Wrocław University
of Science and Technology

Faculty of Pure and Applied Mathematics
Field of study: Applied Mathematics
Specialty: Data Engineering

Master's Thesis

APPLICATION OF MACHINE LEARNING AND STATISTICAL METHODS IN ASTRONOMY

Tymoteusz Cieślik

keywords:
machine learning, astronomy, regression, classification, model evaluation

short summary:
The thesis will deal with the introduction and implementation of machine learning methods to an astronomical set of data. The main goal of the thesis will be to create predictive models for two different aspects – redshift regression and classification of objects. During the work we will present the main astronomical and statistical descriptions of the problem. By creating these models we will be able to compare and evaluate them, so that we could pick the most accurate ones as the solution of a considered issue.

Supervisor	Dr hab. inż. Maciej Zięba	.....	.....
	Title/degree/name and surname	grade	signature

*For the purposes of archival thesis qualified to:**

a) category A (perpetual files)

b) category BE 50 (subject to expertise after 50 years)

** delete as appropriate*

stamp of the faculty

Wrocław, 2022

Contents

1 Introduction **3**
1.1 Related works . . . 4

2 Theoretical aspects **7**
2.1 Astronomical aspects . . . 7
2.2 Mathematical theory . . . 8
2.2.1 Machine learning algorithms . . . 8
2.2.2 Metrics of model evaluation . . . 11
2.2.3 Other definitions . . . 13

3 Exploratory data analysis **15**
3.1 Structure of the data . . . 15
3.2 Statistical analysis of the data . . . 17

4 Prediction of the redshift **23**
4.1 Selection of features . . . 24
4.2 Machine learining methods . . . 26
4.3 Metrics of model accuracy evaluation . . . 28
4.4 Splitting the data . . . 28
4.5 Results . . . 29
4.5.1 Multiple linear regression . . . 29
4.5.2 Random Forest . . . 31
4.5.3 Support vector machine . . . 32
4.5.4 Artificial neural networks . . . 33
4.5.5 Catboost . . . 35
4.6 Summary . . . 36

5 Classification of objects **39**
5.1 Selection of features . . . 40
5.2 Machine learning methods . . . 41
5.3 Metrics of model accuracy evaluation . . . 42
5.4 Results . . . 43
5.4.1 Multinomial logistic regression . . . 43
5.4.2 K–Nearest neigbours . . . 45
5.4.3 Random forest classifier . . . 48
5.4.4 Neural network . . . 50
5.4.5 Support vector machine . . . 52
5.4.6 Catboost . . . 55
5.5 Summary . . . 57

6 Unified models **59**
6.1 Properties of the problem 60
6.2 Results 62
6.2.1 Multi–output neural network 62
6.2.2 Models with joined methods 63
6.3 Summary 64

7 Summary **67**

Bibliography **69**

Chapter 1

Introduction

Astronomy is considered as one of the oldest known sciences, as since the ancient times people have been wondering what lies beyond Earth. By observing the night sky, various thoughts and perceptions appeared in the minds of philosophers, which later turned into the first theories and beliefs. As primitive as they were, their manifestation allowed to broaden the knowledge about the universe, which then have been passed down from generation to generation. As humanity was developing, different medieval discoveries allowed astronomers to build measuring apparatuses and direct them into space. The turning point was the discovery that the Earth is not in the centre of the universe and that most of the visible objects in the sky are stars similar to our sun. From that moment on, it was possible to thoroughly analyse observed objects and denote their properties. As civilization progressed further, the telescopes were built bigger and more powerful what allowed to observe more and more distant objects. All of them were described and assigned a different category, and at that time it was not problematic, even if every object had to be catalogued manually, due to their limited number. The biggest index of stars was created by John Flamsteed, who managed to label over 3000 stars. With the advent of the 20th century, knowledge about the universe developed exponentially. As a result, it led to the expansion in the field of astronomy and the new branch appeared which was called the observational astronomy. It was mainly concerned with recording data about the observable universe and was considered as an aftermath of the discovery of infrared and ultraviolet waves, which helped observers to capture light spectra of objects which were hidden behind dust impermeable to visible light. As observation data kept getting bigger, there emerged a need for their regularization and archiving. Fortunately, the invention of computers and virtual databases has satisfied this demand. Along with the astronomical and IT development, there was an evolution within the statistics, there were created numerous methods which allowed an automatic processing of different types of analysis, algorithms and primitive forecasting. Scientists realised, that the implementation of some of them would make their work easier. It began with the simple predictions and visualizations [42], but everything changed with deployment of the Hubble Space Telescope. It was able to capture immense amounts of data every night, so that the usage of statistical methods was not so much possible but necessary. Since then, many equally powerful telescopes have been built and placed all over the Earth, so that almost no visible object could escape being seen. Nowadays, without computers and statistics, astronomers wouldn't be able to accurately compute regularities occurring in the space. Their application along with machine learning allowed to broaden the consciousness about how big is the universe, and how little do we know about it. However, human curiosity has

