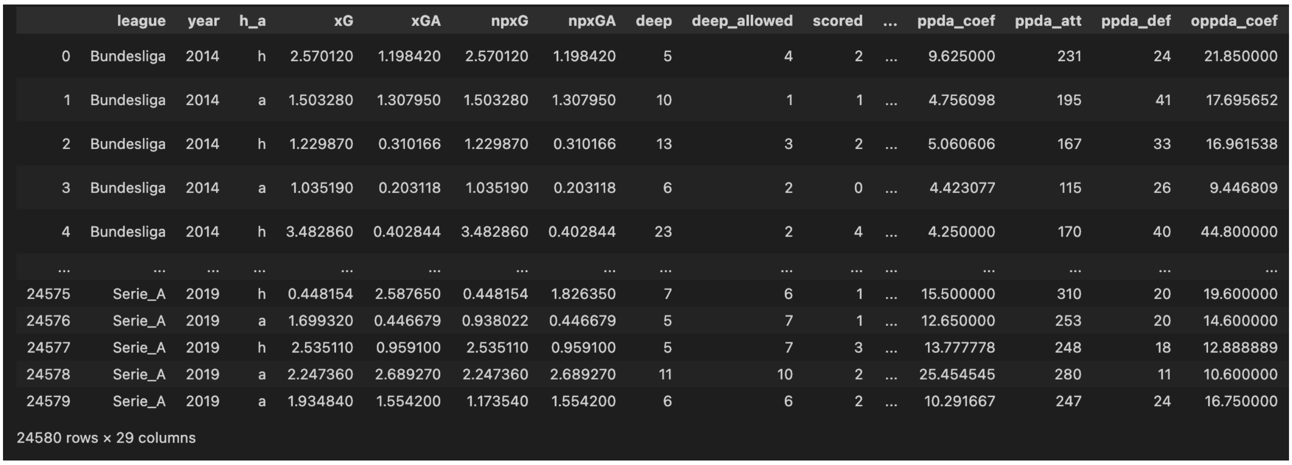
ST3189
Coursework
Student number: 210530797

Table of Contents

Part 1: Unsupervised Learning

Description

In this section the data used is a dataset from Kaggle «Football Data: Expected Goals and Other Metrics». This dataset describes advanced statistics gathered from every national championship game in top 5 leagues from 2014 to 2019. It contains 26 features, describing each game for each team.



```
            league  year  h_a        xG       xGA      npxG     npxGA  deep  deep_allowed  scored  ...  ppda_coef  ppda_att  ppda_def  oppda_coef
0       Bundesliga  2014    h  2.570120  1.198420  2.570120  1.198420     5             4       2  ...   9.625000       231        24   21.850000
1       Bundesliga  2014    a  1.503280  1.307950  1.503280  1.307950    10             1       1  ...   4.756098       195        41   17.695652
2       Bundesliga  2014    h  1.229870  0.310166  1.229870  0.310166    13             3       2  ...   5.060606       167        33   16.961538
3       Bundesliga  2014    a  1.035190  0.203118  1.035190  0.203118     6             2       0  ...   4.423077       115        26    9.446809
4       Bundesliga  2014    h  3.482860  0.402844  3.482860  0.402844    23             2       4  ...   4.250000       170        40   44.800000
...            ...   ...  ...       ...       ...       ...       ...   ...           ...     ...  ...        ...       ...       ...         ...
24575      Serie_A  2019    h  0.448154  2.587650  0.448154  1.826350     7             6       1  ...  15.500000       310        20   19.600000
24576      Serie_A  2019    a  1.699320  0.446679  0.938022  0.446679     5             7       1  ...  12.650000       253        20   14.600000
24577      Serie_A  2019    h  2.535110  0.959100  2.535110  0.959100     5             7       3  ...  13.777778       248        18   12.888889
24578      Serie_A  2019    a  2.247360  2.689270  2.247360  2.689270    11            10       2  ...  25.454545       280        11   10.600000
24579      Serie_A  2019    a  1.934840  1.554200  1.173540  1.554200     6             6       2  ...  10.291667       247        24   16.750000

24580 rows × 29 columns
```

A goal of this task is to clusterize the team performances into several types, describe and interpret each one. This will allow us to see which teams tend to perform in some specific ways, and, furthermore, we could find out the types of performances and distinctions between them.

