Coursework

Subject: Machine Learning ST3189

Student Number: 210531059

Table of contents

Credit-g dataset **3**

Unsupervised Learning 3

Classification 6

Stock dataset **9**

Regression 13

Credit-g dataset

Unsupervised Learning

Explanatory data analysis

The dataset consists of the following attributes: checking_status(Status of existing checking account, in Deutsche Mark), duration(Duration in months), credit_history(credits taken, paid back duly, delays, critical accounts), purpose(Purpose of the credit (car, television,...)), credit_amount(credit amount), savings_status(Status of savings account/bonds, in Deutsche Mark), employment(Present employment, in number of years), installment_commitment(Installment rate in percentage of disposable income), personal_status(Personal status (married, single,...) and sex), other_parties(Other debtors / guarantors), residence_since(Present residence since X years), property_magnitude(Property (e.g. real estate)), age(Age in years), other_payment_plans(Other installment plans (banks, stores)), housing(Housing (rent, own,...)), existing_credits(Number of existing credits at this bank), job, num_dependents(Number of people being liable to provide maintenance for), own_telephone(yes, no), foreign_worker(yes, no).

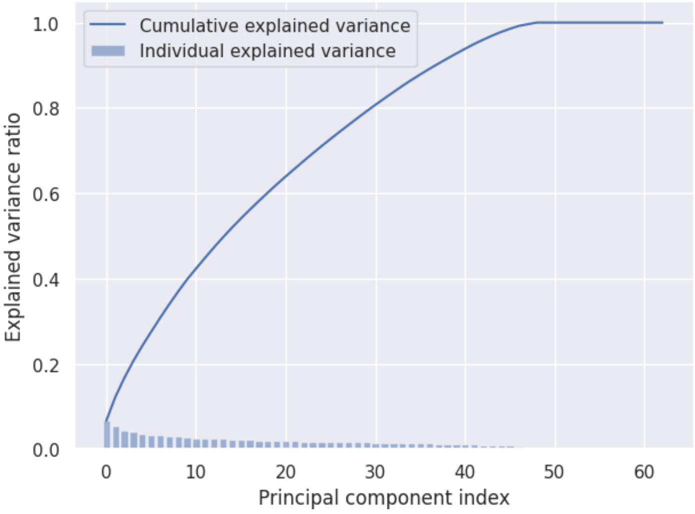
The primary target of this dataset is to classify each person as 'good' or 'bad' credit risks. We will focus on it in Classification part, but now we will try to make 'portraits' of typical bank clients.

We can see that there are plenty of categorical variables – the ones that take a certain amount of values, like foreign_worker or purpose. We will encode all these variables as dummies (replacing categorical variable having n categories with n variables that take 1 if person is described using this variable and 0 if not).

Now we are going to do the following: 1) normalize data and apply PCA, 2) use K-means algorithm to get the 'portraits' of clients.

Normalization and PCA

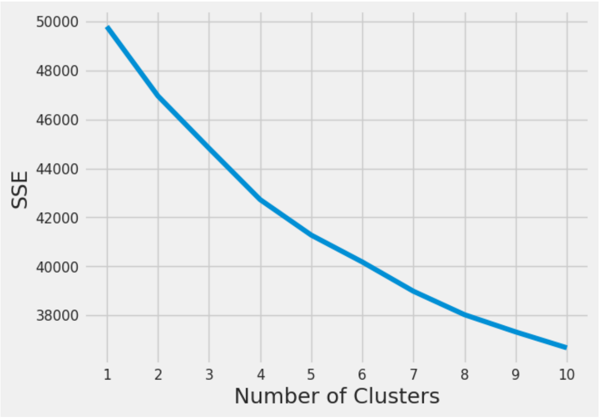
First we normalize the data and then apply PCA to reduce the number of variable and leave only the most meaningful ones. PCA makes components, maximizing variance of first ones and minimizing of the last ones. It is usually considered that



80% variance is a good threshold. As we can see on the graph, the optimal number of components is 30.

K-means clustering

K-means clustering checks the values of k nearest values, so we first need to choose optimal k. To do it, let’s plot explained sum of squares against number of clusters. To find the optimal value we will use elbow method. The points where inflection the graphed line happens are of our interest. K that we will take for consideration is 4.