no limits, and for this reason, many various theories have emerged, which thanks to various methods, can be confirmed or rejected. One of them was the Hubble's law, which says that the velocity of a remote object is proportional to its distance from Earth. For a long time, it hasn't been proved, because no one knew how to measure any of these parameters. The answer came as a result of applying another physical law called a Doppler effect. Due to the fact that the majority of distant objects are moving away from us, the length of light waves telescopes were detecting seemed to be elongated, what resulted in the visible shift to the red concerning their colour. Based on the magnitude of this shift, it was possible to calculate the approximate distance of mentioned objects, and thus their velocity. As a result, it was possible to catalogue them as the proper type, what made categorization much easier, as new kinds of objects with similar characteristics are being discovered all the time. The whole process involved was too tedious and complicated to carry it out manually, but the perfect solution was found within machine learning. Thanks to the methods it consists of, it is possible to analyse terabytes of data describing the observed objects and predict their characteristics based only on their captured electromagnetic waves. Even 50 years ago, such approach would seem quite impossible, but due to the rapid development of observational technology, we are able to deepen the knowledge about the outer space around us at an ever faster pace. For that reason we can find out on our own how the implementation of machine learning and statistics to the publicly available photometric data is carried out. As for the mentioned redshift, that parameter is an essential variable in most of the post–observational analysis, as it allows finding out the most about a given object and its properties. For that reason, in our thesis we will be trying to intertwine the two types of predictions, as we'll be using the regression methods implied on objects' data to foresee the values of redshift, as well as trying to classify a particular object to the corresponding class. At first, we will use separate models to perform these predictions, and then we'll try to create unified models which will simultaneously provide the desired variables. In order to accomplish that, we will use data from Sloan Digital Sky Survey (SDSS), which is regularly published by University of Chicago. Such problem and the approach to it seems to be really fascinating, especially considering that astronomy is a very broad branch of science so that we had to decide which aspects could be covered in the thesis. In addition to that, the possibility of applying mathematics to astronomy seems to be an extremely intriguing experience, and the observations and results of the work themselves can be somewhat useful for further analysis.

1.1 Related works

To begin with, there are many astronomical articles, books and scientific research on topics related to redshift prediction and classification of objects. Besides that, there are also some works that are based on the data derived from various releases of SDSS. In this chapter, we will try to introduce and briefly describe them.

One of the most prominent articles in this field dealt with the application of machine learning methods in astronomy for different kind of issues [3]. Authors proposed a couple of approaches to handle the most commonly encountered problems using the data from one of SDSS releases. It was considered as a big deal within the astronomical community and was considered as one of the foundations of the common use of machine learning. Since then, different scientists tried to take the advantage of what these datasets offered. There appeared works on the application of unsupervised methods to deal with the clustering of galaxies [7] [32], where the surface brightness profile and mass-to-light ratio were taken

into consideration. When it comes to the use of supervised learning, one of the most popular researches were related to the topic of star formation [10] [5], where scientists were using different light spectra to find out about the origin of stars and predict the places where new ones could form. With respect to the specific topics regarding the matter of redshift, there were published many papers either dealing with its prediction [48] [49] [11] or with its influence on observable universe [6] [4]. In addition to that, in many of them the SDSS data was used or mentioned so that we can confirm it as one of the most legitimate and favoured source of information. With regard to the classification of objects, it is a widely known problem across every astronomical society. Many authors tried to apply diverse methods in order for the results to be possibly the most accurate [50]. It happened with varying degree of success, but the room for improvement in that matter allowed the others to propose their solutions [38] [43] and gain recognition in the scientific community. There have been published many more papers related to these topics, but we managed to present the most significant ones which deal with the similar problems, that we are probably going to encounter during the thesis. Thanks to that, we will be able to find out how to manage different uncertainties and their content itself would outline for us a path of proceedings.

