Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University

Zhangir Rayev

Development of mathematical models of system for determining the optimal objects with multi-criterial parameters

THESIS

Presented in Partial Fulfillment for the
Degree of Master of Science in Mathematics
(degree code: 6M060100)
Department of Mathematics and Natural Sciences Faculty of
Engineering and Natural Sciences

Supervisor: **Aziza Aipenova**

Kaskelen, 2019

Abstract

This thesis is based on the construction of a mathematical model for determining unknown parameters using multivariate regression analysis. Structured data are given for the derivation and elimination of significant factors and coefficients. Also, machine learning simple regression models are used for modelling. The results have been evaluated and shown for comparative purposes.

Аңдатпа

Бұл тезис көп өлшемді регрессиялық талдау көмегімен белгісіз параметрлерді анықтаудың математикалық моделін құруға негізделген. Маңызды факторлар мен коэффициенттерді шығару үшін құрылымдалған деректер келтірілген. Сонымен қатар, модельдеу үшін машиналық оқытудың қарапайым регрессиялық модельдері қолданылады. Нәтижелер бағаланып, салыстырмалы мақсаттар үшін көрсетілді.

Аннотация

Данный тезис основан на построении математической модели определения неизвестных параметров с помощью многомерного регрессионного анализа. Приведены структурированные данные для вывода и исключения значимых факторов и коэффициентов. Кроме того, для моделирования используются простые регрессионные модели машинного обучения. Результаты были оценены и показаны для сравнительных целей.

Acknowledgements

I express my deep gratitude for the help in the preparation of the thesis work for my supervisor - Aziza Aipenova. For her serious scientific approach and strict mastery of science deeply infected and encouraged me. From choosing a topic to the final completion of the work, she always gave me empathetic guidance and tireless support. She helped me a lot in my work, not only helped me correct mistakes in my work, but also gave me valuable advice. I would like to take this opportunity to express my heartfelt thanks and deepest respect.

I thank my comrades and friends for carefully reading this work, expressing their opinions and commenting on my materials.

To all of them and to many others - my sincere appreciation and gratitude.

Contents

1 Introduction 6

2 Multivariate regression 9

2.1 Regression tree model's type 19

3 Correlation analysis 29

3.1 Kendall's correlation coefficient 35

3.2 Spearman's rank correlation coefficient 36

4 Modelling with tree based ensemble models 37

4.1 Random forest 38

4.2 Adaboost 40

5 Results 51

5.1 Random forest 52

5.2 Adaboost 53

6 Conclusion 56

References 57

1. Introduction

The considered structural data are taken from Kazakhstani sources and not all parameters have been fully described. During analysis process, implicit data were identified and replaced with average values.

	names	year	volume	fuel_type	transmission	body	drive	mileage	wheel	color	city	customs clearance	price
0	Toyota Camry	2018	2.5	бензин	автомат	седан	передний привод	1.0	слева	серебристый металлик	Алматы	Да	13300000
1	Mercedes-Benz G 63 AMG	2014	5.5	бензин	автомат	внедорожник	NaN	10300.0	слева	черный	Алматы	Да	68000000
2	Toyota Corolla	2013	1.6	бензин	автомат	седан	передний привод	39258.0	слева	серый металлик	Алматы	Да	5499999
3	Mercedes-Benz E 220	1993	2.2	газ-бензин	автомат	седан	задний привод	NaN	слева	серый металлик	Шымкент	Да	1850000
4	Mitsubishi Outlander	2013	2.4	бензин	автомат	кроссовер	полный привод	60000.0	слева	серый	Актобе	Да	8100000

Figure 1.1: Cars dataframe

The first five rows of dataframe has been shown in Figure 1.1.

```
RangeIndex: 4997 entries, 0 to 4996
Data columns (total 13 columns):
names                4997 non-null object
year                 4997 non-null int64
volume               4995 non-null float64
fuel_type            4995 non-null object
transmission         4997 non-null object
body                 4997 non-null object
drive                4405 non-null object
mileage              3276 non-null float64
wheel                4952 non-null object
color                4596 non-null object
city                 4997 non-null object
customs clearance    4997 non-null object
price                4997 non-null int64
dtypes: float64(2), int64(2), object(9)
```

Figure 1.2: Dataframe information

General information about dataframe content(Figure 1.2) is quite clear and enough to model and analyze.

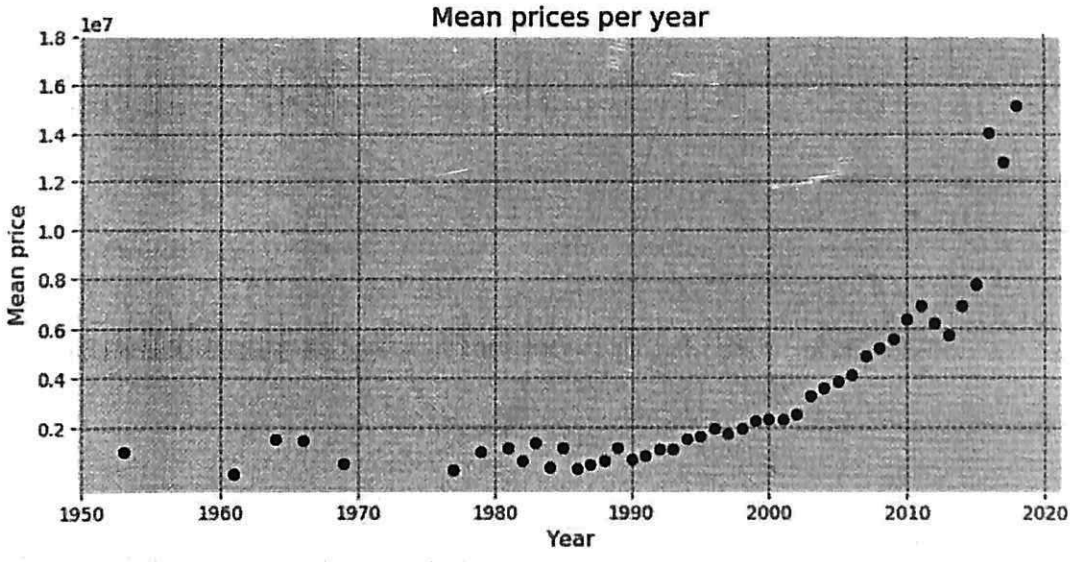


Figure 1.3: Mean prices

Mean prices of cars after 2016 has been looked overwhelming (Figure 1.3), but it has been up to individual production models of cars as 'Rolls Royce' or so on. Those data has not been counted as outliers.

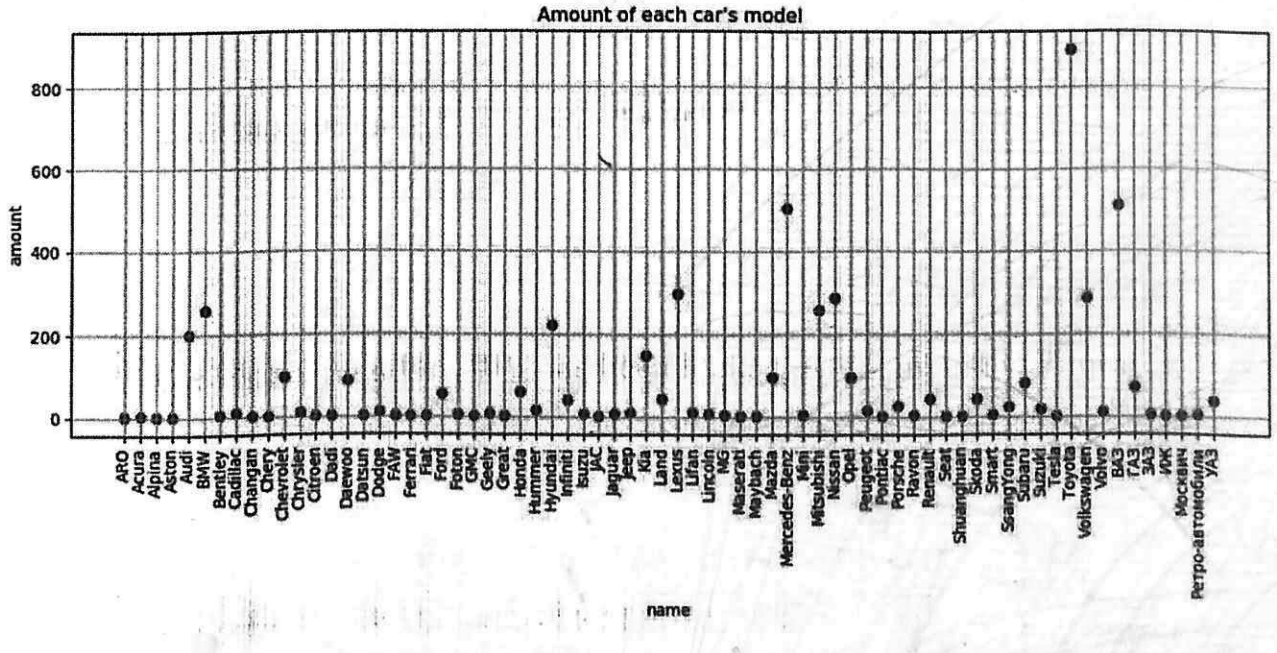


Figure 1.4: Name per amount

The minimum point of amount was 1 and it has been grown in certain models of cars which are popular and high sold in Kazakhstan (Figure 1.4 and Figure 1.5).

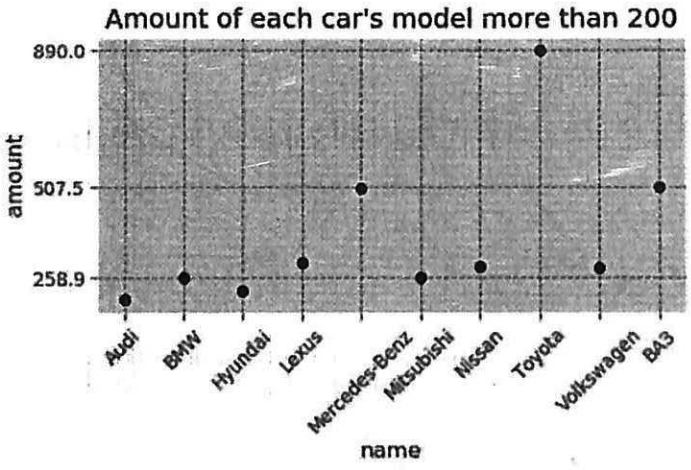


Figure 1.5: More than 200

The total number of unique city is 137 and there were a lot of unimportant suburbs and villages. Figure 1.6 has shown the important places in general which amount is more than 30.

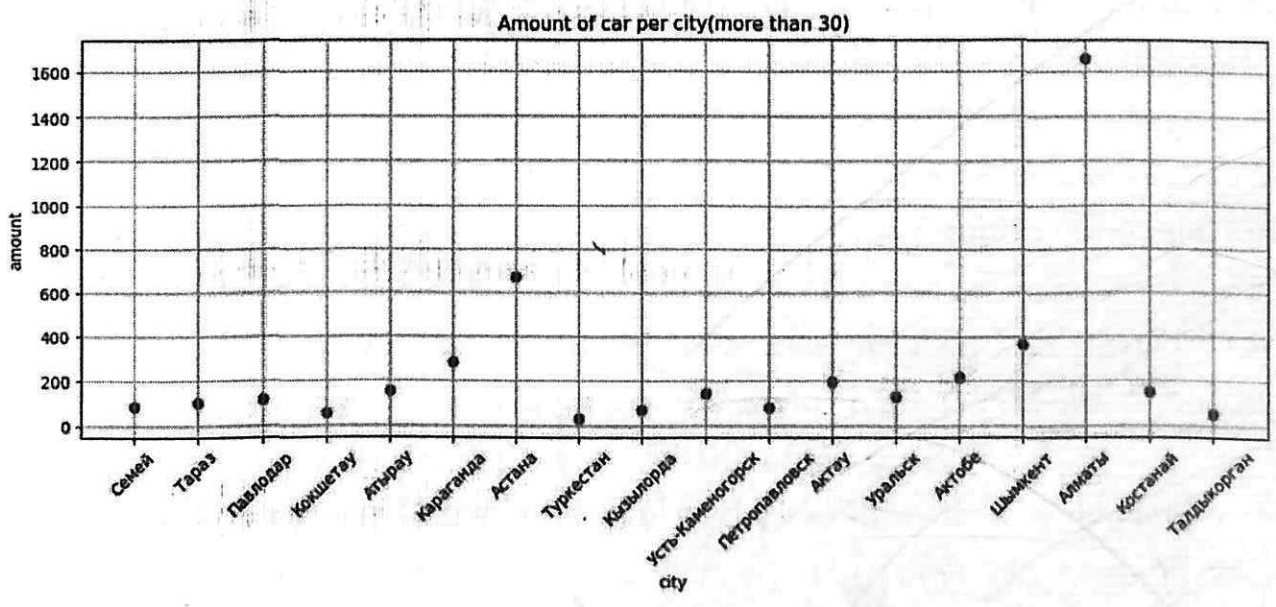


Figure 1.6: Amount per city

Figure 1.3 and Figure 1.4 have fully described the unbalance of data. In the next step we have cut the data by triangle method.

2. Multivariate regression

In all the problems considered so far - estimation, ranking, and probability estimation - the label space was a discrete set of classes. In this section, we consider the case of a real-valued target variable. The evaluation function is called $f : J \to \mathbb{R}$ mapping.