Preprocessing

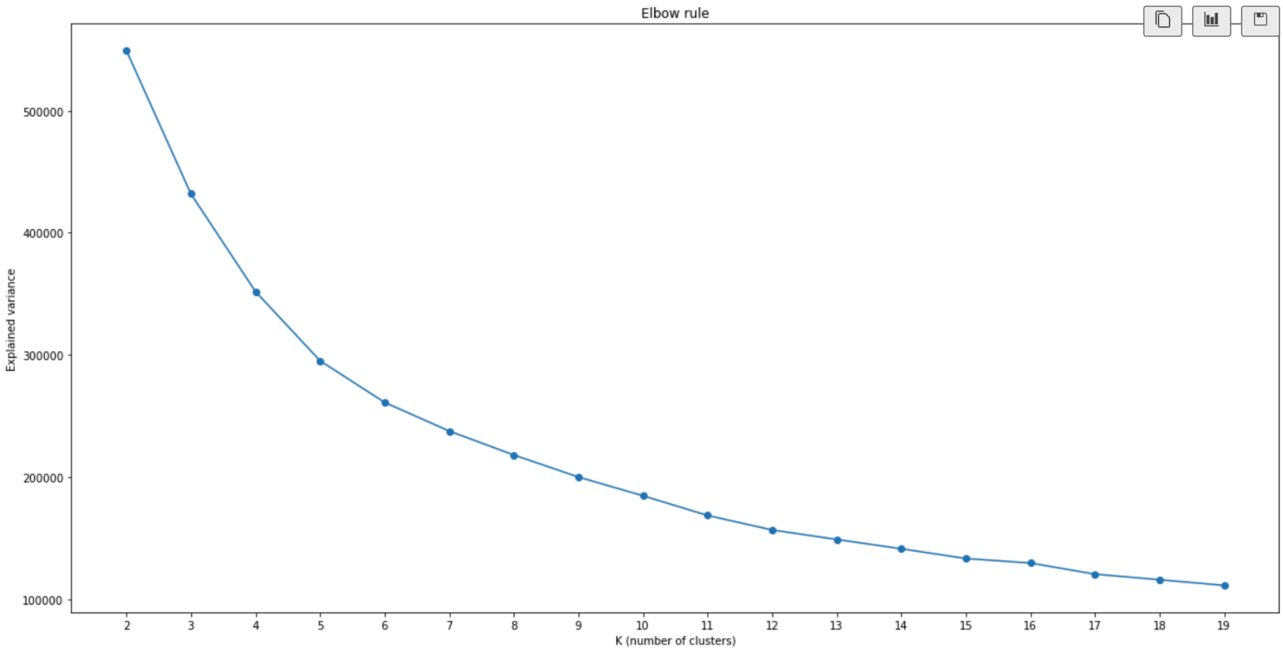
First of all, dataset contains some irrelevant to the cause matter, which contextually fits this dataset, yet does not help in terms of modeling. Features ‘result', ‘wins', ‘draws', ‘loses' were deprecated since they all can be expressed by the feature ‘points’ which tells the outcome of the game. Feature ‘date’ does not hold any value in our task since team performance is time independent, therefore it is dropped as well.

Column ‘h_a’ which shows whether the team is a home or away side was encoded into a dummy variable ‘is_h’ which takes value 1 if the team plays at home, 0 - away. This will enable us to use it in modeling. Especially, since playing at home is often considered as an advantage it also can be considered to have ordinal meaning.

Modeling

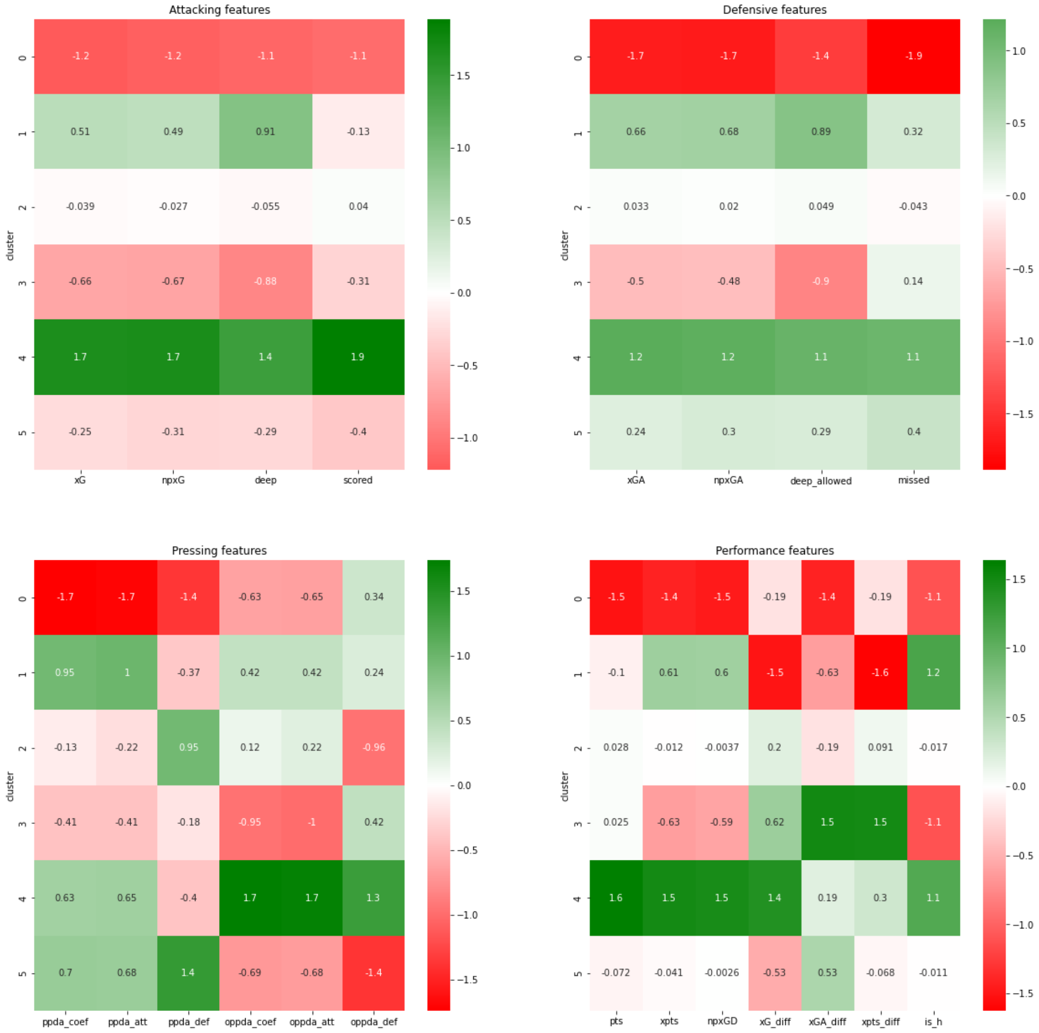
To tackle the clusterization problem a K-means approach will be used. This is an appropriate approach here, since we have to performances have a tendency to be similar with specific statistics, each type having its own mean in the cluster, which will be reflected by the cluster centers.

