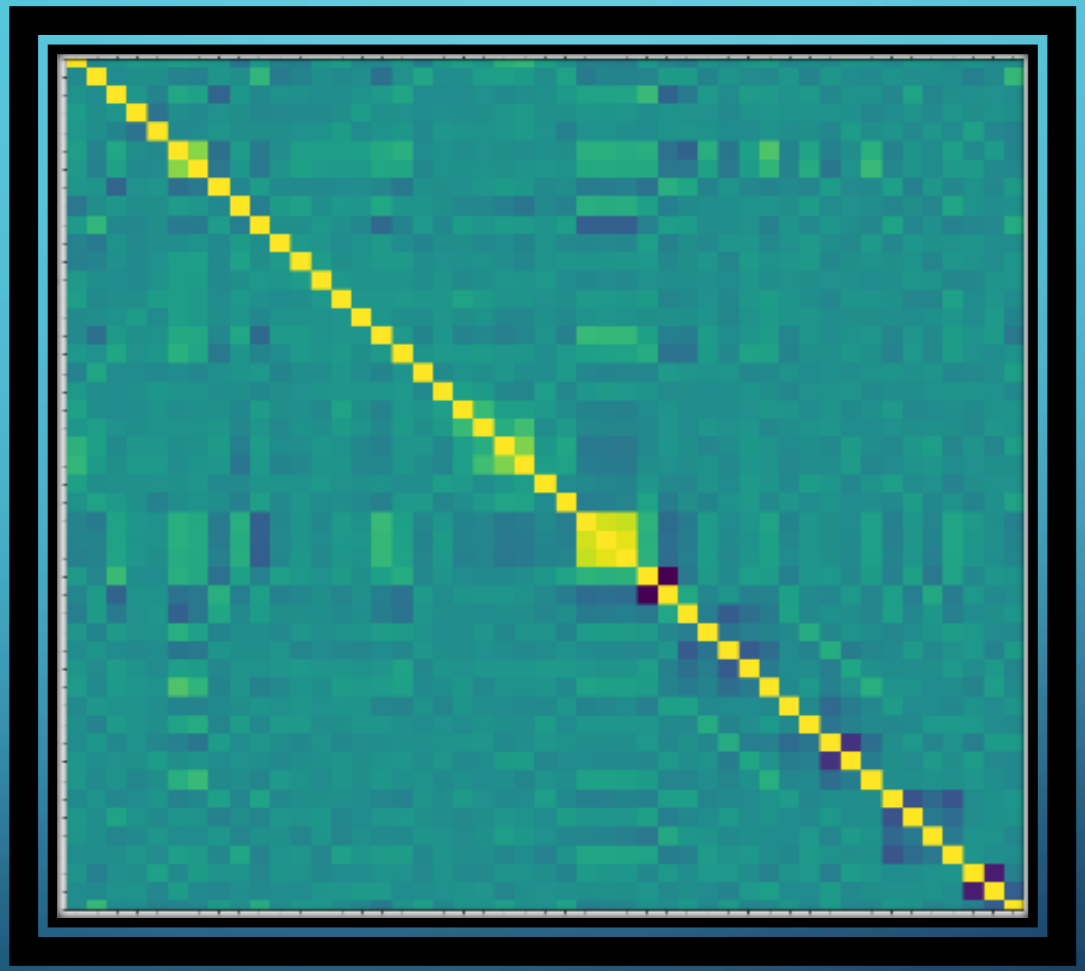
BEFORE LABEL ENCODING

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	schoolsup	famsup	paid	ac
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother	2	2	0	yes	no	no	
1	GP	F	17	U	GT3	T	1	1	at_home	other	course	father	1	2	0	no	yes	no	
2	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother	1	2	0	yes	no	no	
3	GP	F	15	U	GT3	T	4	2	health	services	home	mother	1	3	0	no	yes	no	
4	GP	F	16	U	GT3	T	3	3	other	other	home	father	1	2	0	no	yes	no	

AFTER LABEL ENCODING

	sex	age	address	famsize	Pstatus	Medu	Fedu	traveltime	studytime	failures	schoolsup	famsup	paid	activities	nursery	higher	internet	romantic	f
0	0	18	1	0	0	4	4	2	2	0	1	0	0	0	1	1	0	0	
1	0	17	1	0	1	1	1	1	2	0	0	1	0	0	0	1	1	0	
2	0	15	1	1	1	1	1	1	2	0	1	0	0	0	1	1	1	0	
3	0	15	1	0	1	4	2	1	3	0	0	1	0	1	1	1	1	1	
4	0	16	1	0	1	3	3	1	2	0	0	1	0	0	1	1	0	0	

CORRELATION MATRIX



- There is not much we can see from Correlation Matrix because there a lot of columns. But it is obvious that G1,G2 and G3 are really corelated between each other as well as weekly and daily alcohol consumptions. So, to make this project more relevant I won't use G2 and G1 in training models.