The problem of learning regression is to construct an evaluation function by examples $(x_i, f(x_i))$. For example, the task may be to find an evaluation function for the Dow Jones or London stock index FTSE 100, based on the selected economic indicators. Although at first it looks like a natural and not promising special difficulties generalization of discrete classification, there are differences - and significant. First, we move from a relatively low-resolution target variable to a variable whose resolution is infinite. An attempt to achieve such a resolution in the training of the evaluation function will almost certainly lead to retraining, and in addition, it is very likely that some of the values of the target variable appeared as a result of fluctuations that the model is not able to catch. It is therefore reasonable to assume that the examples are noisy and that the evaluation function should only capture the General trend, or form of the function.

To avoid this over-training, as shown in the example, it is recommended to choose the smallest possible degree of polynomial - often assuming a simple linear dependence. Regression is a task in which the difference between grouping and ranking models. The idea of the grouping model is to intelligently divide the object space into segments and train the simplest local model in each segment. For example, in decision trees, the local model is a classifier on the basis of the majority class. In the same vein, to obtain a regression tree, we could predict a constant value at each leaf node.

In a one-dimensional problem from the example, this would lead to a piece-

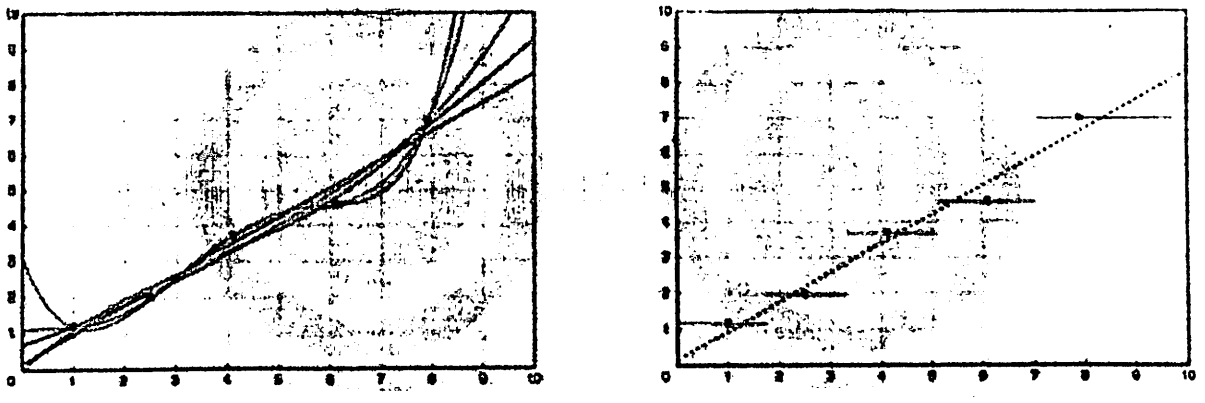


Figure 2.1: One dimensional problem observing

wise constant the function shown in Figure 2.1 on the right. Note that such a grouping model can be precisely adjusted to the specified points. In this sense, it is similar to a polynomial of a sufficiently high degree and suffers from the same disadvantages associated with the danger of retraining.

To better understand the phenomenon of retraining, consider the number of parameters in each model. The polynomial of degree n of parameters n + 1: so, the line y = Ah + B has two parameters, and the polynomial of degree 4, which passes through five points, has five parameters. Piecewise constant model with n segments has 2P - 1 parameters: n values y and p -1 values x, in which there are 'jumps'. So that the model can accurately interpolate the given points, it must have more parameters. The heuristic rule is that to avoid overfitting, the number of parameters estimated from the sample data should be significantly less than the size of the available sample.

We have seen that classification models can be calculated by applying the loss function to gaps, penalizing negative gaps (for incorrect classification), and rewarding positive ones (for correct classification). Regression models are calculated by applying the loss function to residuals $f(x) - f'(x)$. In contrast to the loss functions for classification, the loss function for regression is usually symmetric with respect to O (although it is acceptable to specify different weights for positive and negative residuals). The most common loss function is taken to be the square of the residuals. This is convenient from a mathematical point of view, and the hypothesis that the observed values of the function are true values contaminated by additive noise with a normal distribution is put forward as a justification.

Underestimating the number of parameters of the model, we will not be able

to reduce losses to zero, no matter how much training data is presented. On the other hand, a model with a large number of parameters will be more dependent on the training sample, and small sample changes can lead to a significant change in the model. This is sometimes called the dilemma of the offset dispersion: less complex the model is not severely affected by variance due to random fluctuations in the training data, but can give systematic error can not be eliminated even by increasing the volume of training data; on the other hand, a more complex model eliminates this bias, but can be subject to unsystematic errors due to variance.

We can refine this reasoning a bit by noticing that the mathematical waiting for the square of losses on the training example x can be represented as follows:

$$E[(f(x) - f'(x))^2] = (f(x) - E[f'(x)])^2 + E[(f'(x) - E[f'(x)])^2]$$

It is important to note that the expectation is calculated by different the training sets and hence the various evaluation functions, but the learning algorithm and example are fixed. The first term on the right side of equation [2.1] is zero if the evaluation functions are correct on average; otherwise, the algorithm exhibits a systematic error, or bias. The second term quantifies the variance of the evaluation function f (x) resulting from variations in the training set.

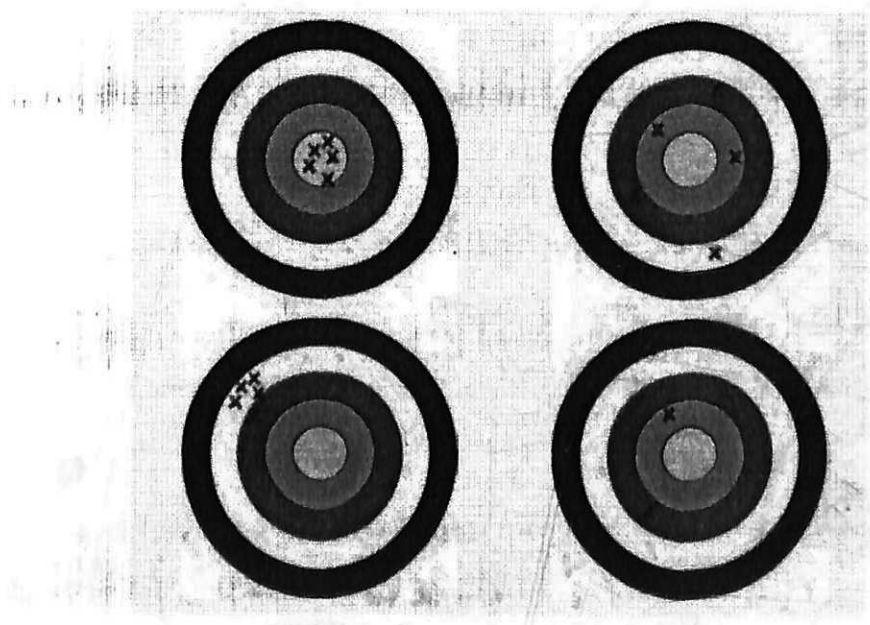


Figure 2.2: Dartboard illustration

The dartboard target on Figure 2.2 illustrates the concepts of displacement and variance. Different targets correspond to different learning algorithms, and

Darts denote different training samples. For the algorithms in the upper row is characterized by a small displacement - Darts on average are located close to the center of the target (the true value of the function for a particular x), and for the algorithms in the lower row - a large displacement. For algorithms in the left column there is a small variance and a large variance in the right column.

It is obvious that the best case is in the upper left corner, but in practice such luck is rare, and it is necessary to choose either low displacement and high dispersion (for example, to approximate the target function by a high degree polynomial), or high displacement and low dispersion (for example, to apply linear approximation). We will return to the displacement-variance dilemma more than once: although the decomposition shown above is not unambiguous for most loss functions (except quadratic), it serves as a useful conceptual tool for understanding over-and under-learning.

So far, we have been interested only in teaching predictive models with a teacher. That is, as a result of the training, we find the mapping of the space of objects x to the space of outputs y, using the marked examples.

This type of training is called "supervised", because of the presence in the training data of the target variable f(x), which must be provided by the "teacher" with certain knowledge of the true marking function l. Models are called "predictive" because the results they produce are either direct estimates of the target variable or provide enough information about its most likely value. Thus, we paid attention only to the type of models. In the rest of this Chapter, we'll look at three other kinds of models:

1. learning without a teacher predictive model on the example of predictive clustering;
2. teaching without a teacher descriptive model on the example of descriptive clustering and detection of Association rules;
3. supervised learning of descriptive models on the example of identifying subgroups.

Let's think about the nature of descriptive learning. The challenge is to get a description of the data - to generate a descriptive model. It follows that the output of the problem, being a model, has the same type as the input.

It turns out that it makes no sense to use a separate training set to generate a descriptive model, because we want the model to describe our actual data, not some test data. In other words, in descriptive learning, the task and the problem of learning coincide.

This complicates things: for example, the existence of some "unquestionable truth" or "gold standard" is unlikely. on which the model could be tested, and hence the evaluation of descriptive learning algorithms turns out to be much more difficult than in the case of predictive models. On the other hand, it can be said that descriptive learning brings out really new knowledge, so it is often at the intersection of machine learning and data mining.

The difference between predictive and descriptive models is evident in clustering problems. One possible interpretation of clustering is to train a new tagging function on untagged data.

Hence, we could define the "clusterizer" in the same way as the classifier, namely as the map $f : X \to Y$, where $Y = Y_1, ..., Y_k$ - lot new label. This interpretation is consistent with the prognostic view of clustering, since the scope of the displays space objects, and therefore it is generalized to non-presented model objects. A descriptive clustering model trained on data is a mapping $f : X \to Y$ with the definition area D rather than X. In any case, labels have no intrinsic meaning except that they Express belonging to the same cluster. Therefore, an alternative method of determining clusterization - equivalence relation or that the same most, partition with X or D.

The difference between predictive and descriptive clustering is subtle and not always clearly identified in the literature. Some well-known clustering algorithms, including the mean method , build a predictive model. That is, based on training data they generate a clustering model that can then be used to classify new data into clusters. This is consistent with the difference we introduced between the problem (clustering of arbitrary data) and the learning problem (building a clustering model on training data). However, this difference does not apply to descriptive clustering methods: here, the clustering model built from data D can only be used for clustering D. essentially, the goal is to train the appropriate clustering model for the available data.

Without additional information, any clustering is neither better nor worse any other. Good clustering is distinguished from bad by the fact that the data is

divided into compact groups or clusters. By "compactness" here we mean that on average two objects from one cluster have more in common (more similar) than two objects from different clusters.

Thus, it is assumed that there is some way to assess the similarity, or that it is usually more convenient, the divergence of an arbitrary pair of objects, that is, the distance between them.

If all our features are numeric, i.e. , the most obvious distance is the Euclidean metric, but other options are possible, some of which are generalized to non-numeric features. Most of the methods metric clustering depends on the concept of the "center of mass", or centroid of an arbitrary set of objects - the point at which a certain distance-dependent value is drawn to a minimum, calculated for all objects of the set. This value is called spread of the set. Thus, it is a good clustering, in which the amount of dispersion for all the clusters it is called vitriolic ally spread - much less scatter only dataset.

The analysis allows to determine the problem of clustering as finding such a partition $D = D_1 \cup ... \cup D_k$, which minimizes the intracluster spread. However, there are several disadvantages to this definition:

1. so, the problem has a trivial solution: put K = |D|, so that each "cluster" contains only one object from D, then its the spread will be zero;

2. if you fix the number of clusters To in advance, the problem cannot be solved efficiently for large data sets (it is NR-difficult).

The first problem is an analogue of retraining in the clustering problem. It can be solved by penalizing for large K. However, in most practical approaches it is assumed that it is possible to put forward a reasonable hypothesis about the value of K. The second problem remains: the computational impossibility to find a globally optimal solution to the problem for a large data set.

This situation is common in computer science and is resolved in one of two ways:

1. apply a heuristic approach that finds a "good enough" solution, even if not the best possible;

2. relax the task conditions, in this case allow "soft" clustering when the object can be incomplete" member of multiple clusters.

An interesting question: how to evaluate clustering models? In the absence of labeled data, we cannot use the test set as we would in classification or regression problems. As a measure of clustering quality you can take intra-cluster variation. For predictive clustering you can calculate the intracluster spread on the reserved data, which were not used to build clusters. Other evaluation method the quality of the clustering is shown if we know something about the objects that should or, conversely, should not fall into the same cluster.

A subgroup is essentially a binary classifier, so one of the ways to develop a system to identify sub-groups is to adapt some existing learning algorithm of a classifier. You may only need to select a search heuristics so that it reflect the specific objective of the subgroups (to identify the subset of data with a significantly different distribution of classes).

How to distinguish interesting from uninteresting to the group? To do this, you can build a contingency table similar to those used in the binary classification.

As we will see below, this boils down to the use of a variety of average completeness as an indicator of evaluation. Another idea is to consider subgroup as a partition of the decision tree and borrow the partition criterion from the learning of the decision trees. You can also use the x2 criterion for estimating how much each g; differs from the magnitude that would be expected from the marginals C and |G|. These indicators have a common feature: they prefer that the distribution by class in the subgroup and its complement differ from the General distribution in D, and in addition, they prefer large subgroups to small ones. Most of these metrics are symmetric in the sense that they evaluate the subgroup and the CE complement equally, which implies that they also prefer large complements to small ones. In other words, subgroups whose size is approximately equal to the juvin size of the entire set (ceteris paribus).