First off, we need to derive the optimal amount of clusters to use in in K-means which will hold the most explanatory value. The mean distance from the cluster center tends to decrease with the rise in the number of clusters, because there will be less outliers with more clusters. However, the more there are clusters the less there is explanatory value to the clusterization problem. If there are too much clusters, it would be mere impossible to tell which one represents which type of observation and the differences between the clusters becomes less observable, so the clusterization would serve no purpose. In order to avoid both issues above, we need to pick the optimal amount. This can be done with the elbow method, the idea of each is to plot the explained variance as a function of the number of clusters, so the elbow of the curve would be the optimal choice



Even though the curve her is fairly smooth, we can still see a slight bend at point 5. Therefore, we choose 5 as K and compute a 5-means model.

Deviation from norm for each cluster



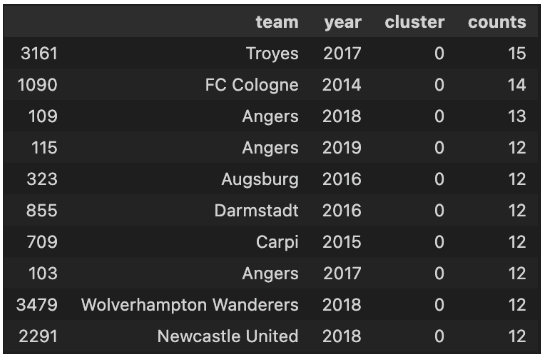
Next, the features were separated to the following ones attacking, defensive, pressing and performance in order to visualize possible distinctions between clusters. Below there are four heat maps which describe the deviation from the mean data for every respective feature to be able to tell them apart and interpret the results of the model.

Interpretation

The K-means has generated 5 clusters and with the help of the graph above the clusters were characterized as follows.

Cluster 0 describes a very poor performance by the team:

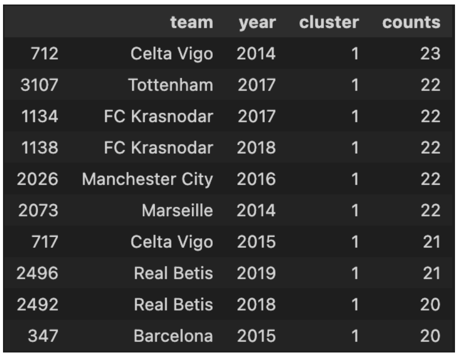
- It has the worst average statistic in terms of attacking performance, ranked last in every single attacking metric (xG, npxG, deep scored) with a significant lead.
- It shows the worst defensive performance, averaging the poorest average statistics in every defensive category (xGA, npxGA, deep_allowed, missed)
- Such team performances usually do not try to press the opponent in their half, so usually they are setting up a low block of defense and defend mostly in their own half, ranking last in ppda_def, ppda_att, ppda_coef.
- Represents the least successful performances with the worst results in terms of point accumulation, gathering the least pts, xpts, npxGD.



	team	year	cluster	counts
3161	Troyes	2017	0	15
1090	FC Cologne	2014	0	14
109	Angers	2018	0	13
115	Angers	2019	0	12
323	Augsburg	2016	0	12
855	Darmstadt	2016	0	12
709	Carpi	2015	0	12
103	Angers	2017	0	12
3479	Wolverhampton Wanderers	2018	0	12
2291	Newcastle United	2018	0	12

Cluster 1 describes a what people would call an unlucky performance by the team:

- It has good average stats all-round, except for performance stats, ranking second in most of stats in attacking, defending and pressing stats, which tells that the performance is fairly good and generally is the second best.
- Despite their good performance they are ranked second last in average points accumulated which is contrary to the above statement. However, while considering performance stats notice how extreme are the negative deviations, when it comes to xG_diff and xGA_diff, which means that the team scores less than it should have based on their performance and concedes more than it should have. In the long run xG_diff and xGA_diff should be 0, which means that the team is significantly underperforms.



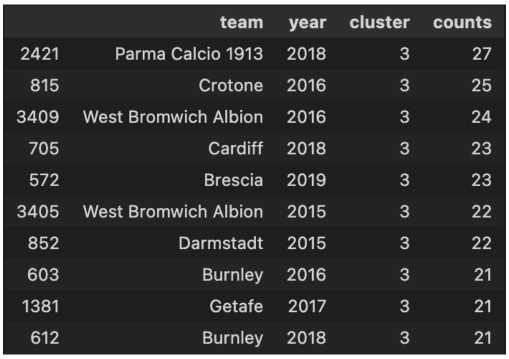
	team	year	cluster	counts
712	Celta Vigo	2014	1	23
3107	Tottenham	2017	1	22
1134	FC Krasnodar	2017	1	22
1138	FC Krasnodar	2018	1	22
2026	Manchester City	2016	1	22
2073	Marseille	2014	1	22
717	Celta Vigo	2015	1	21
2496	Real Betis	2019	1	21
2492	Real Betis	2018	1	20
347	Barcelona	2015	1	20

Cluster 2 describes a fairly average performance, probably a performance expected from a mid-table team.