Results of clustering

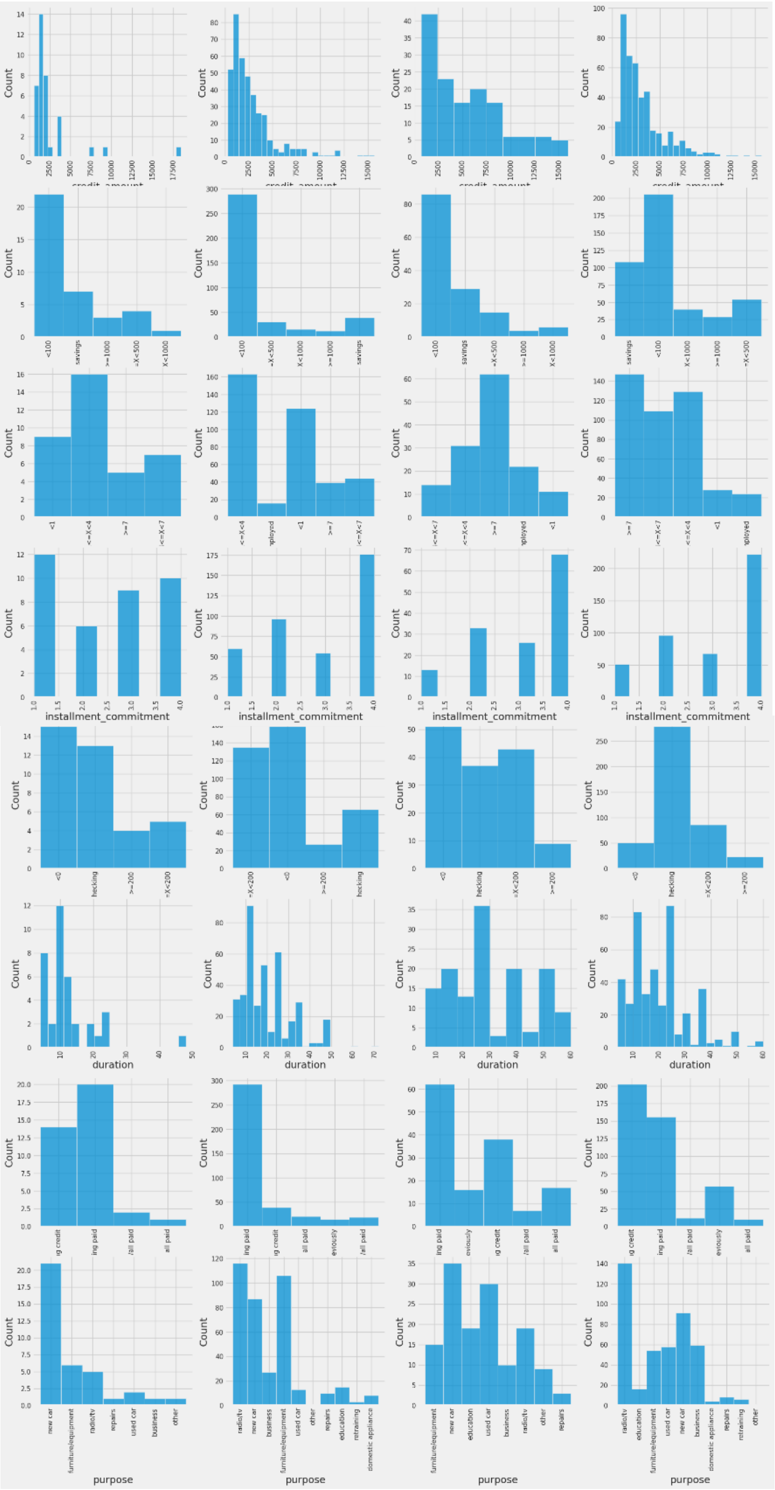
Let’s have a look at distributions of attributes of classes that we have got.