LINEAR REGRESSION

```
lr=LinearRegression()
lr.fit(stSCA[mask].drop(['G3','G2', 'G1'], axis=1),
       stSCA[mask]['G3'])

pred1 = lr.predict(stSCA[~mask].drop(['G3','G2','G1'],
                                     axis=1))

print('RMSE = {0:.04f}'.format(
    np.sqrt(np.mean((pred1-stSCA[~mask]['G3'])**2))))
```

```
RMSE = 2.4477
```

COEFFICIENTS

	coeficient
Mjob_other	-943,693,933,600.25
Mjob_at_home	-943,693,933,600.15
Mjob_services	-943,693,933,599.86
Mjob_teacher	-943,693,933,599.62
Mjob_health	-943,693,933,599.49
schoolsup	-1.48
failures	-1.38
sex	-0.50
nursery	-0.37
romantic	-0.33
paid	-0.29
freetime	-0.24
health	-0.21
Dalc	-0.17
Walc	-0.16
Medu	-0.06
absences	-0.03
goout	-0.01
traveltime	-0.00
age	0.06
activities	0.07
famsup	0.11
famrel	0.14
Fedu	0.20
Pstatus	0.30
internet	0.42
address	0.51
studytime	0.52
famsize	0.53
higher	1.69
reason_other	5,639,976,155.53
reason_course	5,639,976,155.68
reason_home	5,639,976,155.96
reason_reputation	5,639,976,156.13
school_MS	125,074,680,865.78
school_GP	125,074,680,867.25
Fjob_services	811,059,325,384.98
Fjob_health	811,059,325,385.09
Fjob_other	811,059,325,385.50
Fjob_at_home	811,059,325,385.77
Fjob_teacher	811,059,325,386.17
guardian_mother	1,478,335,634,407.10
guardian_other	1,478,335,634,407.25
guardian_father	1,478,335,634,407.78

LASSO REGRESSION AND COEFFICIENTS

- RMSE = 2.3737

From Lasso regression model we can see that things which influence final grade(G3) result positively is school in which students study, how much they study and if students want to take higher education . On the other hand, past failures also has a strong contribution for G3. In overall, Lasso gives us better RMSE and better look at important features.

	coeficient
failures	-1.41
studytime	0.52
higher	0.85
school_GP	1.14

RANDOM FORREST REGRESSOR

- RMSE =2.3988
- Training score = 0.9
- Test score =0.21
- We can see that Random Forest Regressor doesn't give us such a good result as expected and it looks like it's overfitting. Further I will try some methods to solve this...
- Using Random Forest the most important features are how many times students failed in past classes (failures) and how many absences they had. Other columns are pretty much equal (any coefficient is not bigger than 0.04)

RECURSIVE FEATURE ELIMINATION

- I will try to use one of feature selection module which is Recursive Feature Elimination. Recursive feature elimination removes the weakest feature (or features) until the specified number of features is reached. To find the optimal number of features cross-validation is used.
- What I got after RFE:
- RMSE =2.4075
- Training score = 0.91
- Test score = 0.13 . It doesn't look better…

HYPERPARAMETER TUNING

- Next thing I will try to do is hyperparamater tuning. It tries to get the most optimal hyperparamaters values(using cross-validation) to get the best result.
- RMSE=2.3207
- Training score = 1
- Test score =0.26
- So, it is useless to make hyperparameter tuning for RF because it leads to overfitting (100% accuracy on train set???).

BAGGING REGRESSOR

- I tried bagging for Random Forest Regressor.
- RMSE=2.3380
- Training score =0.68
- Test score =0.25
- We can see that bagging makes a situation better and gets us pretty good RMSE.

ENSEMBLE METHODS. VOTING REGRESSOR

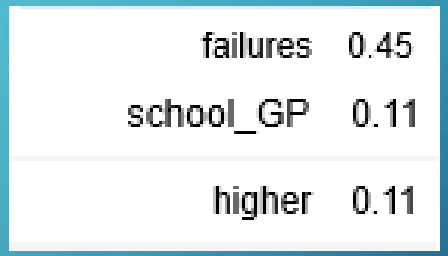
- Bagging is also Ensemble method, but I used it to solve RF overfitting problem.
- So, the next one is Voting Regressor. The idea behind the Voting Regressor is to combine conceptually different machine learning regressors and return the average predicted values.

VOTING REGRESSOR

- For Voting Regressor I used linear regression, bagging regressor and Lasso regression.
- RMSE=2.3320
- Surely, Voting Regressor gives us pretty good result. Next Ensemble method I will try is Gradient Boosting Regressor.

GRADIENT BOOSTING REGRESSOR

- RMSE = 2.3756 (nothing special)
- Here we can see that failures feature is even more important than in RF. Also, higher education and school can be seen as more important features.



GRADIENT BOOSTING REGRESSOR. FEATURE SELECTION

- Optimal number of features: 20
- RMSE = 2.3665
- After feature selection Gradient Boosting Regressor gave a little bit better result. But it is still worse than Bagging Regressor and Voting Regressor. I will also try to apply hyperparameter tuning using cross-validation.

GRADIENT BOOSTING REGRESSOR

- After hyperparameter tuning:
- RMSE = 1.7372
- Training score:0.69
- Test score:0.58
- We can see that after hyperparameter tuning Gradient Boosting Regressor is the best model by far.

Thanks for your attention