- It has average stats all-around, at most of the stats the performance is ranked third or fourth
- The only 2 significant deviations are ppda_att and oppda_att

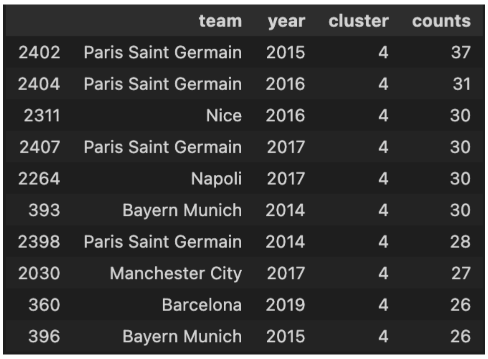
Cluster 3, describes a what people would call an undeservingly good performance by the team.

- It has below average stats all-round, except for performance stats, ranking second last in most of stats in attacking, defending and pressing stats, which tells that the performance is not that great and generally is the second last.
- Despite their below-par performance, they rank first at xpts_diff, which explains why they still accumulate 1.38 pts on average which is third best. While considering performance stats notice how extreme are the positive deviations, when it comes to xG_diff, xGA_diff and xpts, which means that the team scores more than it should have based on their performance and concedes less than it should have. In the long run xG_diff and xGA_diff should be 0, which means that the team is



	team	year	cluster	counts
2421	Parma Calcio 1913	2018	3	27
815	Crotone	2016	3	25
3409	West Bromwich Albion	2016	3	24
705	Cardiff	2018	3	23
572	Brescia	2019	3	23
3405	West Bromwich Albion	2015	3	22
852	Darmstadt	2015	3	22
603	Burnley	2016	3	21
1381	Getafe	2017	3	21
612	Burnley	2018	3	21

significantly overperformes.



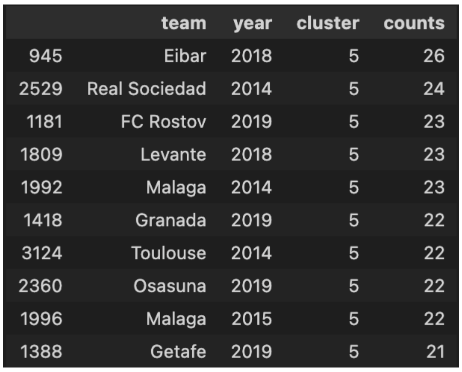
	team	year	cluster	counts
2402	Paris Saint Germain	2015	4	37
2404	Paris Saint Germain	2016	4	31
2311	Nice	2016	4	30
2407	Paris Saint Germain	2017	4	30
2264	Napoli	2017	4	30
393	Bayern Munich	2014	4	30
2398	Paris Saint Germain	2014	4	28
2030	Manchester City	2017	4	27
360	Barcelona	2019	4	26
396	Bayern Munich	2015	4	26

Cluster 4 describes a very dominant performance by the team.

- It has the best average statistic in terms of attacking performance, ranked first in every single attacking metric (xG, npxG, deep scored) with a significant lead.
- It shows the best defensive performance, averaging the best average statistics in every defensive category (xGA, npxGA, deep_allowed, missed)
- Such team performances show incredible ball posession control and ability to withstand pressure from the opponnent, ranking first by a a significan margin in oppda_coef, oppda_att, oppda_def
- Represents the most successful performances with the best results in terms of point accumulation, gathering the most pts, xpts, npxGD.

Cluster 5 describes a dynamic defensive performance

- On average they have a below average attacking stats, with great defensive stats ranking second to third in the majority of the defensive categories.
- They show great intensity in defense pressing their opponent, ranking first in defensive actions in the opposition half (ppda_def), and ranking first to seconf in ppda stats. Conversely, they show poor performance under the opponents pressure ranking fifth in every oppda coefficient which means that they are likely to have low ball posession.
- These teams are more defensive than the others, being able to grab points even with poor attacking performance, which they compensate with defensive intensity