First class:

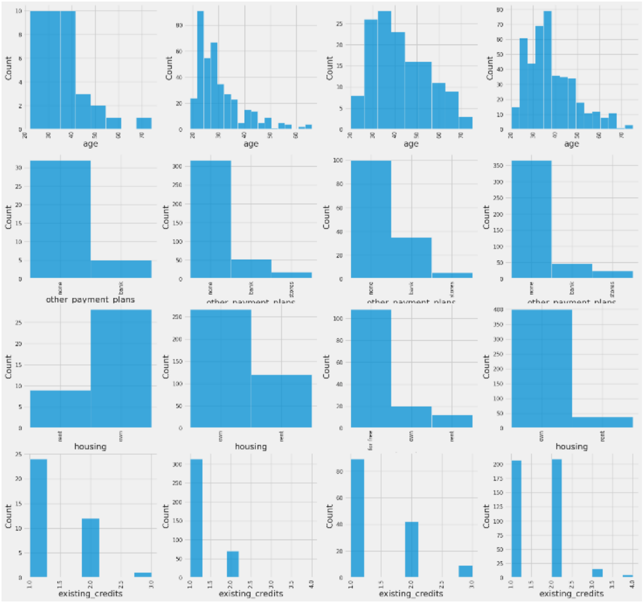
Single man of 20-40 years old. Does not use checking account or has negative balance on it. Takes credit for a new car. Quite low credit amount, low or no savings. Has been employed for less than 4 years, lives in one place for about 2 years. Skilled worker or unskilled resident, has real estate, own housing. Doesn’t have telephone and is not a foreign worker.

Second class

Woman or single male of 20-30 years old. Has under 200

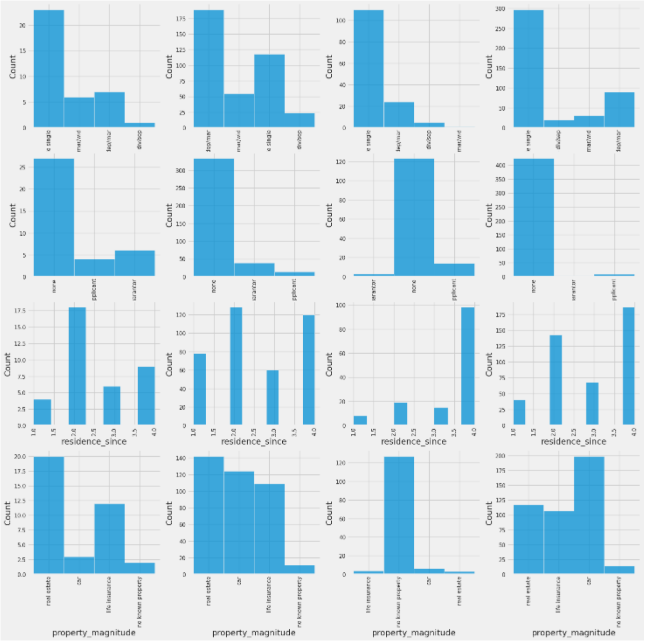


Deutsche Mark on checking account. Takes credit for radio/tv, furniture or new car. Moderately low credit amount – less than 2500. Has been employed for less than 4 years. Skilled worker or unskilled resident, has real estate/car/life insurance, own housing. Doesn't have telephone. Is a foreign worker.