Now I will give an example of teaching descriptive models without a teacher. Assoniacii - these are things that tend to occur together. For example, when analyzing a shopping cart, we are interested in what products are often bought together. An example of an associative rule is "if beer, then chips"; this means that people who buy beer often buy potato chips as well. Detection of Association rules starts with finding the characteristic values that are often found together.

There is a superficial resemblance to the identification of subgroups, but these subjects of the so-called frequent sets are detected without any intervention of

the teacher, without the need for marked-up training data. Further, the rules describing the joint occurrence of feature values are derived from the subject sets. These associative rules have the form if - then and, therefore, similar to the rules of classification, with the difference that the part is not limited to a variable of a particular class, and can contain any sign (and even several features). Instead of adapting an existing learning algorithm, we need a new algorithm that first finds frequent subject sets and then converts them into associative rules. In this process, various statistical criteria must be taken into account to avoid the generation of trivial rules.

Having discussed the various tasks in the previous Chapter, we now have sufficient training to begin considering machine learning models and algorithms. This and the next two chapters are devoted to logical models, the trademark of which are logical expressions for dividing the space of objects into segments and, consequently, the construction of grouping models. The goal is to find a partition where the data in each segment is the most homogeneous - in terms of the problem to be solved. For example, in the case of classification, we are looking for a partition in each segment of which the object belongs mainly to one class, and in the problem a good regression is a partition for which the target variable is a simple function of a small number of independent variables. There are two large classes of logical models: tree-based and rule-based. A rule-based model consists of a set of implications, that is, rules of the form "if – then", in which part "if" defines a segment and part "then" defines the behavior of the model in that segment. A tree model is a special type of rule-based model where the "if" parts of all rules were organized as a tree.

In this Chapter, we consider methods of learning logical expressions, or Kozlov examples. These methods underlie both tree models and rule-based models. In conceptual learning, you only need to write a description of the positive class, and all that does not meet this description, mark as belonging to the negative class. We will pay special attention to ordering by degree of generality, as it plays an important role in the role of logical models. In the next two chapters, we will look at tree-like models and rule-based models that go far beyond conceptual learning because they can work with multiple classes, evaluate probabilities, solve regression and clustering problems.

Although this example was quite simple, the space of possible concepts - usu-

ally it is called the space of hypotheses - already quite large. Assume that there are three different lengths: 3, 4, and 5 meters, and the other three features have two values. Then all will be $3 \cdot 2 \cdot 2 \cdot 2 = 24$ possible objects. How many conjunctive concepts exist in which these features occur?

To answer this question, we will treat the absence of a trait as an additional value. Then it turns out $4 \cdot 3 \cdot 3 \cdot 3 = 108$ different concepts. And although, at first glance, it is a lot, you need to understand that the number of possible extensions - sets of objects - is much more: 224 more than 16 million!

So, if we take a random set of objects, the chances of us not being able to find a conjunctive concept that accurately describes these instances are greater than 100,000 to 1. This is actually a good thing, because it forces the researcher to generalize, going beyond the training data, and cover objects that he had not seen before. For Table 2.3 shows this space using order by community.

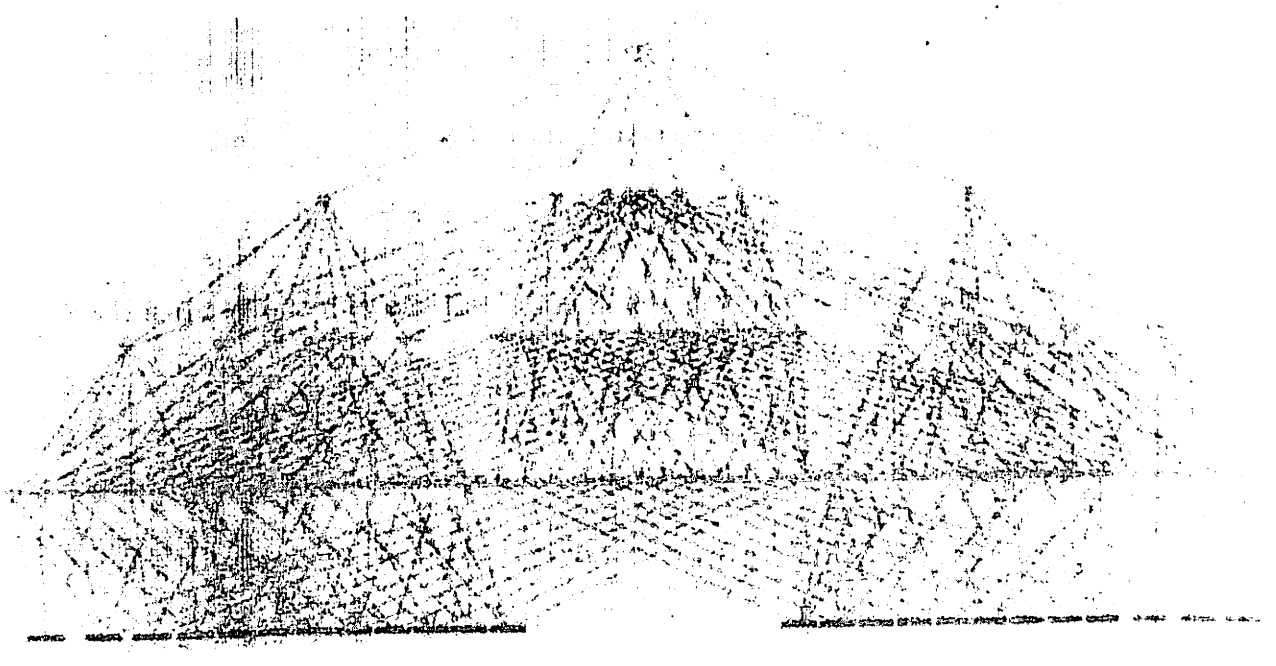


Figure 2.3: Negative and Positive graph

We can establish a useful connection between the spaces of logical hypotheses and the coverage graphs. Suppose we go up a path in the space of hypotheses leading from a positive example through a series of generalizations to an empty concept. The last one covers all positive and negative examples and therefore occupies the upper right point (Neg, Pos) on the coverage graph.

The starting point, being a single positive example, occupies the point (0,1)

on the coverage graph. In practice, it is customary to fill the space of hypotheses with the lower element, which does not cover any example and therefore is less common than any other concept. If we take this point as the beginning of the path, it turns out that we start the journey from the lower left corner (0,0) on the coverage chart.

We have seen several languages of conjectures for conceptual learning, including conjunctions of literals (possibly with internal disjunction), conjunctions of horn disjuncts, and disjuncts in first order logic.

It is intuitively obvious that these languages differ in terms of expressiveness: for example, the conjunction of literals is simultaneously the conjunction of horn's conjuncts with the empty "if" part, so horn's theories are strictly more expressive than conjunctive concepts. But a more expressive language of concepts has a drawback: it is harder to teach. The section of the theory of computational learning in which this question is studied is called educability.

First, we will need a model of learning: a clear formulation of, what we mean when we say that we teach the language of concepts. One of the most common models of learning is the model of almost specific (probably approximately correct - RACES) learning. RAS-learning means that there is a learning algorithm that gives almost the right result in most cases. The model introduces an error correction in the case of atypical examples, hence the expression "approximately correct". The model also recognizes the possibility that sometimes it gives completely incorrect results, for example, if the training data contains a lot of atypical examples, hence the expression "probably".

2.1 Regression tree model's type

Tree models are among the most popular in the machine training. For example, the algorithm is based on the recognition of poses in the sensor Kinect motion gaming consoles Hoh lies the crucial tree (actually an ensemble of decision trees called a random forest). The trees are expressive and easy to understanding of the mechanism of interest, which they call the experts in computer science, associated with the recursive nature, allowing algorithms divide and conquer.

We see that the path in the space of hypotheses can be transformed into an equivalent feature tree. To get a tree equivalent to the i-th concept, counting from the bottom point of the path, we can either trim the tree by combining the i-left most leaves into one leaf representing the concept, or mark the i-leftmost leaves as positive and the rest as negative, turning the feature tree into the decision tree[7].

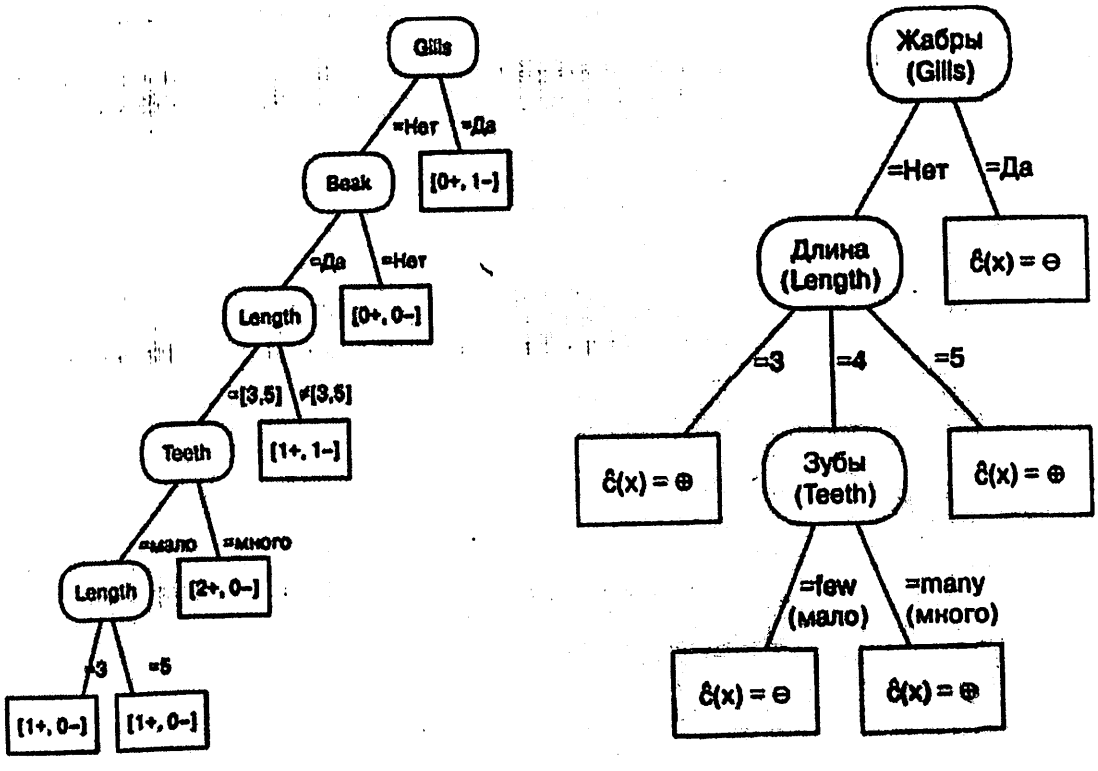


Figure 2.4: Decision Tree without Branching

In the decisive trees do not use the internal disjunction for sign, having more than two values, instead for each value is produced branching. Marks that do not necessarily follow the order of leaf traversal from left to right are also allowed. This tree is shown in Figure 2.4 on the right. This tree you can convert to a

Boolean expression in many ways.

There are many other logical expressions equivalent to the concept defined by the decision tree. Is it possible to obtain an equivalent conjunctive concept? Interestingly, the answer to this question is negative: some trees represent conjunctive concepts, but most do not, and the one shown is one of them. Decisive trees build more expressive than conflictive concepi. In fact, from the fact that the decision trees correspond to DNF expressions, and that each logical expression can be written as equivalent to DNF, it follows that the decision trees are as expressive as possible: they cannot separate only data with conflicting labels when the same object is labeled with different labels. This explains why data that is not conjunctively separable, as in our example, is nevertheless separated by a decision tree.

There is a potential problem with the use of such an expressive language of hypotheses. Let L be the disjunction of all positive examples, then L is represented as the disjunctive normal form. Obviously, L covers all positive examples - in fact, the extension of L is exactly a set of positive examples. In other words, in the space of hypotheses consisting of DNF expressions (or decision trees), L is the smallest generalization of positive examples, but it does not cover any other objects. Hence, L is not generalized beyond the set of positive examples, but just remembers them - that's confused here about retraining! If you reverse this reasoning, it turns out that Odia of the ways to avoid retraining and promote learning is to IOM to consciously choose restrictive language of hypotheses, such as conjunctive concepts: language even surgery LGG usually generalizes beyond the positive examples. And if our language is expressive enough to represent an arbitrary set of positive examples, then we have to make sure that the learning algorithm uses some other mechanisms that guarantee generalization beyond the examples and the lack of retraining - this is called inductive.,t S. 1еще11ие.m algorithm. As we will see below, learning algorithms that work in the expressive spaces of hypotheses have an inductive bias towards less complex hypotheses: either implicitly - by the way of searching in the space of hypotheses, or explicitly - by penalizing the complexity in the objective function.

Tree models are not limited to classification, but can be used to solve almost all machine learning problems, including ranking and probability estimation, regression and clustering. The tree structure common to all these models can be

defined as follows.

These functions depend on the problem to be solved. For example, for classification problems, the set of objects is homogeneous if most of them belong to one class, and the most appropriate label is the majority class. For clustering problems a set of objects is homogeneous if they are near, and the most suitable label will be some standard, for example, the average

The first option corresponds to the marking based on the majority class, and it is recommended in most textbooks when considering decision trees. And I also recommended it when discussing the label(D) function in the context of algorithm. In many cases, this is the most practical approach.