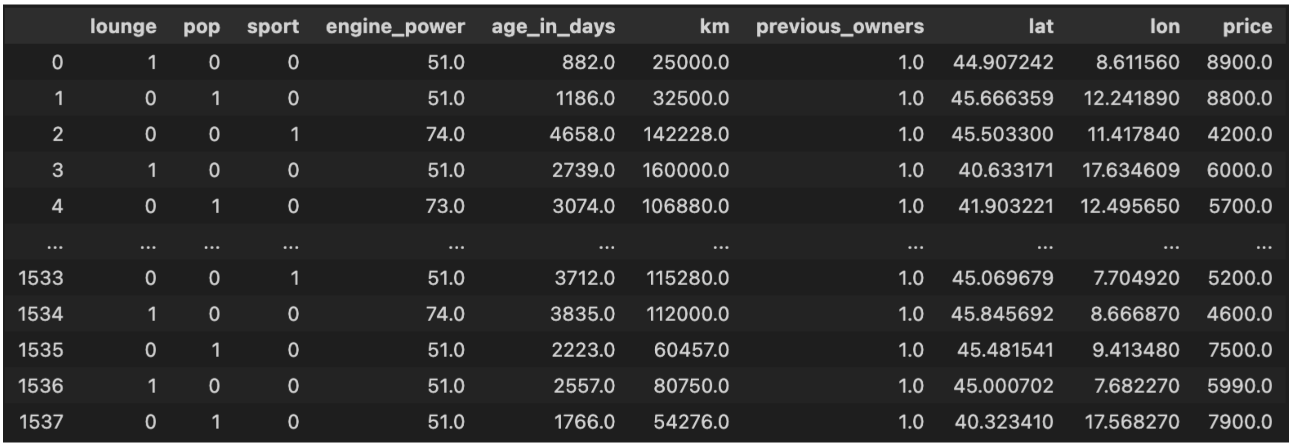


	team	year	cluster	counts
945	Eibar	2018	5	26
2529	Real Sociedad	2014	5	24
1181	FC Rostov	2019	5	23
1809	Levante	2018	5	23
1992	Malaga	2014	5	23
1418	Granada	2019	5	22
3124	Toulouse	2014	5	22
2360	Osasuna	2019	5	22
1996	Malaga	2015	5	22
1388	Getafe	2019	5	21

(The graph to the right of cluster description represent teams which are best represented by the cluster it is next to, counts - times the teams performance has been classified to the cluster during a year)

Part 2 - Regression

Description



	lounge	pop	sport	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	0	0	51.0	882.0	25000.0	1.0	44.907242	8.611560	8900.0
1	0	1	0	51.0	1186.0	32500.0	1.0	45.666359	12.241890	8800.0
2	0	0	1	74.0	4658.0	142228.0	1.0	45.503300	11.417840	4200.0
3	1	0	0	51.0	2739.0	160000.0	1.0	40.633171	17.634609	6000.0
4	0	1	0	73.0	3074.0	106880.0	1.0	41.903221	12.495650	5700.0
...	...	...	...	...	...	...	...	...	...	...
1533	0	0	1	51.0	3712.0	115280.0	1.0	45.069679	7.704920	5200.0
1534	1	0	0	74.0	3835.0	112000.0	1.0	45.845692	8.666870	4600.0
1535	0	1	0	51.0	2223.0	60457.0	1.0	45.481541	9.413480	7500.0
1536	1	0	0	51.0	2557.0	80750.0	1.0	45.000702	7.682270	5990.0
1537	0	1	0	51.0	1766.0	54276.0	1.0	40.323410	17.568270	7900.0

The data is taken from an OpenML dataset «Another-Dataset-on-used-Fiat-500-(1538-rows)». This dataset describes the prices for a car (Fiat 500), each one has different model, mileage, age and other features which describe the car as of its current condition. Dataset contains 7 features.

The task here is to based on these features to price the cars to match their market value.

Assessment metric

Since we have a regression problem here, the optimal accuracy metrics in our case for assessing performance of model would be either MAE or MSE. The differences between them is just that MSE squares the errors and MAE does not. Squaring the errors is better for punishing outliers and grading models with high number of outliers significantly worse. However in our case, we should use MAE because it is important to focus on the general accuracy and interpretability of the models.

Preprocessing

First of all, feature model which contained info about what type of model the car is: lounge, pop, sport it is. This is a categorical feature which needs to be one-hot encoded into 3 dummy variables which would tell whether the car is of that model (value is 1) or not (value is 0).

Then, another feature was added ‘km_per_day’ which is a result of dividing ‘km’ by ‘age_in_days’, which would tell us how intensively the car was in use. This was done because the original 2 features were extremely correlated (R2 = -0,86), thus to improve the model performance ‘age_in_days’ was substituted by the new feature to deal with that.

Modeling

To derive the best model, several types were chosen, so we would evaluate their performance and choose the best one. The models under consideration: Linear, Lasso, Ridge regressions, also we have decided to try Random Forest method and Gradient Boosting.

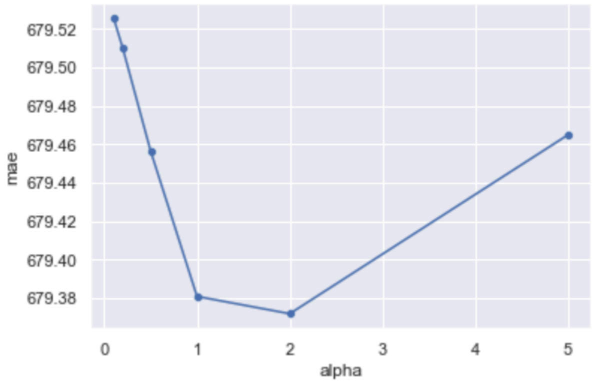
The data was split into train/validation and test sets with 70/30 ratio. To make sure there are no flukes in the assessment of model performance, each model was scored using an average score of 5-Fold Cross-validation on the train/validation set. After that, the trained model would be used on test data to asses whether the model would be able to reproduce its performance on unseen data.