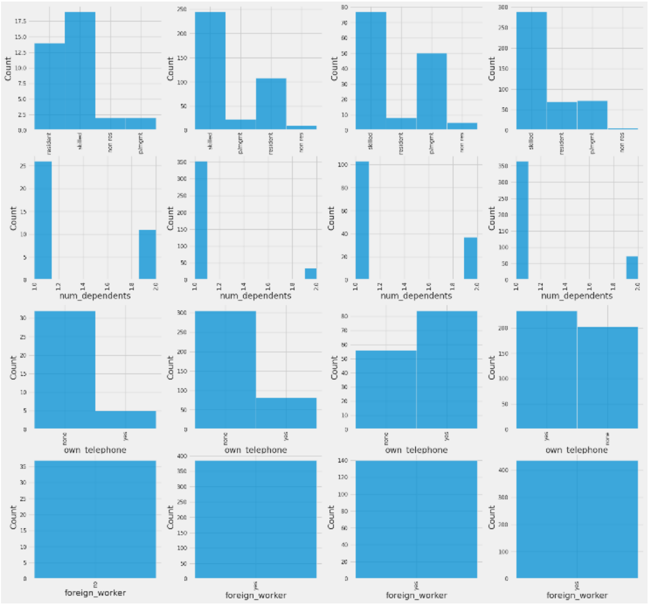
Third class

Single male of 30-60 years old. Has under 200 Deutsche Mark on checking account or does not have account at all. Takes credit for a new or used car. Credit amounts up to 8000. Low savings. Has been employed for more than 7 years, has been living in one place for 4+ years. Skilled worker or highly qualified/self-employed. Does not have any known property, free housing. Has telephone and is a foreign worker.



Fourth class:

Single man of 27-45 years old. Does not have checking account. Takes credit for radio or tv. Generally low credit amount, low savings. Has been employed for 1+ years. Skilled worker or unskilled resident. Has 1-2 existing credits. Has a car, own housing. Has a telephone and is a foreign worker.



All in all, these four groups are quite distinct, even though there are some similarities (generally clients of all classes are single males). This information may be later used for development of targeting strategies.

Classification

The same dataset as for unsupervised learning is used here. The primary goal of this dataset is to predict whether a client is a good or bad credit risk. In order to do that we should first choose the metric, then choose models to test on the data, choose best configuration of the model and then compare achieved results.

Just as for unsupervised learning we will first produce dummy variables and then normalize all variables.

Metric

Let's first consider popular metrics. *Accuracy*, which is a fraction of all values that we guessed to all values present is a good metric for general model assessment, however it does not give us any information about True Negatives (TN), True Positives (TP), False Negatives (FN) and False Positives (FP). So, we will consider metrics that do consider these values. *Recall* metric shows what part of true values has the algorithm found. It equals to $\frac{TP}{TP+FN}$. *Precision* metric shows proportion of what part of the values that the algorithm called true are true indeed. The formula is $\frac{TP}{TP+FP}$. While these metrics are great for their own purposes, they have very specific tasks, so instead *f1 metric* is used – harmonic mean of *precision* and *recall*. However, the description of the dataset says that we would better have good client classified as bad rather than bad client classified as good. For this reason, we will used *f-beta* score, with beta equal to 0.2. Beta allows to set specific relative weight to precision and recall and as a result one of metrics is favored more (in our case - precision). The formula is $\frac{(1+beta^2)*precision*recall}{beta^2*precision+recall}$.

Choosing models

For classification we will use the following models: Random Forest, Logistic Regression, Linear Discriminant Analysis and K Nearest Neighbors.

These algorithms were chosen as they represent different approaches: Random Forest as ensemble of Decision Trees, Logistic Regression computes probability of client belonging to 'good' based on linear models, QDA uses a more complex approach for estimation of probabilities than Logistic Regression (Bayes techniques) and KNN looks at values of k nearest training observations and assigns the one which appears most.

Models' optimization

We need to choose right parameters for the models in order to make them perform best. To do it we will use 10-folds cross validation technique: randomly split observations into 10 folds, when train model on 9 of them and test on the remaining one. This operation is performed 10 times (for each fold to be testing) and average metric score is returned. This is done for the following reasons: 1) single train-test split is generally not representative as we may have accidentally split the dataset so that it produces better/worse result than it would do in general; 2) we have only 1000 observations, which is usually not enough to produce good estimation of metric with single train-test split.

Among several generated models with different parameters, we will choose the one with the best average metric score among folds' test results.

Random Forest

The best model that we achieved is Random Forest with 150 trees, maximum depth of each tree of 13 and entropy splitting criterion. Entropy splitting criterion means that for deciding where the split should occur, Shannon Entropy is used.

Logistic Regression

For this model we may not always be sure whether it will converge or not, eve though we used normalization. Among the ones that have converged, the best model is the ones with no intercept assumed, l1 penalty (penalty is computed using sum of absolute values of distances) and liblinear solver – a good solver for small datasets.

QDA

QDA is pretty straightforward, although priors could be specified, but it is better to use priors obtained from train dataset if distribution is unknown.

KNN

The optimal number of neighbors to consider appeared to be 11, using 'ball tree' algorithm – costly algorithm, although efficient.