However, it is important to understand what assumptions underlie this labeling: it is required that the distribution of classes in the training set be representative and the costs are uniform, or, more generally, that the product of the expected costs and class relationships be equal to the class ratio observed in the training set. (Hence a useful way of manipulating and training set to reflect the expected class relationship: to simulate the expected relationship of classes C, we should include positive examples in C times more than negative if $C > 1$, and include negative examples are $1/s$ times more than positive if with < 1. Following we will come back to this advice.)

So, suppose that the distribution by class is representative and that false-negative prediction (for example, undiagnosed in a patient disease) costs approximately 20 times more expensive than false positive. As we have just seen the optimal under such working conditions is the + - ++ markup, which means to filter out negative examples only the second sheet is used. In other words, the two right sheets can be merge into one - their common parent node. The merge operation of all the leaf of the subtree is called the reduction of the subtree. This process is illustrated in Fig. 5.6. The advantage of reduction is that it allows simplify the tree without affecting the selected work point – sometimes this is useful if we want to pass the tree model to someone else. The disadvantage is that we lose as a ranking, as seen in Table 2.5 below. Therefore, reduction is not recommended if (i) you intend to use the tree not only for classification but also for ranking and grading probabilities, and (ii) if you are unable to determine the expected operating conditions with sufficient accuracy. One of the popular algorithms of reduction of solving trees is called reduction. The algorithm uses a

separate reducing set of labeled data, which was not presented during training, since the reduction never has been improved accuracy on training data. However, if the simplicity of the tree is not an important argument, I recommend not to reduce it and choose a working one the point is only by marking the leaves; this can also be done using a reserved dataset[5].

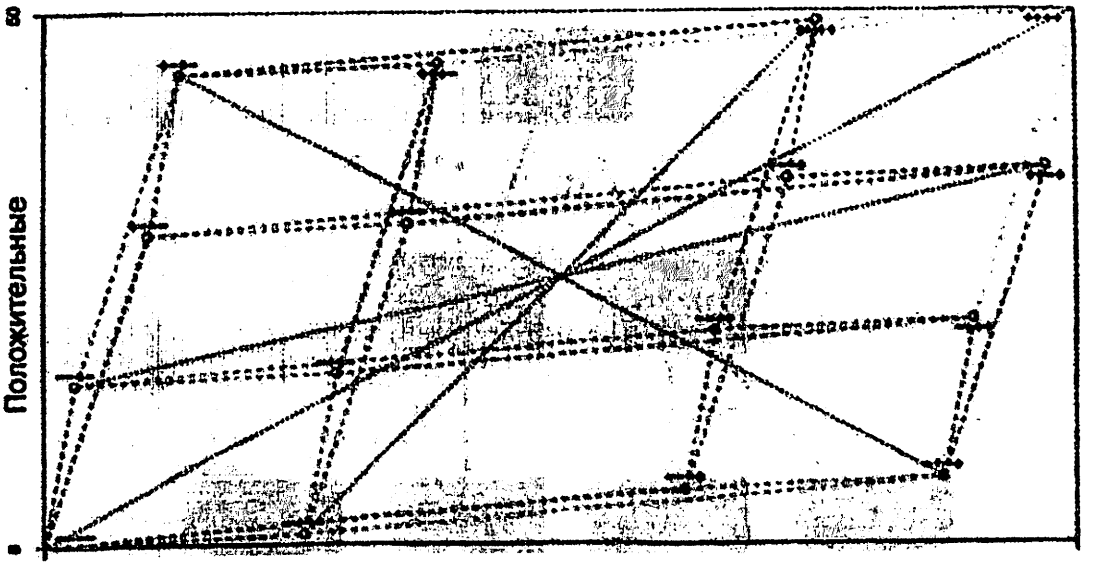


Figure 2.5: Sign conditions

I just mentioned in passing that one way to ensure that the training set reflects the correct working conditions, is to duplicate positive or negative examples so that the ratio of classes in the training set is equal to the product of the expected costs on class relationships when you deploy the model. In fact, it changes the ratio of the sides of the rectangle representing the coverage space.

The advantage of this method is its direct applicability to any model, without the need to conjure with search heuristics or evaluation measures. The disadvantage - to increase the training time, and, in addition, it can not affect the trained model. I'll illustrate with an example.

It is important to note that the latter relation will not change if we multiply all numbers related to the number of positive examples per one and the same number C. This means that the root of the Gini index is intended for minimization of relative impurity, and therefore insensitive to changes in the distribution of classes. In contrast, the relative frequency in the case of an index The Gini includes the fraction n1/n, which changes if you increase the number of positive examples.

Something similar happens with entropy. As a result, these two splitting criteria give preference to descendants, covering more examples'.

Even better to clarify the situation will help the picture. Separation criteria so same as loyalty and average completeness, are contours in the spaces of the coating and RHP. Because of the nonlinear nature, these contours are curvilinear. They also pass on both sides of the diagonal, as we can swap left and right child, without changing the quality of separation. The landscape of impurity can be imagined as a mountain viewed from a height - the top is a ridge along the ascending diagonal representing the separation, when whose descendants have the same impurity as the parent. This mountain is falling on both sides of the ridge and reached the level of the earth at the top of the RHP and at the opposite point (which could be called the "depression of the RHP"), since it is here that the impurity is zero. Contour contours mountains - passing through all the points with equal height.

Look at the Figure 2.5 above. The two split options that you had to choose between example (before adding positive examples) are shown by points on the graph. I did a six contour lines in the left top corner of the chart: two split options multiplied by three criteria divisions. Any criterion prefers a separation whose contour passes above (as close as possible to the top of the RCP): as you can see, only one of the three (the root of the form index Jinn) prefers right the upper division. Upon rice Figure 2.6 and Figure 2.7 below shows how the picture changes after increasing the number of positive examples 10 times (the coverage graph would not fit on the page, so I pictured how it looks in the space of the RCP, and the grid lines shows you how to change the distribution of classes). Now all three criteria divisions prefer the top right option because "mountains" entropy index Gini turned clockwise (index Gini in a greater degree than the entropy), whereas the root of the index of the Gini did not move from place.

The moral of the story is that when you are learning a decision tree or the probability tree you use entropy or index the Gini as a measure of impurity - which is what is done in almost all tree training packages - then be prepared for the fact that the model will change if you will change the distribution of classes by adding extra examples.

At the same time, if we take the square root of the Gini index as a measure,

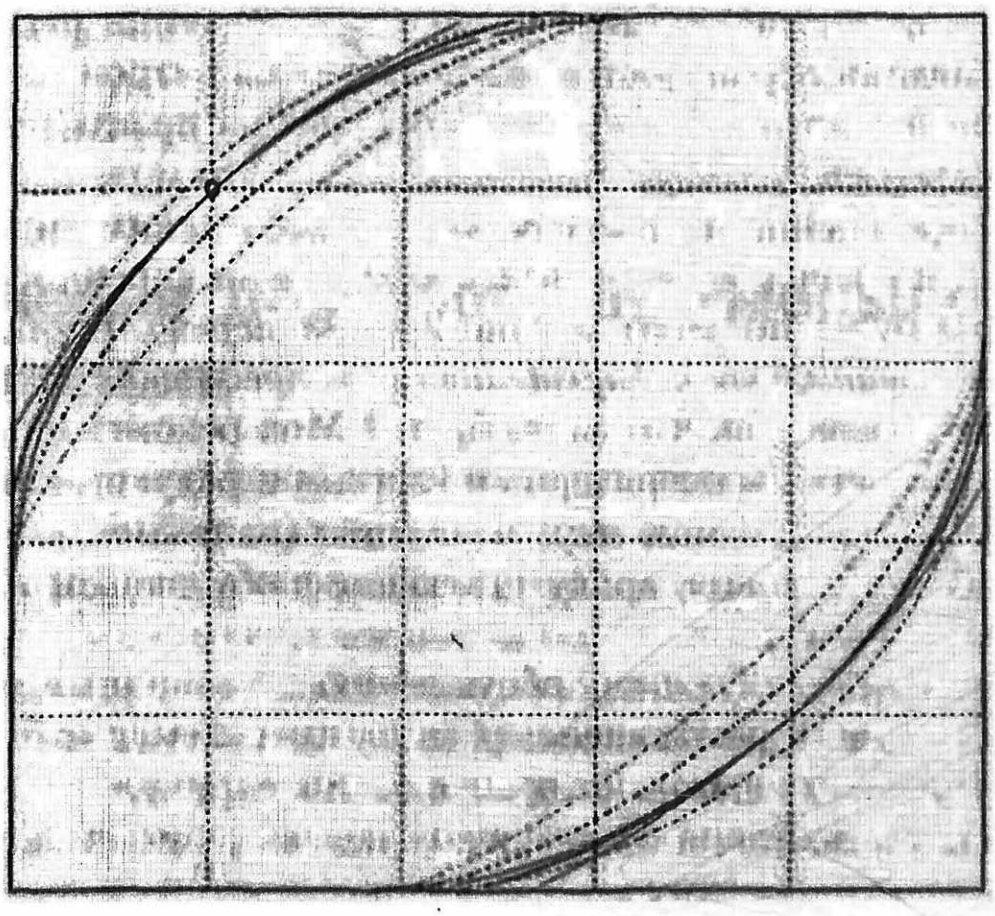


Figure 2.6: Gini's entrophy branching

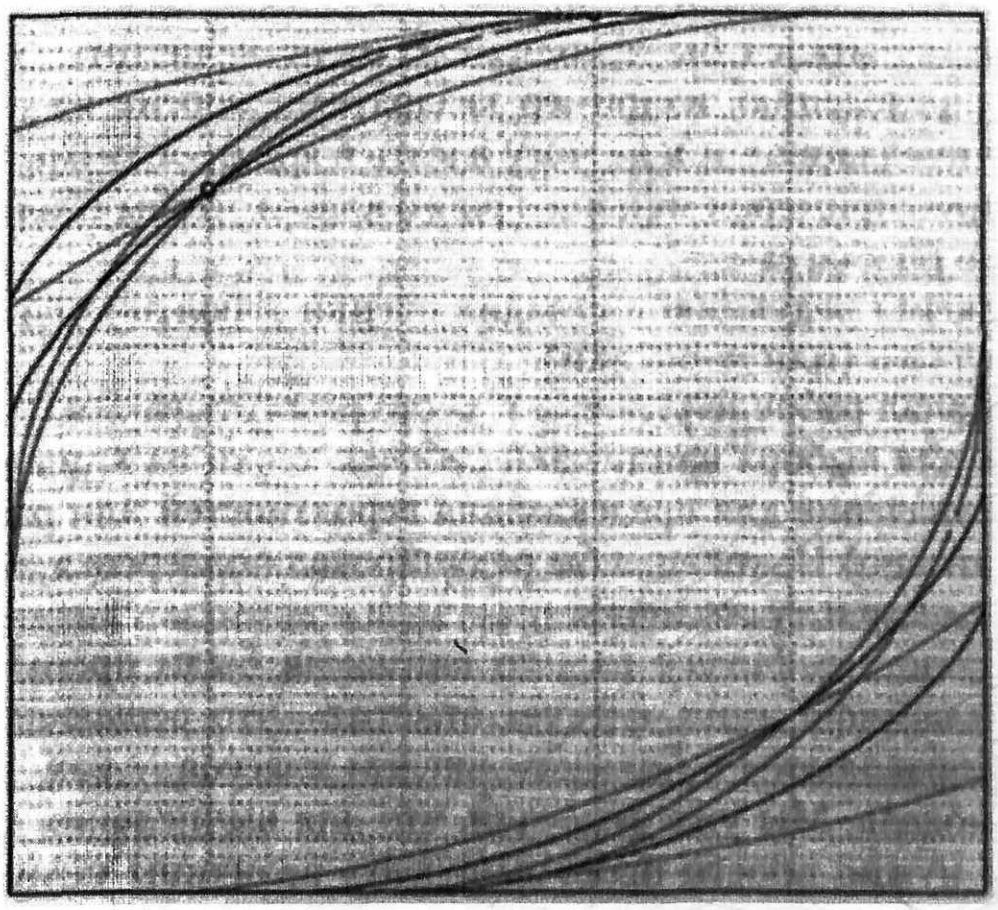


Figure 2.7: Gini's entrophy branching with testing packages

we will get the same tree each time.

Let's summarize the previous discussion of tree models. "As you say would we train the decision tree on an existing data set?" - you asked. Here is a list of steps I would take.

1. First of all, I would put the main goal to achieve good ranking behavior, because having a good ranking officer, I can get a good classification and a good evaluation of probabilities, but the opposite may not be true.

2. So I would try to use a measure of impurity that is not sensitive to distribution, such as the square root of the Gini index; if it is not offered, and I could not get into the program code, I would have resorted to the addition of minority class examples to balance the class distribution.

3. I have disabled reduction ratio and the smoothed estimate of the probability of using amendment Laplace (or t-score).

4. If I knew the operating conditions at the deployment site, I would use them to select the best operating point on the CCP curve (that is threshold value for probability prediction or tree layout).

5. (Optional) Finally, I would reduce subtrees whose leaves all have the same label.

Although the discussion focused mainly on the problems of binary classification, it should be noted that the decision trees without any effort to cope with a large number of classes - as, indeed, any grouping model. We have already noted that to calculate a multi-class measures of impurity you just need to sum the values of impurity for each class according to the scheme "one against the others".

In regression problems, the target variable is continuous, not binary, in this case, the variance of the set of target values is defined as the root mean square distance to the middle:

Most of the pitfalls characteristic of regression trees apply to clustering trees: the smaller clusters, the discrepancy is usually small, so they are vulnerable to overfitting. Recommends reserve reduction set to remove lower divisions if they do not improve the connectivity of clusters of the reducing set. Single examples can dominate: in the example above, deleting the first trade reduces the pairwise

divergence from 2.94 to 1.5, and therefore it will be difficult to find something better than the split that puts this trade in a separate cluster.