Linear regression

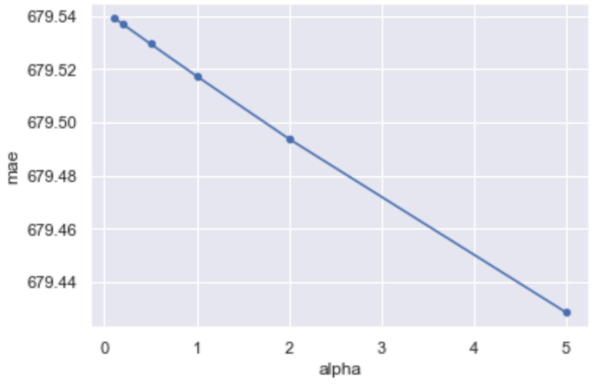
For our purposes all 3 types were considered. Linear regression is pretty straightforward and has no parameters to optimize in the model, so we will talk only about its performance only at the end while comparing models. MAE = 663.259868

Lasso regression

Lasso regression is a regression which contains the regulation parameter alpha. So let us, choose the best parameter. The graph to the right shows MAE for each alpha coefficient. We can see that alpha=2 produces the best results. TRAIN_MAE(alpha=2) = 679.3718135420079



Lasso



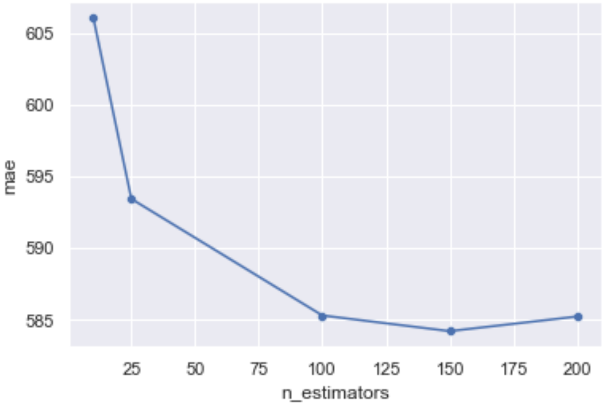
Ridge

Ridge regression

Ridge regression also uses the regularization parameter alpha and is pretty similar to Lasso, however Ridge uses the square magnitude penalty terms. The graph to the right shows MAE for each alpha. Notice that alpha=5 yields the best performance.
TRAIN_MAE(alpha=5)= 679.4285075856942

Random forest

Random forest is an ensemble method which is based on constructing a multitude of decision trees, they average deep trees with low bias and high variance, which individually tend to overfit, but used together yield good estimates. The main hyper parameter here is the number of trees (n_estimators), which greatly affects the performance, so we have to choose the best one. The best performance is generated by the model with 150 trees.
TRAIN_MAE(n_estimators=150) = 584.1986898143364



Random Forest

Gradient Boosting

Gradient boosting involves contains 2 main hyper parameters: learning_rate and n_estimators. From tuning of both, the best performance is given by learning_rate=0.1 and n_estimators=200.
MAE(learning_rate=0.1, n_estimators=200) = 589.4564190207436

regression_scores

	score
LinearRegression	663.2598679267327
Lasso	663.5623967845513
Ridge	663.387414450013
RandomForest	592.1733574359774
GradientBoosting	604.7514949690603

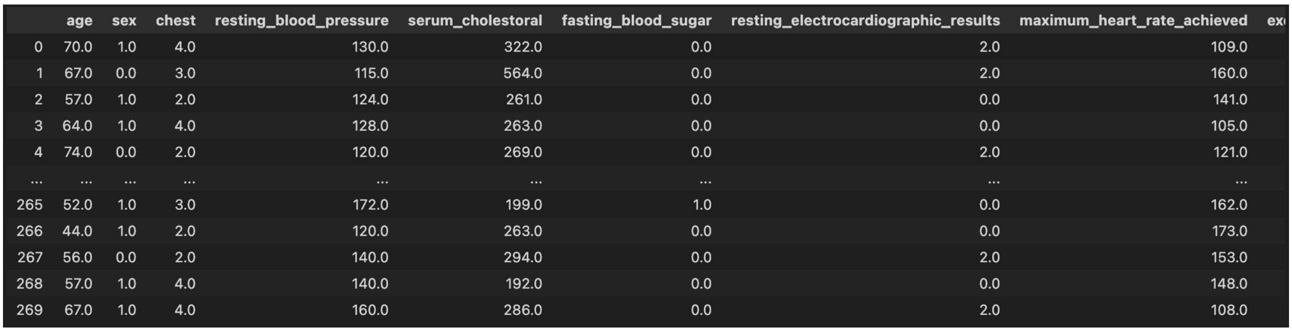
Comparison

What we can see from the final scores in the table to the left, having used trained models of each type on unseen data, is that:

- Lasso and Ridge regressions were almost useless, yielding the roughly the same performance as `Linear Regression, so regularization does not help improve performance on our dataset
- Ensemble methods significantly outperformed regression models, both models improve the score by approximately 60 which is a great result. This, however is what was expected since, these models generally outperform regression, but have less interpretability. However in our case, it is not that important as long as the model predicts the price of cars with great accuracy, which it does.
- Random Forests slightly outperform Gradient Boosting, yielding the best estimates. Therefore, the optimal model choice would be Random Forests.