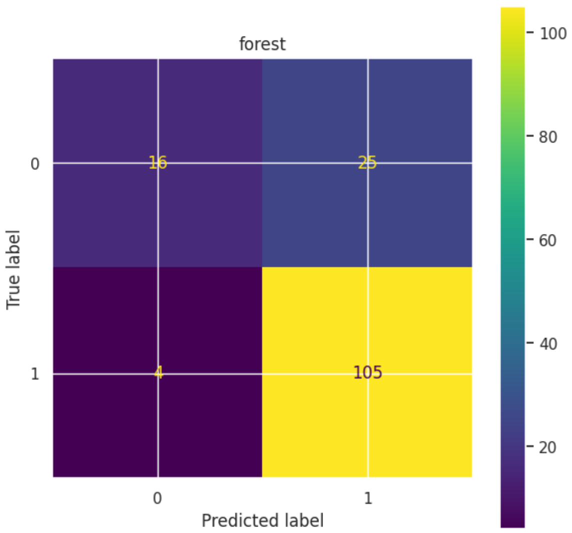
Model comparison

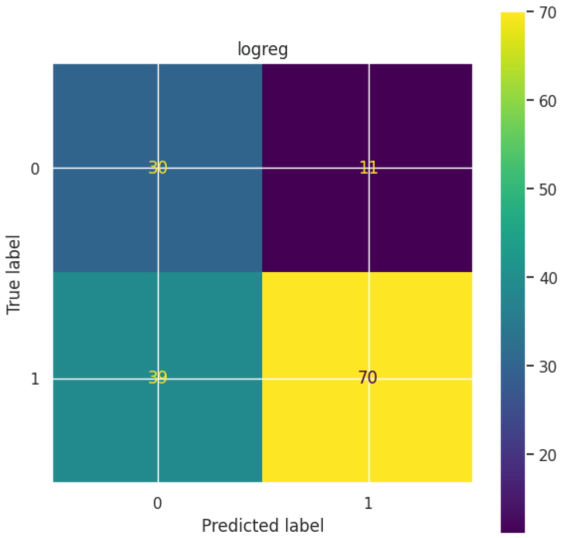
To compare models, we should find the metric of optimized models on testing set. We will use average k-fold computed metric.

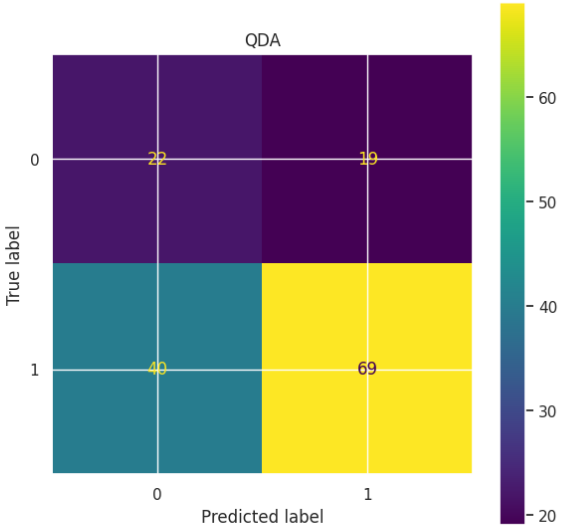
Model	Random Forest	Logistic Regression	QDA	KNN
F-beta score	0.763	0.760	0.638	0.727

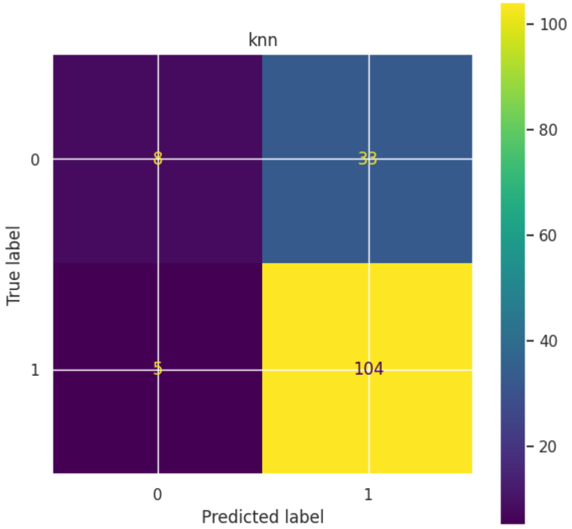
As we can see, the best performing model is Random Forest Classifier, although it is the costliest model. For a bigger dataset we may have got a different picture, as QDA seems to be lacking enough data in order to simulate priors. However, none of the models reaches 0.8. This may happen due to unbalance of targets, paired with high irreducible error, which can only be reduced by improving dataset.