An interesting question arises: how to mark the leaves of a clustering wood? Intuitively, it seems reasonable to mark the cluster as the most representative object for it. The most representative object can be define as such, for which the total difference with all the others, such an object is called a medoid. For example, in cluster A 100, the most representative transaction is 6, because it the difference with trades 3 and 8 is 1, while the difference between trades 3 and 8 is 2.

Similarly, in cluster T202 the most representative transaction 7. However, it is not necessary that the most representative object only.

A typical situation in which it is possible to simplify the calculations required for finding the best split and get the unique label of the cluster occurs when a divergence is defined as the Euclidean distance, calculated by numeric signs. Note Figure 2.8 shows that if Dis(x, x) is the square of the Euclidean distance, then Dis(D) is the doubled root mean square the Euclidean distance to the average. This allows to simplify calculations, because as the average and the root mean square distance to the middle possible to calculate O(IDI) steps (for one pass through the data), not O(IDl2) steps, necessary if we have nothing but a matrix of divergences. On themselves in fact, the standard Euclidean distance is simply the sum of the variances individual signs.

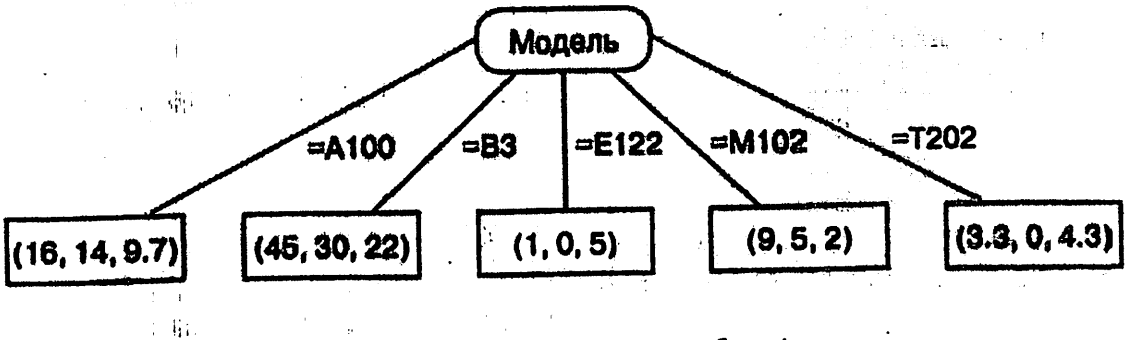


Figure 2.8: Model's classification

Experimental results demonstrating more accurate estimates of the probabilities are obtained if we abandon the reduction of the tree and smooth the empirical probabilities using the correction Laplace, given in a recent paper and confirmed. The degree of insensitivity of splitting criteria of the decision tree to the imbal-

ance of the classes or value of the incorrect the classifications were studied and explained in the works of [2], [1] and [8]. Of the three above-mentioned criteria for separation only the root of the Gini index is insensitive to such an imbalance classes and costs.

It should be remembered that the high expressiveness of tree models, according to comparing, for example, with conjunctive concepts means that you need to be insured against retraining. In addition, the greedy algorithm type "divide and rule".

3. Correlation analysis

Values can be either independent or related by functional or stochastic (probabilistic) dependence. Functional dependence of quantities is realized when each value of one quantity (argument) corresponds to a certain the value of another value. An example of a functional dependence is the length of a circle $l = 2nr$ depending on its radius r. Obviously, there is no such correspondence for random variables, therefore, strict functional dependences occur only when the values are not subject to the action of random factors.

In most cases, there are dependencies between variables, in which each value of one quantity (argument) corresponds not to a certain value of another quantity, but to a set its possible values are a certain distribution. Such dependence it's called stochastic, or probabilistic. For example, with the same area of land with equal amounts of fertilizers remqved different crops. Random values – the amount of fertilizer applied and the harvest – are related to each other stochastic dependence: the second variable is influenced by a number of factors in addition to the number of introduced fertilizers (precipitation, air temperature, etc.). In addition, measurement of the values of both variables is inevitably accompanied by.

A special case of probabilistic dependence is the correlation dependence – stochastic dependence between random variables. Values at which there is a functional relationship between the values of one quantity and the average values of another quantity.

Let's return to the above example. The relationship between the amount of fertilizers and the harvest is correlated, because experience shows that the average yield and the amount of fertilizers introduced into the soil are related to each other by functional dependence. The term "correlation" (from lat. correlatio – relationship, connection, dependence) appeared in the XIX century through

the work of the English mathematician Karl Pearson (1857-1936) and English anthropologist and psychologist Francis Galton (Galton) (1882-1911).

The points depicted on the coordinate plane (x_i, y_i), where x_i and y_i are the values of the first and second variables are called the correlation field. The analytical function approximating (approximately describing) the observed empirical values is called the regression function. Function name (from lat. regressio – movement ago) gave F. Galton, who, studying the relationship between the growth of parents and their children discovered the phenomenon regressions to the mean: in children born to very high parents, the growth tended to be closer to the average value.

The regression function reflects the trend of change of one quantity under the influence of another and is constructed thus, to the empirical point correlation fields lay as close to her as possible. The regression function can be linear, parabolic, hyperbolic, logarithmic, etc.

The presence of correlation between variables do not always mean that these quantities are directly related to each other: the observed relationship often exists due to other variables (not the two in question), and the study of values can be linked by itself through latent (hidden from the researcher) variables.

An example of such an artifact ("artificial" result) is the relationship between the level of intelligence and the level of human income discovered by American psychologists. The latent variable that determines this correlation is the structure of society: the research conducted in modern Russia gives different results. Another example is the correlation of the speed of recognition of the image as it the copies (fast pulsing) presentation and human vocabulary (latent variable – the General intelligence of the subject). As can be seen, the relationship of variables in psychology is too complicated, so they can be explained by a single reason.

The correlation, in contrast to the functional, shows only the tendency of one quantity to change under the influence of another, therefore, on the basis of correlation can be argued only about the degree of connection between variables, but not about the existence of a causal relationship between them. In other words, the fact that variables are correlated does not mean that one causes the other, however, it makes it possible to put forward such a hypothesis. For example, between the child's academic performance in primary school and the age at which he or she learned to read, there is a correlation. However, this the fact does not

follow causal dependence: you can find a child who has learned to read long before entering the University. school and Vice versa.

Consider, is a correlation between students' anxiety level and their software testing results the end of the course "Mathematical methods in psychology". This fact on the one hand, it can be explained by the fact that the excitement experienced by some students could lead to the fact that they did worse with the test task, and more calm students were able to successfully demonstrate their abilities. But isn't it just as plausible to think that the test itself is a worrying factor? Less than capable (usually lazier) students are intimidated by testing, and capable and responsible not find in it nothing alarming, except another test of knowledge. Causation is not possible interpret without experimental verification.

Sometimes in psychological studies, a random correlation is established that is not due to any cause. An example of this correlations is the relationship between anxiety and English performance in middle school students. Does this indicate that increased anxiety makes the student work harder? Not at all. When tested on the Taylor anxiety scale, girls show higher rates than boys. It is also known that girls in the middle classes tend to have higher grades in English compared to boys. To establish such the relationship separately for boys and girls have not yet managed to anyone.

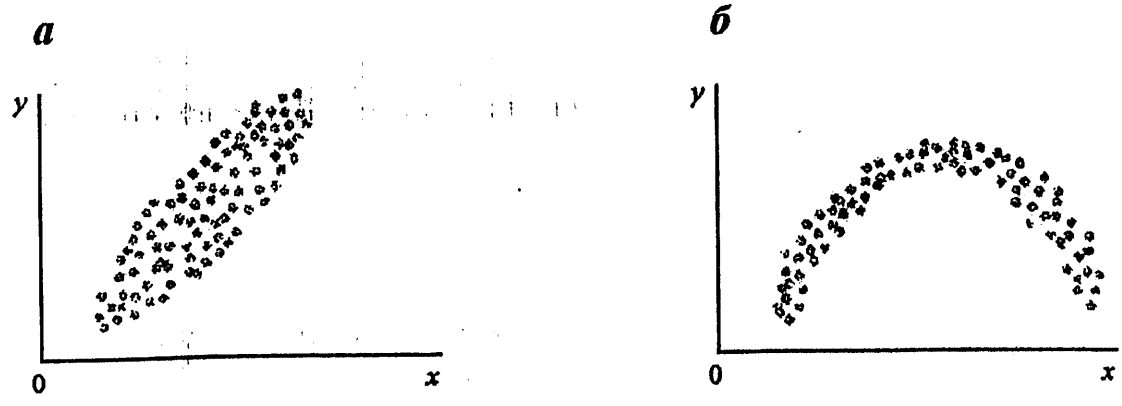


Figure 3.1: Regression

A well-known example of nonlinear correlation is the first law Yerkes–Dodson[4]: as you increase the intensity of motivation quality the activity varies along a bell-shaped curve: first it rises and then gradually decreases (Another example of a nonlinear communication is Hick's law: the speed of information processing is

proportional to the logarithm of the number of alternatives. Conclusion about the form of pair correlation can be made, by drawing the correlation field on the coordinate plane.

Pair linear correlation, in turn, can be positive ("direct") and negative ("reverse"). With a positive correlations with the increase of one trait increases on average the other, in the case of a negative correlation with the increase of one sign of the other in average is declining. Examples: the level of personal anxiety positively correlates with the risk of stomach ulcer, the number of children in the family is negative correlates with the indicator of their intelligence, the increase in sound volume accompanied by a feeling of increasing its tone, and the number of daily cigarettes smoked is negatively correlated with duration lives. As can be seen from Figure 3.2, Figure 3.3, Figure 3.4, Figure 3.5, in the case of pair linear correlation the correlation field is an ellipse. The closer correlation, the more compressed the ellipse; in the case of functional it is transformed into a straight line, and in the absence of a link – into a circle.

The possibility of indirect estimation of some characteristics over other due to the fact that they are dependent on a number of common factors. If the same factor F affects both variables, then between they are always correlated in empirical research.

In this case, the observed spread of variables X and Y is due not only to action of the General factor F, but also other reasons, irrelevant in relation to it. As a result, each value of the variable X corresponds to the distribution of the variable To its undefined value:

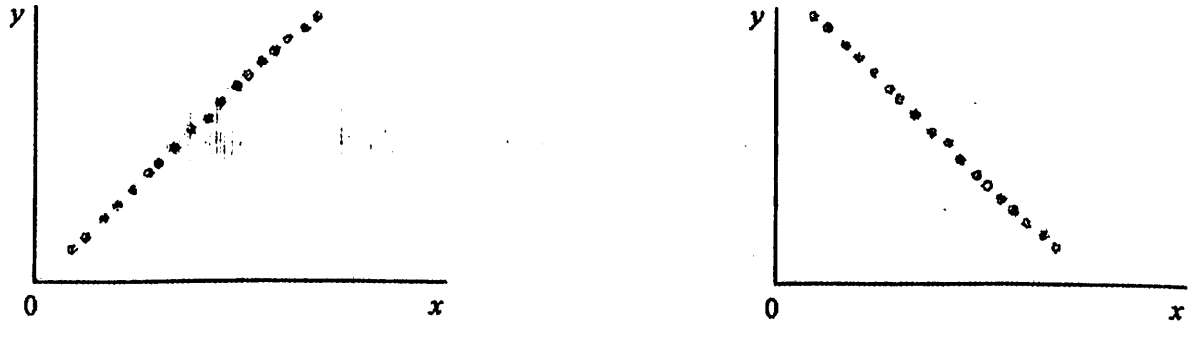


Figure 3.2: R1

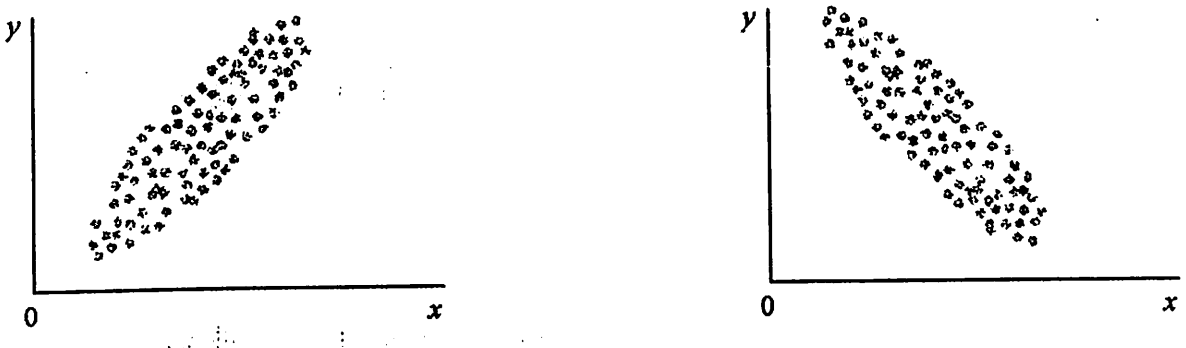


Figure 3.3: R2

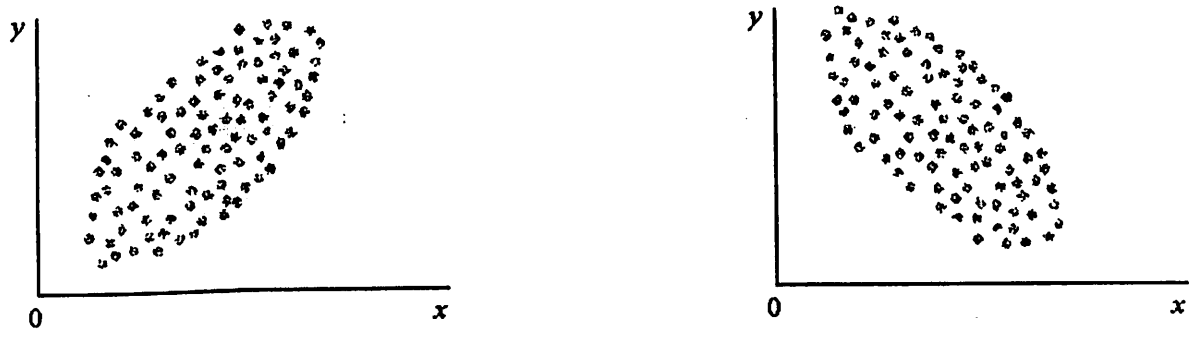


Figure 3.4: R3

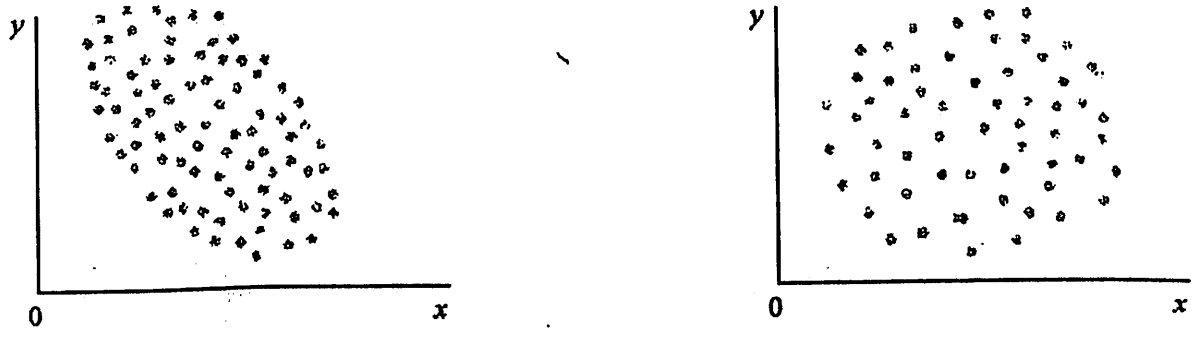


Figure 3.5: R4

In other words, changing the variable Y when changing X can it should be presented as two components (relevant to the General factor F and irrelevant to it), and the variance of the measured value consists of the "factor" and "residual" components:

$$\sigma^2 = \sigma_F^2 + \sigma_R^2$$

The factor component of variance is related to the dependence of the considered values and is determined by the effect of the General factor on both variables.

If the factor component of the variance is zero, the quantities studied are independent.

The residual component of the dispersion causes the action of many other factors that do not affect both variables at the same time (individual differences of subjects, measurement errors, etc.). The residual variance causes the values of y_i to be randomly scattered at a fixed value of x_i, so it is also called "random". At zero random variance there is a functional relationship between the values (each value of one value corresponds to a single value of another). Thus, the ratio between the factor and random components of dispersion can serve as a quantitative measure of tightness (force) correlation between the values: the greater the proportion of the factor the components in the total variance, the relationship between the quantities closer to functional. The ratio of factor dispersion to the total is called the coefficient of determination:

$$R = \frac{\sigma_F^2}{\sigma^2} = \frac{\sigma_F^2}{\sigma_F^2 + \sigma_R^2}$$

The coefficient of determination is a dimensionless non-negative value varying from 0 to 1 (it is often expressed as a percentage). It shows the proportion of the total variation of one variable due to the variability of another variable. The value of the coefficient of determination does not change with increasing or decrease by the same number or the same number of times of all variable value.

3.1 Kendall's correlation coefficient

Kendall's correlation coefficient has the same properties as the Spearman coefficient(varies from -1 to +1, for independent random variables is zero), but it is considered more informative. The first step in the calculation of the "Tau" Kendall is variable series ranking (the same series values are assigned the average rank number). The first variable must be ordered in ascending order of rank. Kendall's correlation coefficient is determined by the formula

$$\tau = \frac{4 \cdot \sum R_i}{n(n-1)} - 1$$

Here n is the sample size (number of matched pairs); Ri is the number ranks in the second variation series greater than the given rank number and below it. Testing the hypothesis of the significance of the Kendall rank correlation coefficient is to compare the calculated value of the Tau coefficient modulo with the critical values:

$$\tau_\alpha(n) = z_{(1-\alpha)} \sqrt{\frac{2(2n+5)}{9n(n+1)}}$$

Kendall's concordance coefficient is used when the set of objects is characterized by several sequences of ranks, and the researcher needs to establish a statistical relationship between these sequences. Such tasks arise, for example, in the analysis of expert assessments: several experts rank one and the same subjects for a certain quality, and a psychologist to conduct an in-depth analysis of the situation and make an informed decision the degree of consistency of the views of the expert group needs to be determined. Kendall's concordance coefficient is determined by the formula

$$W = \frac{12 \sum D_i^2}{n^2(n^3-n)}$$

Values of the concordance coefficient, as opposed to the coefficient correlations are enclosed in the interval $0 \leq W \leq 1$. The coefficient of concordance it is equal to one if all rank sequences coincide. If the opinions of experts (ranking order) fully opposite, the concordance coefficient is zero (coefficient correlation in this case will be -1).

3.2 Spearman's rank correlation coefficient

Each of the two sets is located in the form of variation series with assignment to each member of a series of the corresponding ordinal number (rank) expressed as a natural number. The same series values are assigned the average rank number. The compared features can be ranked in any direction: in the direction of deterioration (rank 1 receives the biggest, fastest, smartest, etc. of the subject), and Vice versa. The main thing is that both variables are arranged in the same way. Spearman rank correlation coefficient is based on the formula

$$r_s = 1 - \frac{6\sum d_i^2}{n^3 - n)}$$

If in the variation series for X and Y there are members of the series with the same rank numbers, the formula for the Spearman's correlation coefficient must be amended T_x and T_y on the same rank:

$$r_s = 1 - \frac{6\sum d_i^2}{n^3 - n - \frac{1}{2}(T_x + T_y))}, \; T = \sum(t_k^3 - t_k)$$

Here l is the number of groups in the variational series with the same rank numbers; tk is the number of members in each of the l groups. Spearman's rank correlation coefficient, as well as linear, varies from -1 to +1, but the value of Spearman's rank correlation coefficient is always less than the value of Pearson's linear correlation coefficient: $r_s < r$. Testing the hypothesis of the importance of Spearman's rank correlation coefficient is carried out differently depending on the sample size.

The value of the correlation coefficient $r_s = 0,206$ falls into the range of acceptable values, which does not allow to reject the null hypothesis. The correlation coefficient is not it differs from zero, indicating that there is no relationship between the expression qualities of the test subject at the moment and the ideal representation.

4. Modelling with tree based ensemble models

The assumption that the regression model is linear in parameters is convenient for the construction of numerical methods, but does not always agree well with knowledge about the subject area. This paragraph discusses cases where the model regression nonlinear in the parameters, when the linear model adds a non-linear transformation of the initial indication or target symptom, and when a non-quadratic loss function is introduced. The general idea in all these cases is the same: the nonlinear problem is reduced to the solution sequences of simpler linear problems.

In practice, there are situations when the linear regression model seems unreasonable, but to offer an adequate nonlinear model.

4.1 Random forest

The first says that everything that has some part of the body is a fish. Second specializes this rule, saying that there should be a pair of uncorrected parts of the body. The third specializes even more, saying that the fish is anything with a couple of gills. A reasonable search strategy is to test hypotheses in this order and move to the particular only when the more General is excluded by negative examples. This is how IP systems work, working from top to bottom. By using a simple technique - the explicit representation of substitutions of terms instead of variables by adding literals equal to the sequence of sentences shown above can be rewritten as:

```
fish(X):-bodyPart(X,Y).
fish(X):-bodyPart(X,Y),Y=pairOf(Z).
fish(X):-bodyPart(X,Y),Y=pairOf(Z),Z=gill.
```

Figure 4.1: Pseudocode

As an alternative to iterating over literals considered as candidates for inclusion in the body of a sentence, we can derive them from the data by acting from the bottom up.

By forming the smallest generalization[3] of each of the literals in the first literal example from the second example, we obtain all the generalized literals discussed above.

In this brief discussion of learning rules in first order logic, we left behind many important details, and therefore look at the problem turned out to be overly simplistic. Although the task of learning the prologue sentences it is possible to formulate very briefly, naive approaches to its decision are computationally unrealizable, and the devil is covered in details. Basic described above approaches can be supplemented by including background knowledge that affects the ordering of the hypothesis space by generality.

However, this cannot be determined by purely syntactic means, and logical inference is required. Another intriguing possibility offered by first-order logic is learning recursive sentences.

This blurs the distinction between background predicates that can be use in the body of the hypothesis, and target predicates that are required teach. In addition, there are computational problems, such as a work in progress problem. But this does not mean that can not be done at all. There are the methods of simultaneous training of several interrelated predicates and the generation of new background predicates, which have never been observed, are also adjacent to the subject.

4.2 Adaboost

Random forests resulting from the techniques described earlier can be naturally used to assess the importance of variables in regression and classification problems. The next method of such evaluation was described by Breiman[1].

The first step in evaluating the importance of a variable in a training set is to train a random forest on that set. During the process of building a model for each element of the training set called out-of-bag-error (error on unselected samples). Then, for each entity, the error is averaged over the entire random forest[6].

In order to assess the importance of the j-th parameter after training, the values of the j-th parameter are mixed for all the training set records and the out-of-bag error is considered again. The importance of the parameter is estimated by averaging the difference between the out-of-bag errors before and after mixing the values in all trees. The values of such errors are normalized to the standard deviation.

Sample parameters that give larger values are considered more important for the training set. The method has the following potential drawback — for categorical variables with a large number of values, the method tends to consider such variables more important. Partial mixing of values in this case may reduce the effect of this effect. Of the groups of correlating parameters, the importance of which is the same, the smaller groups are selected.

Strengthening, or boosting (boosting), - technique of building ensembles, on first the view is similar to banking, but it uses more sophisticated methods to add variety to the training sets. The basic idea is simple and seductive. Let us assume that we have trained a linear classifier on a certain data set and found that the error rate of learning is e. We want to add one more classifier to the ensemble, which makes mistakes less often, than the first. This can be achieved, for example, by duplicating incorrectly classified objects: if our model is a base a linear classifier, it will lead to the displacement of the average values of the classes in the direction of duplicate.

In the amplification algorithm, another component is required - the confidence factor a for each model in the ensemble; we will use it to form ensemble prediction equal to the weighted average of individual models. It is clear that we would like that a increased with decreasing e usually it is calculated by the formula

$$\alpha_t = \frac{1}{2} - \log \frac{1-\epsilon_t}{\epsilon_t} = \log \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$$

the rationale for which we will give below. The basic gain algorithm is given in the Figure 4.1. For rice, Figure 4.2 on the left is shown as a reinforced ensemble of five basic linear classifiers can achieve zero learning error. It is obvious that the resulting decisive boundary is much more complicated than the one that can build one of the basic linear classifier. On the contrary, the ensemble of the five basic linear classifiers, built by the method of bagging, I gave five very similar decisive boundaries, and this is explained by the fact that the amplified samples were very similar[2].

```
public override void Train(LabeledDataSet<double> trainingSet)
  {
   // Инициализируем наши коэффициенты D_i
   if (_weights == null)
   {
    _weights = new double[trainingSet.Count];
   for (int i = 0; i < trainingSet.Count; i++)
    _weights[i] = 1.0 / trainingSet.Count;
   }

   Random rnd = new Random();
         double minimumError = double.PositiveInfinity;
   _d = -1;
   _threshold = double.NaN;
   _sign = 0;

   // Цикл по всем координатам пространства
   for (int d = 0; d < trainingSet.Dimensionality; d++)
   {
    // Для оптимизации в случае, если пространство огромной размерности
    if (rnd.NextDouble() < _randomize)
     continue;
```

```
// Сортируем данные для эффективного поиска прямой x_d = threshold
double[] data = new double[trainingSet.Count];
int[] indices = new int[trainingSet.Count];
for (int i = 0; i < trainingSet.Count; i++)
{
 data[i] = trainingSet.Data[i][d];
 indices[i] = i;
}
Array.Sort(data, indices);

// Определяем максимальную ошибку
double totalError = 0.0;
for (int i = 0; i < trainingSet.Count; i++)
 totalError += _weights[i];

// Инициализируем значение ошибки для поиска минимальной ошибки
double currentError = 0.0;
for (int i = 0; i < trainingSet.Count; i++)
 currentError += trainingSet.Labels[i] == -1 ? _weights[i] : 0.0;
```

```
for (int i = 0; i < trainingSet.Count - 1; i++)
{
 // Обновляем ошибку при рассмотрении текущего объекта
 int index = indices[i];
 if (trainingSet.Labels[index] == +1)
  currentError += _weights[index];
 else
  currentError -= _weights[index];

 // Если идут одинаковые данные, то порог не ищем
 if (data[i] == data[i + 1])
  continue;

 // Определяем потенциально возможное значение для threshold
 double testThreshold = (data[i] + data[i + 1]) / 2.0;



 // Запоминаем значение threshold, если ошибка минимальна, и определяем знак
 if (currentError < minimumError) // Классификатор с _sign = +1
 {
  minimumError = currentError;
  _d = d;
  _threshold = testThreshold;
  _sign = +1;
 }
 if ((totalError - currentError) < minimumError) // Классификатор с _sign = -1
 {
  minimumError = (totalError - currentError);
  _d = d;
  _threshold = testThreshold;
  _sign = -1;
 }
}
}
```