Part 3 - Classification

Description

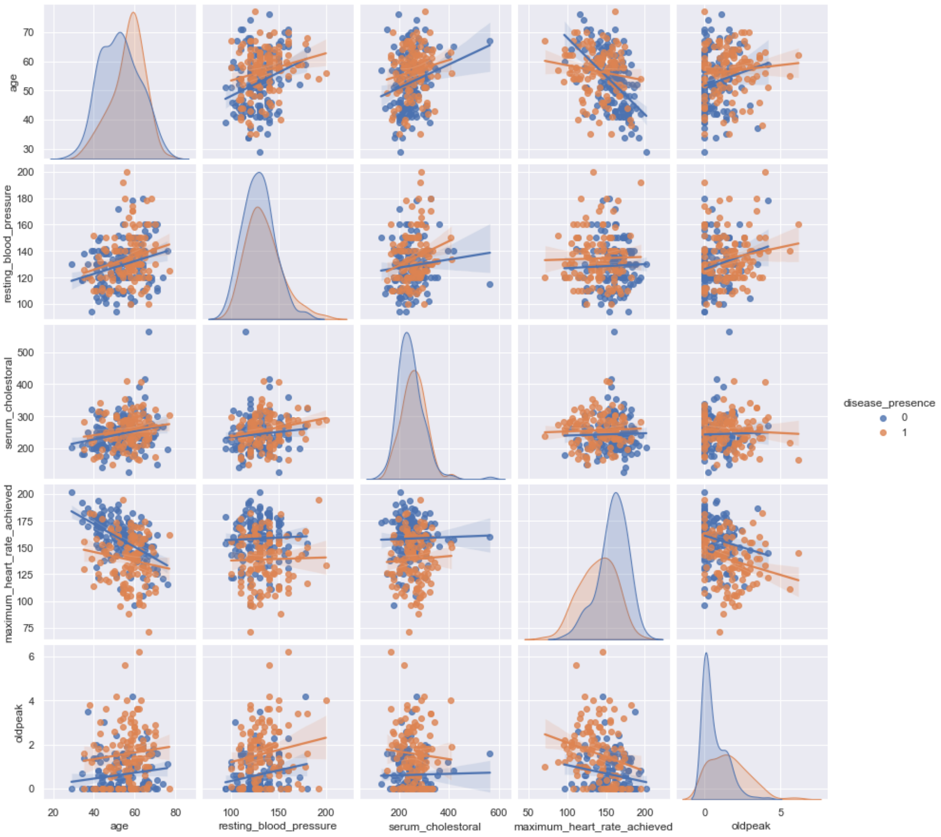


```
      age  sex  chest  resting_blood_pressure  serum_cholestoral  fasting_blood_sugar  resting_electrocardiographic_results  maximum_heart_rate_achieved  ex
  0  70.0  1.0    4.0                   130.0              322.0                  0.0                                   2.0                        109.0
  1  67.0  0.0    3.0                   115.0              564.0                  0.0                                   2.0                        160.0
  2  57.0  1.0    2.0                   124.0              261.0                  0.0                                   0.0                        141.0
  3  64.0  1.0    4.0                   128.0              263.0                  0.0                                   0.0                        105.0
  4  74.0  0.0    2.0                   120.0              269.0                  0.0                                   2.0                        121.0
...   ...  ...    ...                     ...                ...                  ...                                   ...                          ...
265  52.0  1.0    3.0                   172.0              199.0                  1.0                                   0.0                        162.0
266  44.0  1.0    2.0                   120.0              263.0                  0.0                                   0.0                        173.0
267  56.0  0.0    2.0                   140.0              294.0                  0.0                                   2.0                        153.0
268  57.0  1.0    4.0                   140.0              192.0                  0.0                                   0.0                        148.0
269  67.0  1.0    4.0                   160.0              286.0                  0.0                                   2.0                        108.0
```

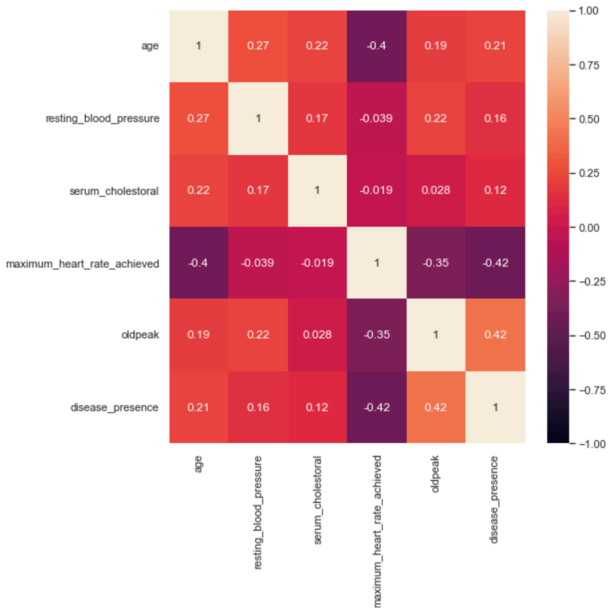
The dataset used for classification task is taken from OpenML «heart-statlog». This dataset describes the heart conditions of patients. It contains 14 features, which tell us basic information about the patient like sex, age, chest_pain_type, blood_pressure and etc. and some advanced metrics used by doctors to determine whether there is a conditions or not.