Now let’s have a look how these models behave on a random split, using confusion matrices.









As we can see from the picture, QDA generally makes a lot of mistakes and that is why it produced relatively low score. Forest makes lowest number of mistakes, but unfortunately, it makes a lot of unfavored mistakes (predicting 1 for true 0). Logistic Regression makes a lot of mistakes, however it does not make many unfavored mistakes, so its score higher than it would have been with equal weights for precision and recall. Comparing forest and knn, we can see that forest’s performance is generally superior to performance of knn.

So, we actually have a choice between logistic regression and Random forest. This choice depends completely on for what extant we favor False Negatives than False Positives.

Stock dataset

Regression

For this problem a new dataset was chosen. Based on stock prices of 9 aerospace companies we need to find stock price of the 10th aerospace company. The data consists of 9 features – stock prices of 9 aerospace companies and 1 target variable – stock price of the target company.

All features are purely numerical here, so there is no need to encode categorical features.

In order to try to produce better predictions we will also try to add second degrees of features (X^2) and features’ cross products. To test efficiency of such additions we will test models both with these additions and without.

We will again use 10-fold cross validation for the same reasons as for classification (this dataset is also small – 950 observations).

As the main metric we will use MSE – mean squared error, which is the mean of squared differences between true and predicted value. It enables us to understand, which model makes more significant mistakes on average.

For regression task we will use the following models: Linear Regression, Random Forest regression, Lasso and Ridge regressions. Linear regression tries to draw a line, that would explain the data best, while Ridge and Lasso are extensions that try to add penalty for model complexity and thus, make it simpler, escaping overfitting. Random Forest Regression uses trees in order to make best prediction.

Models’ optimization

Linear regression

Linear regression is pretty simple model, so there is no actual need to make some adjustments there.

Random Forest – without second degree features

Best model appears to be with 100 trees, maximum depth of each tree equal to 13.

Random Forest – with second degree features

Best model appears to be with 200 trees, maximum depth of each tree equal to 15.

Lasso regression - without second degree features

Lasso regression optimization returned the model with alpha = 2 (penalty weight) and random selection meaning a random coefficient is updated every iteration rather than looping over features sequentially by default. Intercept in this case is not fitted.

Lasso regression - with second degree features

This optimization returned the model with alpha = 2 and random selection. However, compared to case without features, we fit intercept here.

For Ridge Regression models we will first normalize data as in this case better predictions are usually produced.

Ridge regression - without second degree features

Ridge regression optimization returned the model with alpha = 10, fitted intercept and lsqr solver, meaning it uses the dedicated regularized least-squares routine. It is the fastest and is an iterative procedure.

Ridge regression - second degree features

This optimization returned the model with alpha = 2 and sparse_cg solver. It uses the conjugate gradient solver and is an iterative algorithm.