Figure 4.2: Code in python

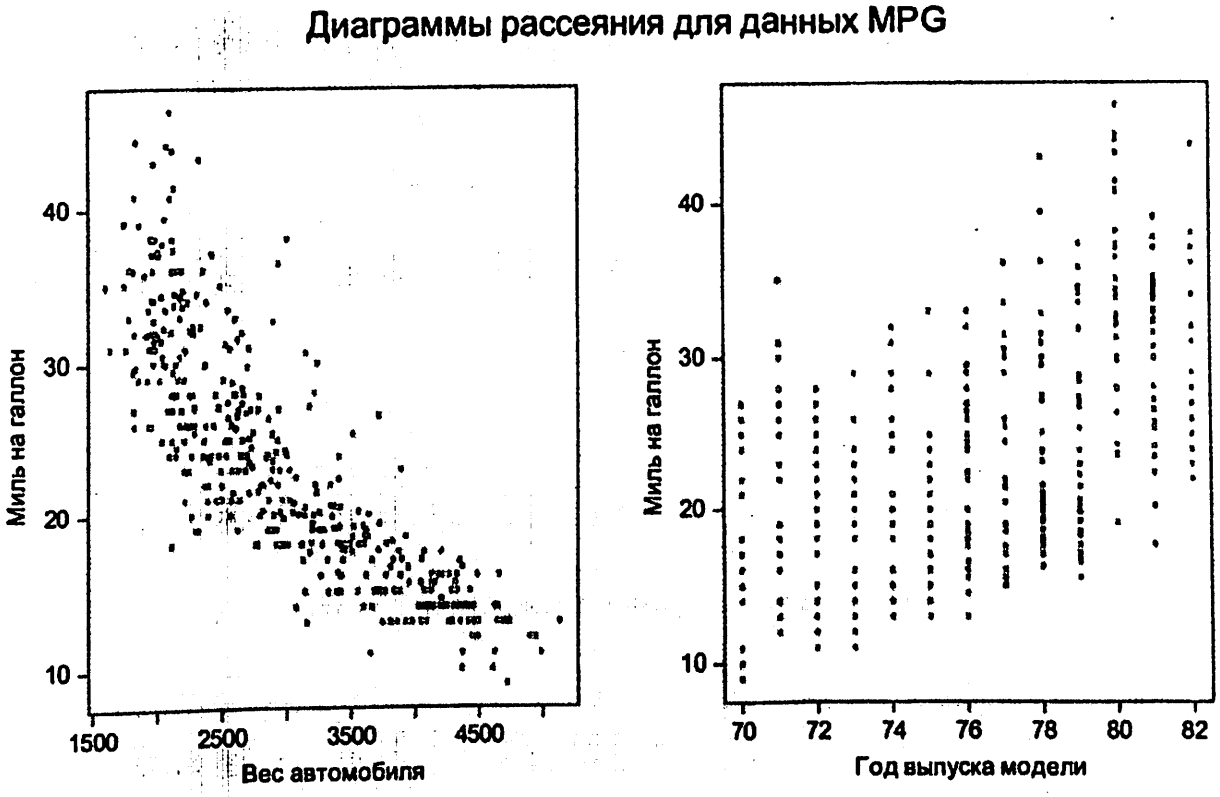


Figure 4.3: MPG data observation

Now that we have become more familiar with the frequently used methods of constructing ensembles, let's talk about how to explain the differences in their quality, and then turn to some of the many ensemble methods described in the literature.

Ensemble construction techniques are a good way to better understand the bias-variance dilemma we discussed in section 3 in the context of regression. Generally speaking, there are three reasons why a test case is misclassified by a model. First, it is simply inevitable if in a given feature space objects of different classes are described by the same features.

In a probabilistic context, this happens when the conditional distributions of P(X/Y) overlap, so that some object has probabilities of belonging to several classes different from zero. In such a situation, the best one can hope for is an approximation of the target concept.

The second reason for classification errors is the insufficient expressiveness of the model to represent the target concept. For example, if the data is not linearly separable, even the best linear classifier will be make mistakes. This is the shift of the classifier, between it and expressiveness there is an inverse relationship.

It may seem that low-offset models are generally preferable. However, in the practice of machine learning there is a heuristic rule in models with low bias, usually high variance, and Vice versa. Variance is the third source of classification errors. The model has a high variance if its decisive boundary depends heavily on the training data. For example, in the case of the nearest neighbor classifier, the segments of the feature space are defined by a single point in the training set, so if we move a point in the segment adjacent to the decision boundary, then the border itself will move. In tree models, the variance is high for another reason: if you change the training data so much that the root of the tree will change the selected to separate feature, then most likely the whole tree will be different. An example of a model with low variance is the basic linear classifier, because it averages all points in the class.

Now look at the Figure 4.2. The basic ensemble of linear classifiers, built by the method of bagging, trained piecewise linear a crucial boundary, which expression is superior to any single linear classifier. This suggests that bagging, like any ensemble method, is able to reduce the displacement of the base model, which is initially characterized by a high displacement, such as a linear classifier. However if we compare this with the results of the amplification method shown in Figure 4.3, then we will see that the reduction of the displacement resulting from the bugging, much less than the result of amplification.

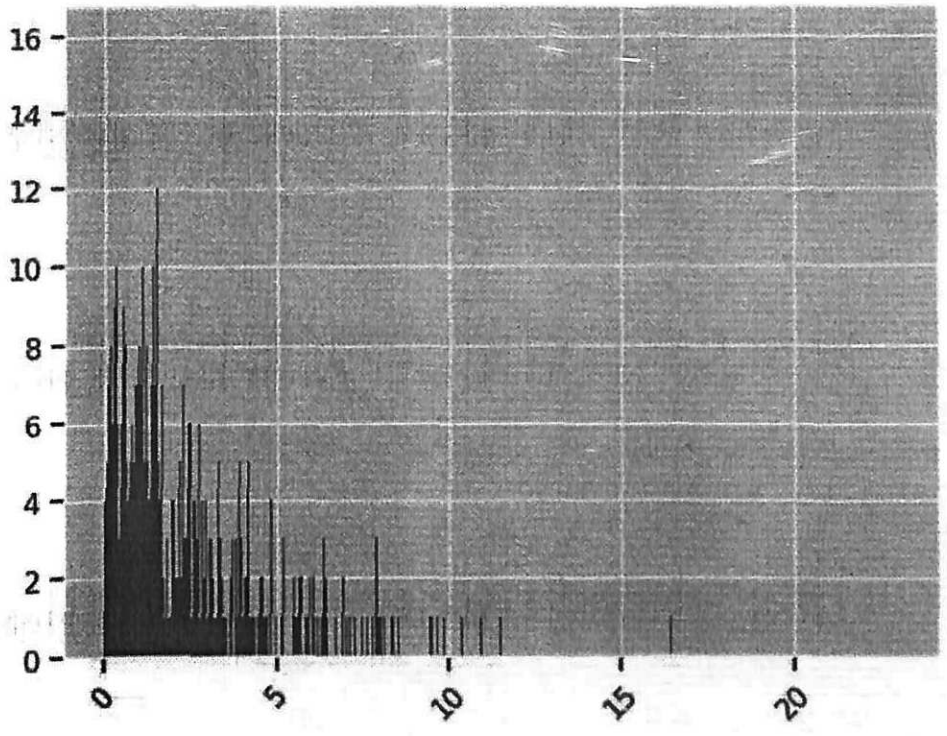


Figure 4.4: Bar expansion for price

Gain can be interpreted differently in terms of gaps. Intuitively, the gap is the signed distance to the decisive boundary, and the sign shows whether we were on the right side of the border. Experimentally, it was noted that the gain gives a good effect in terms of increase the gaps of the examples, even if they are already positioned correctly the side of the decisive border. The quality improvement on the test set as a result of the gain can continue even after the learning error reduced to zero. When you consider that the gain originally arose in the context of PAC-learning is not designed for the increase in gaps, this result may seem surprising.

There are many other methods for constructing ensembles, in addition to bagging and gain. The main differences are related to how the predictions of the basic models are combined. Note that this question in itself could be to determine how the problem of the study: considering the predictions of some base classifiers as features to train a meta-model, which will combine them in the best way. For example, in the amplification method we could would be to train the weights A1 and not remove them from the frequency error of each base model. Learning a linear metamodel is called stacking.

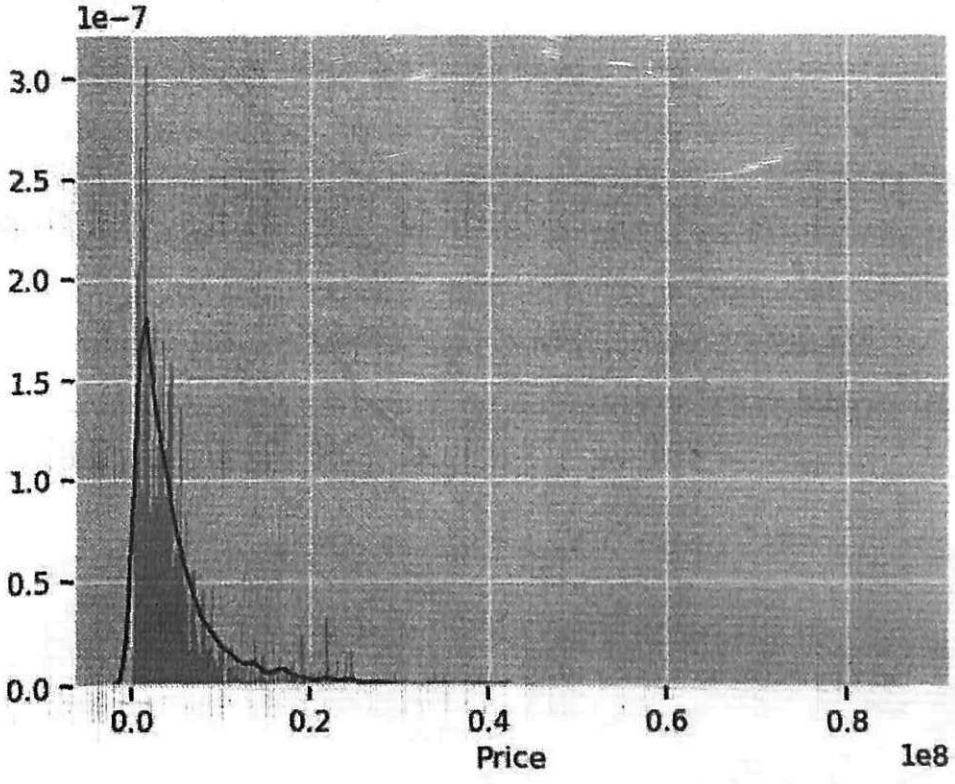


Figure 4.5: Statistical prediction for price

There are several variations on this theme, for example as metamodels decisive trees were used. It is also possible to combine different basic models into a heterogeneous ensemble, and the diversity is achieved by the fact that the basic models are trained by different algorithms, so you can use the same training set. For some basic models you can specify different parameters, for example, an ensemble can include multiple support vector machines with different complexity parameter values, which determines the extent to which clearance errors are tolerable.

So, in General, an ensemble of models consists of a set of basic models and a metamodel, which is trained to decide how to combine the predictions of the base models. Training of the metamodel is implicitly implied assess the quality of each base model, for example, if the metamodel is linear, as in the case of stacking, a weight close to zero means that the corresponding the basic classifier does not make much contribution to the ensemble. You can even assume that the base classifier gets a negative weight, then in context other basic models of its prediction should be inverted. Possible it would go even further and try to predict the expected quality of the base model even before its training! Having formulated it as a problem of training on at the meta level, we are entering the field of meta-learning.

In this short section 3, we have discussed some fundamental ideas of methods building ensembles. All of these methods have one thing in common: they build several basic models based on modified training data, and then apply this or that way of combination of predictions or the assessment of the individual base models for prediction of the entire ensemble. We have considered two common ensemble methods: Baggins and amplification. A good introduction to ensembles of models is available in brown's work (2010). Standard reference for combining classifiers – work Kuncheva (2004), and a more up-to-date review is available in Zhou (2012).

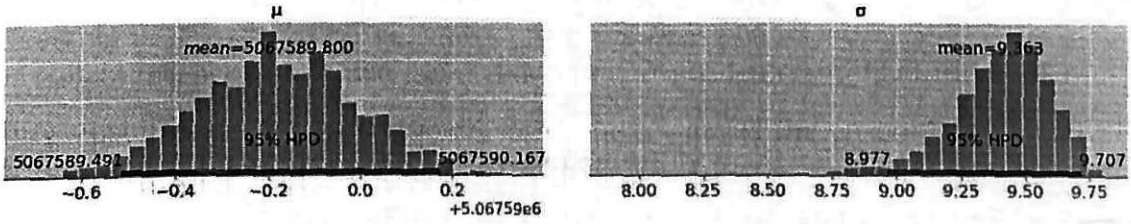


Figure 4.6: Data descriptive statistics after prediction

Machine learning is a discipline that is equally practical, so is computing. In some cases, we can prove that the particular algorithm converges to a theoretically optimal model under certain assumptions, but still need real data, for example, to investigate the extent to which these assumptions are satisfied in the subject area under consideration, or whether the convergence rate is high enough for the algorithm to have practical value. Therefore, we run specific models or learning algorithms on one or more data sets, perform measurements, and use the results to answer our questions. All these activities are called experiments in machine learning.