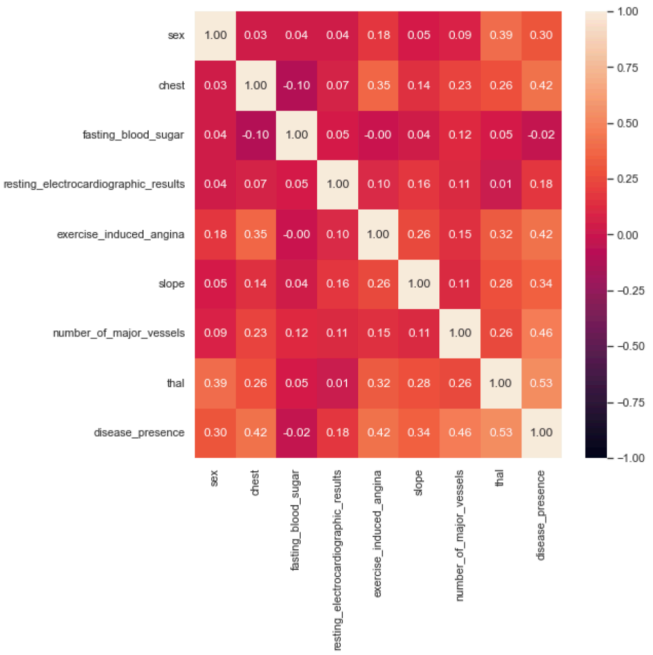
The goal of this classification task is to determine whether the patients have heart diseases or not.

EDA



We can tell the potential of the regression classification model, because observations for most features are noticeably separable by the target, which can even be seen with a naked eye, which is visible in the pairplot, which used numerical features. Consequently, each of them contributes to the problem of separating the classes.

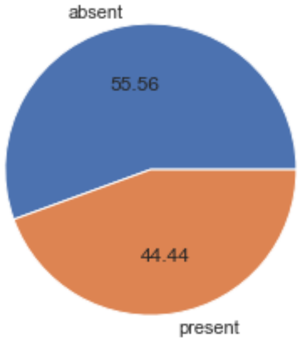




Above, we have two feature correlation heat maps, numerical to the left and categorical to the right. Upon close inspection, it is visible that feature selection is great, because most of variables have low correlations with other variables and have high correlation with the target, which means that they hold great explanatory power. This also will greatly significantly boost regression performance as the dataset avoids multicollinearity.

Assessment criteria

We have a binary classification problem, which has to predict either presence or absence of the disease, the target feature is distributed nearly evenly with a 56/44 ratio, which means we do not have to deal with confusion matrix issues and used advanced metric. In our case simple accuracy metric will provide us with great interpretability and reliability of the results.



Modeling

To derive the best model we will try 3 model types: Logistic regression, Decision Tree classifier and Gradient Boosting Classifier. This will allow us to test different distinctions of model like logit, tree and ensemble to see which methods are better.

Similarly to the previous task, the approach here is to split data into train/validations and test sets with 70/30 ratio and during training applying 5-Fold Cross-validation to achieve reliable scores.

Logistic regression

Logistic regression is a simple and reliable classification model, especially for binary tasks. Its performance in 5-fold is as follows:

	1st fold	2nd fold	3rd fold	4th fold	5th fold
Accuracy	0.94117647	1	0.9375	0.9375	1

The model gives a great performance, even predicting correctly the whole validation set 2 of 5 times

Decision Tree

Unlike regression, Decision Tree Classifier is a more complex model, which is based on producing a tree of decisions, leaves of which lead to the observation being classified. Each node contains a condition which connect the observation to further nodes based on the satisfaction of this condition, slightly subject to overfitting. So let us see its cross validation performance on validation sets.

	1st fold	2nd fold	3rd fold	4th fold	5th fold
Accuracy	1	1	1	1	1

The model gives us perfect score in every single fold of the training set. This is a strong sign of overfitting.

Gradient Boosting

Gradient Boosting was already explained in the previous part, the only distinction being that in this case we are using the classifier version of this model, so we will move straight to its performance. It is also a complex model, and given the performance of Decision Tree we might expect overfitting as well.

	1st fold	2nd fold	3rd fold	4th fold	5th fold
Accuracy	1	1	1	1	1

As previously suspected, the same happened to this model as well: each fold yields the perfect score.

Comparison

classification_scores

	score
LogisticRegression	0.9632352941176471
DecisionTree	1.0
GradienBoosting	1.0

Previously, the models were assessed only on the training sets, however, for model comparison and selection the test set is used to assess models' performances on unseen data.

From the performances of models on the test set, we can make following conclusions:

- Gradient Boosting and Decision Tree have produced the perfect score on unseen data which is a very interesting phenomena. Usually, overfitting results into poor test performance which is not the case here and may suggest that the models are actually extremely good and produce perfect predictions. Nevertheless, such extreme accuracy may be explained by the fact that overfitting issue is true, but the model performs so well on the test set due to similarity of observations across the dataset, which is most likely will not be the case if the scope of observations would increase. Furthermore, dataset contains only 270 overall observations which is low. Thus, it is highly likely that both of the models overfit
- Logistic regression gives great estimates, yielding 96% accuracy. Also, since it is a much simpler model than the other two, it is far more reliable in our case. Therefore, it is unlikely that this model is subject to overfitting. Finally, we can conclude that Logistic regression performs the best for this specific classification problem.