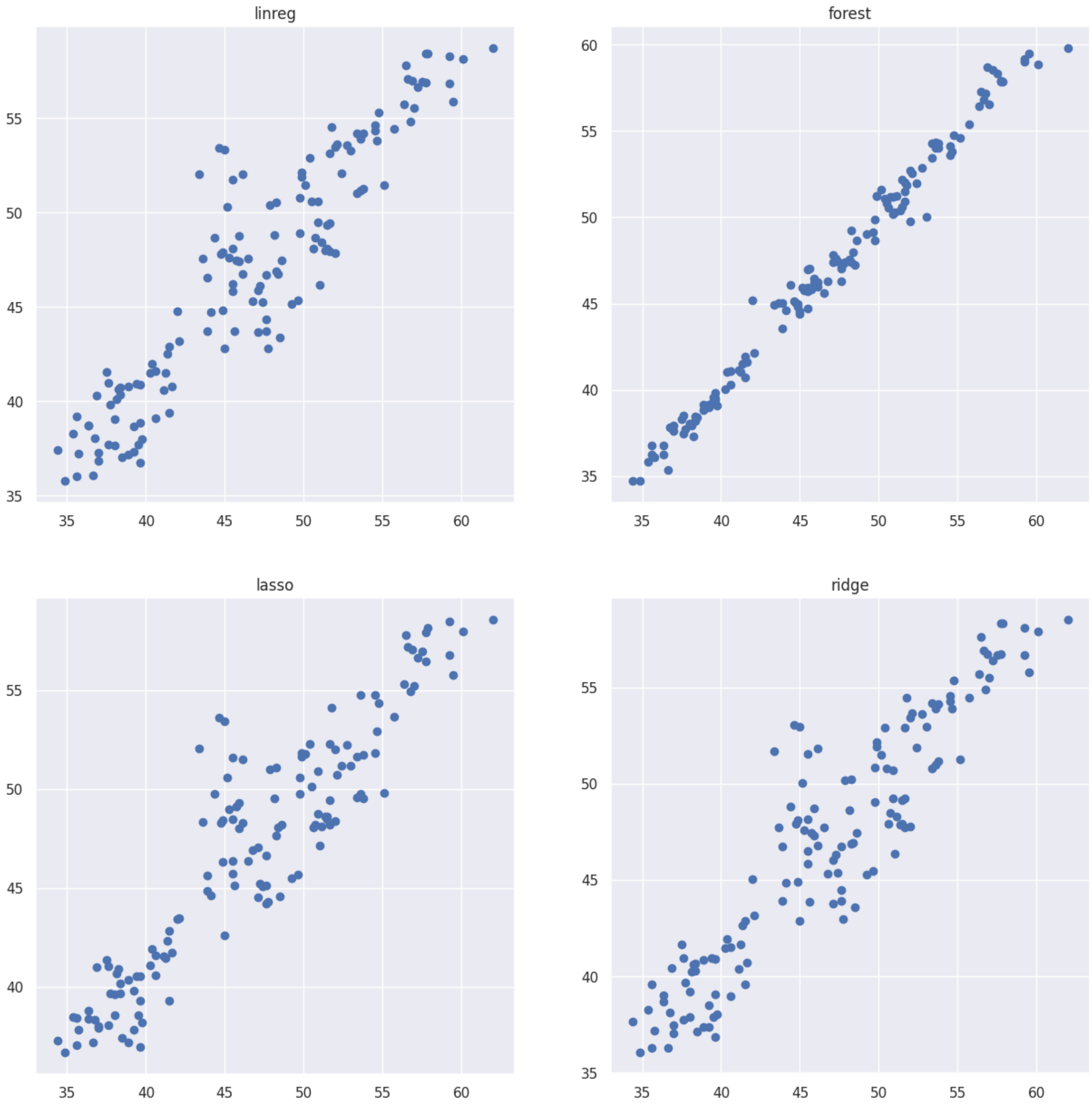
Comparison of models

	Random forest	Linear Regression	Lasso	Ridge (on normalized)
Average MSE for no second degree	8.546	15.453	12.062	15.580