In the natural Sciences, the experiment can be considered as addressed the nature of the issue of a scientific theory. For example, in Arthur Eddington's famous experiment, set in 1919 to test Einstein's General theory of relativity, the question was asked: are the rays deflected light in the gravitational fields of massive cosmic bodies, such as the sun? To answer this question, the observed position of the stars was recorded under various conditions, including a total solar Eclipse. Eddington was able to prove that the results of measurements could not be explained by the laws of Newtonian physics, but were consistent with the General theory of relativity.

As follows from the example, taking loyalty as an indicator of quality, we

do an implicit assumption is that the distribution of classes in the test case is representative of the working context in which the model will be deployed. Besides, if we only measured fidelity in the experiments, we can't. go to the average completeness later, when we understand that you need to consider changing class distribution.. Therefore, it is recommended to remember enough information so that if necessary it was possible to reconstruct the contingency table. Such a sufficient set of measurements can be count the frequency of true positive results, the frequency of true negative (or false positive) results, the distribution by class and the size of the test case. Machine learning is a discipline that is equally practical, so is computing. In some cases, we can prove that the particular algorithm converges to a theoretically optimal model under certain assumptions, but still need real data, for example, to investigate the extent to which these assumptions are satisfied in the subject area under consideration, or whether the convergence rate is high enough for the algorithm to have practical value. Therefore, we run specific models or learning algorithms on one or more data sets, perform measurements, and use the results to answer our questions. All these activities are called experiments in machine learning.

In the natural Sciences, the experiment can be considered as addressed the nature of the issue of a scientific theory. For example, in Arthur Eddington's famous experiment, set in 1919 to test Einstein's General theory of relativity, the question was asked: are the rays deflected light in the gravitational fields of massive cosmic bodies, such as the sun? To answer this question, the observed position of the stars was recorded under various conditions, including a total solar Eclipse. Eddington was able to prove that the results of measurements could not be explained by the laws of Newtonian physics, but were consistent with the General theory of relativity.

As follows from the example, taking loyalty as an indicator of quality, we do an implicit assumption is that the distribution of classes in the test case is representative of the working context in which the model will be deployed. Besides, if we only measured fidelity in the experiments, we can't. go to the average completeness later, when we understand that you need to consider changing class distribution. Therefore, it is recommended to remember enough information so that if necessary it was possible to reconstruct the contingency table. Such a sufficient set of measurements can be count the frequency of true positive results,

the frequency of true negative (or false positive) results, the distribution by class and the size of the test case.

5. Results

In the evaluation phase, all models were redesigned to meet the data quality requirements and models in the data shortage. The data were balanced according to various criteria and methods, but did not lead to the highest evaluation result.

```
Kendall rank correlation coefficient(Close to 1 is better): 0.7174682188508716
Spearmans rank corr coefficient(Close to 1 is better): 0.8813390018550825
```

Figure 5.1: Results of coefficients

The coefficients of data evaluation also led to not the best results, but in accordance with the requirements of the models, we can say that the data can be extracted useful information.

5.1 Random forest

Having estimates of relevant quality indicators for our models or learning algorithms, we can use them to select the best representative. We have discussed two key concepts: confidence intervals and significance criteria. It is necessary to understand them if you want to understand modern approaches to interpretation of results of experiments in machine learning; however, it should not be forgotten that modern approaches are subject to critical analysis. Note also that the methods described here are only a small fraction of the widest range of possibilities.

In the coefficient analysis Kendall's assumes that large differences between indicators quality better, than smaller, but no other assumptions about their commensurate not is done. In other words, the differences are considered as ordinal, not real-valued values. In addition, these differences are not assumed to have a normal distribution and that means, among other things, the criterion is too sensitive to emissions.

```
y_pred2 = randomf_cv.best_estimator_.predict(X_test)
```

```
print(np.sqrt(mean_squared_error(randomf_cv.best_estimator_.predict(X_train), y_train)))
print(np.sqrt(mean_squared_error(y_pred2, y_test)))
```

```
1670808.857660936
3018682.8392637796
```

Figure 5.2: Mean squared error values for training and testing data

Figure 5.1 shows the regression results of last model for training and testing data and the last score is about 71% (Figure 5.3).

```
randomf_cv.best_score_ #Result is not good
```

```
0.7082047522846117
```

Figure 5.3: Result of model

5.2 Adaboost

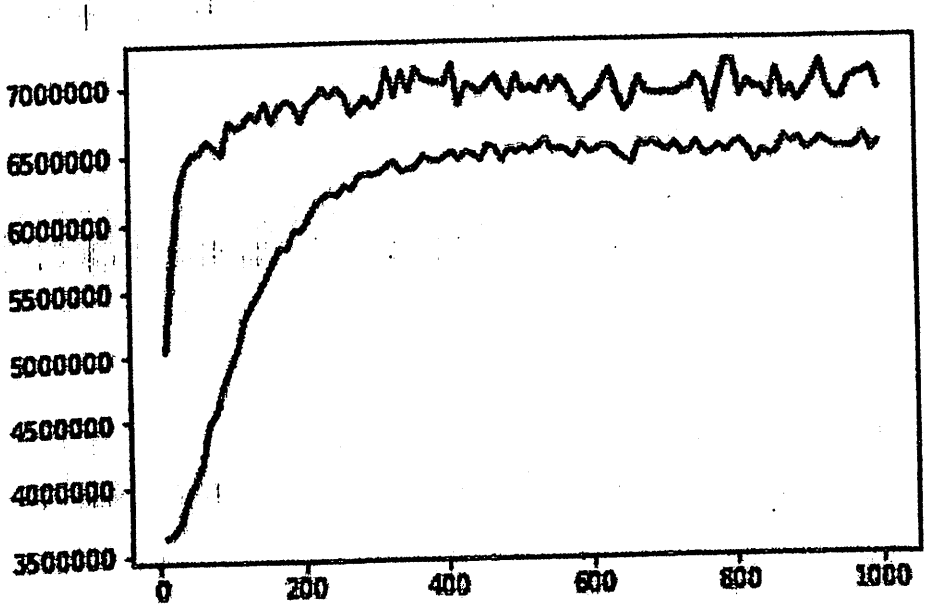
Before model training, the criterion of Friedman says, show whether the average grades overall on significant differences, but further analysis at the pairwise level is needed comparisons. There is an analogy here with clustering in the sense that the second quantity measures the spread between centroids of ranks - we want it to be great, and the third - variation between all ranks. Friedman's statistics is the ratio of the first value to the second. As for the interpretation of the experimental results, we considered confidence intervals and significance criteria.

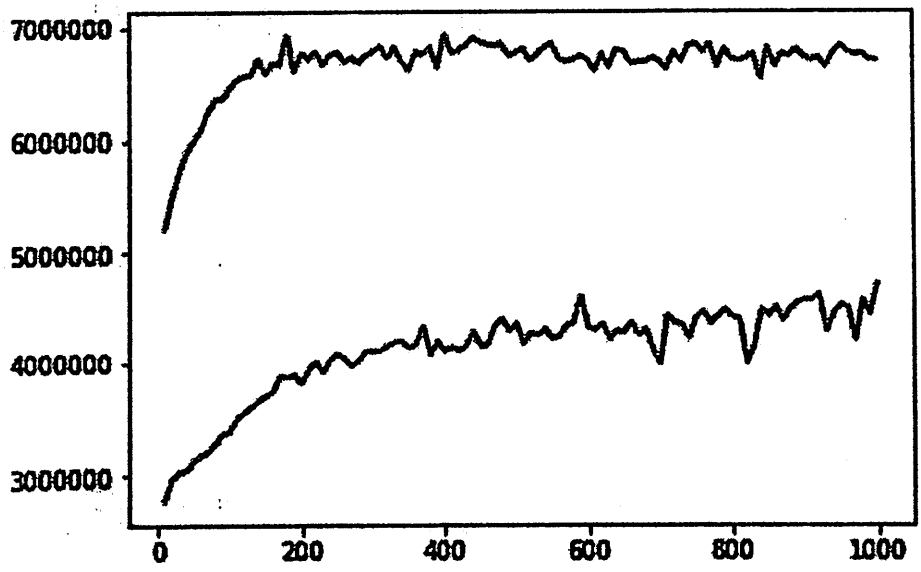
Figure 5.4 shows Mean Absolute Errors for training and testing data. Result is quite better than bagging method.

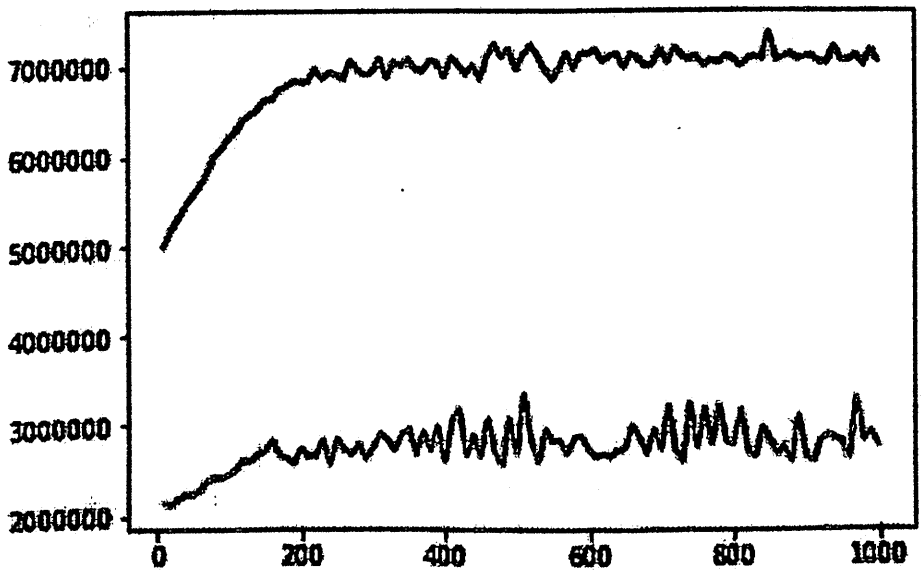
```
print('MAE test with GSV: ', mean_absolute_error(y_pred_test_best, y_test))
print('MAE train with GSV: ', mean_absolute_error(y_pred_train_best, y_train))
```

```
MAE test with GSV:  2743987.1198835857
MAE train with GSV:  2829078.748196364
```

Figure 5.4: Mean Absolute Error values for training and testing data




7000000
6000000
5000000
4000000
3000000
0
200
400
600
800
1000


7000000
6000000
5000000
4000000
3000000
2000000
0
200
400
600
800
1000

And the final touch of the analysis and evaluation of the model(Figure 5.4, Figure 5.5, Figure 5.6) showed at a depth of 1.2 and 3 good performance. But, we could get a better answer at great depth.

6. Conclusion

It should be noted that the use of significance criteria in these models of machine learning as experimental work is the subject of lively discussions. The importance of experiments in machine learning emphasized already by Pat Langley, two made effect works; subsequently, however, he criticized the experimental methodology in machine learning, became, in his opinion, insufficiently flexible. From other authors that are critical of the current state of affairs in this area, mark Drummond and Demsar. By this criterion, adaboost model with low depth shows better result as tree based random forest. As we have seen at previous section, we could use powerful computers to take highest results at some maximal depth.

References

[1] L. Breiman. "Statistical Modeling: The Two Cultures.. [Google Scholar]". In: *University of California* 48 (2003). DOI: https://dialnet.unirioja.es/servlet/articulo?codigo=5080250.

[2] A. Liaw and M. Wiener. *Classification and Regression by randomForest.* URL: http://www.stat.berkeley.edu/users/breiman/.

[3] J. R. Quinlan. "Bagging, Boosting and C4.5.. [Google Scholar]". In: *Proceedings of the Thirteenth National Conference on ArtIficial Intelligence* 14 (2006). DOI: http://www.cs.ecu.edu/~dingq/CSCI6905/readings/BaggingBoosting.pdf.

[4] A. Riccardi, F. Fernández-Navarro, and S. Carloni. "Cost-Sensitive AdaBoost Algorithm for Ordinal Regression Based on Extreme Learning Machine. [Google Scholar]". In: *IEEE Transactions on Cybernetics* 44 (2014). DOI: https://ieeexplore.ieee.org/abstract/document/6719563.

[5] M. R. Segal. "Machine Learning Benchmarks and Random Forest Regression. [Google Scholar]". In: *Division of Biostatistics* CA 94143-0560 (2003). DOI: https://escholarship.org/uc/item/35x3v9t4.

[6] D. P. Solomatine and D. L. Shrestha. "AdaBoost.RT: a boosting algorithm for regression problems. [Google Scholar]". In: *IEEE International Joint Conference on Neural Networks* 19 (2004). DOI: https://ieeexplore.ieee.org/abstract/document/1380102.

[7] J. L. Wei, D.Y. Lin, and L Weissfeld. "Regression Analysis of Multivariate Incomplete Failure Time Data by Modeling Marginal Distributions. [Google Scholar]". In: *Journal of the American Statistical Association* 84.408 (1989), pp. 230–245. DOI: https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1989.10478873#.XQYz0LwzaUk.

[8] Питер Флах. *Машинное обучение, Наука и искусство построения алгоритмо которые извлекают знания из данных.* ДМК, Москва, 2015.