Average MSE for with second degree	9.946	19.067	8.729	5.755

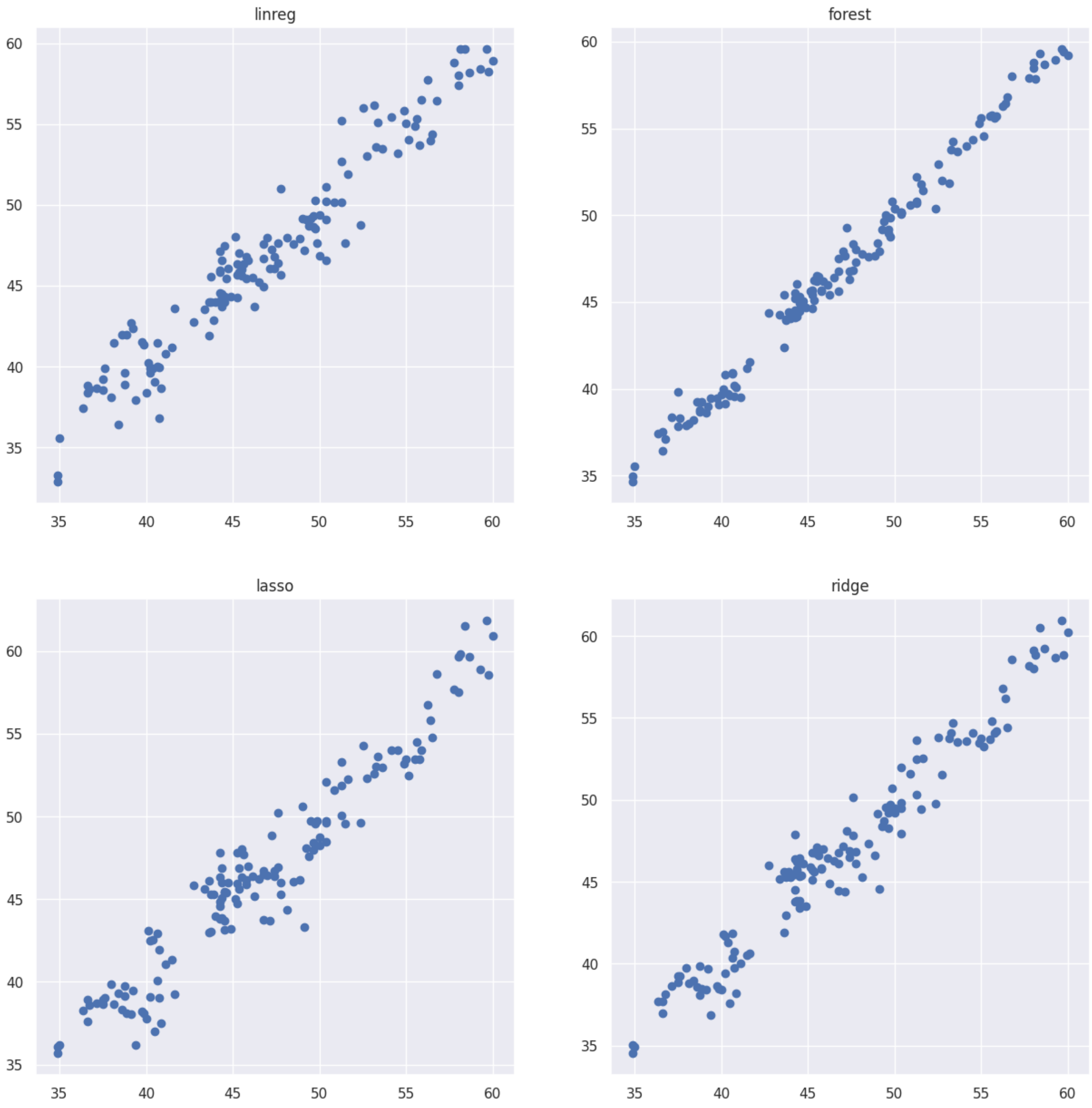
As we can see, in general, addition of second-degree terms benefits prediction quality for Lasso and Ridge and reduces quality for Linear Regression and Random Forest. The lowest value here is produced by Ridge with second-degree features giving average MSE of 5.755.

Now let’s have a look how these models behave on a random split. We will randomly take observations for train and test parts of dataset and then plot results to actually try to see the difference. We will plot predicted values against true values. The closer the points are to line y = x, the better the model.



First, let’s look at models fitted without second-degree variables. We can clearly see that forest is a better model here, which completely corresponds with the results obtained using optimization.

Now, let’s look what if we add degree 2 variables.



Here we can see an interesting picture. While on average Ridge performs better, for this particular case we have Forest being a better model. This proves that sometimes multiple models should be used in order to obtain the best results.

References

Stock dataset - https://www.openml.org/search?type=data&sort=runs&status=active&qualities.NumberOfClasses=lte_1&qualities.NumberOfFeatures=between_10_100&id=223

Credit-g dataset - https://www.openml.org/search?type=data&status=active&id=31

Sklearn library - https://scikit-learn.org