Chapter 2

Theoretical aspects

In this part of the thesis, we would like to introduce some of the issues that we will use in the following stages. As the topic connects two separate branches, mathematics and astronomy, we decided to split these theoretical aspects into two sections, each dealing with one of them. In astronomy part, we will focus more on the description of presented phenomena in order to understand their universal nature and characteristics, while for mathematical notions we would like to be more meticulous.

2.1 Astronomical aspects

Definition 2.1 (Star)**.** A star is an astronomical object consisting of a spheroid of plasma and gas, which is held together by its own gravity [30]. These objects are one of the most common celestial bodies in the observed universe. They produce light and electromagnetic waves due to thermonuclear fusion processes occurring in their core and radiation derived from the internal energy sources. Most of them are forming groups, which are attracted by mutual gravitation, and are called star clusters. The variety of star types is really large, these objects may differ in sizes, energetics, temperatures, masses, and chemical compositions, as there are no two identical stars across the universe. Many of them can be seen on the night sky by a naked eye, but for any details regarding their properties, they need to be observed by telescopes. Stars can live up to billions of years and during their lifetime can be a part of planetary systems, just like the Solar System.

Definition 2.2 (Galaxy)**.** A galaxy is a gravitationally bound assemblage of various celestial objects such as stars, stellar remnants, interstellar gas, dust and dark matter orbiting around some centre of mass [45]. They are considered to be enormous structures with the diameter reaching thousands of light years, while the source of gravitational field bonding all objects in these formations is supposedly a supermassive black hole. Galaxies, similarly as stars can form different gravitationally dependent clusters, called filaments, containing hundreds of them. The existence of galaxies was not discovered until the early 20th century, and since then, they have become one of the most puzzling structures in astronomical research. There are considered three different types of galaxies divided according to their shape: elliptical, spiral and irregular, and each can contain approximately 10^{11} objects.

Definition 2.3 (Quasar)**.** Quasar – abbreviation of quasi-stellar radio source, it is a highly luminous active galactic nucleus (AGN) powered by gas spiralling at high velocity around a supermassive black hole. These objects are so bright, that they can outshine all of the

stars in the galaxies in which they reside, and even the whole galaxies. They are the most distant objects, whose light waves could be captured by telescopes and happen to exist in places with plentiful gas supplies, as the main part of their energy comes from heating up in the process and emitting radiation across the electromagnetic spectrum [15]. In fact, they are not as common as previously mentioned objects, their discovered population is at about one million with the nearest known one being about 600 million light–years away from Earth. Despite their distance, some of them, whose luminosity is thousands of times greater than the one of Milky Way, can be visible on the night sky with the help of an amateur telescope.

Definition 2.4 (Apparent magnitude)**.** The measure of an astronomical object's brightness observed from Earth is called the apparent magnitude. It depends on the luminosity, distance and size, while its scale is reverse logarithmic, what means that the smaller the value of apparent magnitude, the brighter the object appears [37]. Unlike the absolute magnitude, which represents the intrinsic luminosity of celestial bodies, most of the measurements take values which are smaller than 6 and therefore invisible to the naked eye. That is why the night sky seems to be black and empty rather than full of bright spots. The branch of science that deals with apparent magnitude measurements is called photometry and the surveys are made in the ultraviolet, visible, or infrared wavelength bands using common passband filters.

Definition 2.5 (Redshift)**.** Redshift (often denoted as z) is a spectral shift of an astronomical object's emitted light–waves toward longer wavelengths. It's a result either of a Doppler effect, where the source of waves is moving away from the spectator, or of the expansion of space. It's observed especially for distant objects, whose velocity according to Hubble's law is much bigger than for the objects closer to Earth [14]. That concept is really useful in astronomy, as it allows to calculate distance between objects, and helps to recognize different kinds of celestial bodies. Due to this phenomenon, the majority of observed objects on the night sky appear to be seen with a red tint, meaning that they are moving away from Earth. Such discovery was an irrefutable evidence of the expansion of the universe and helped scientists to better understand its principles.

2.2 Mathematical theory

2.2.1 Machine learning algorithms

Before we head to the algorithms themselves, we should primarily focus on the description of learning problems. The ones used in the thesis belong to the group called supervised learning. It describes the problem class, which involves using a model to learn the mapping between input examples and the target variable. In this approach, models are fitted to training data consisting of inputs and outputs, and are used for forecasting on test sets where only inputs are provided, and the model outputs are compared with held target values and used to estimate the model's performance. The two main types of supervised learning problems are classification and regression that involve predicting a class label and a numerical value, respectively. On the other hand, there are some methods, which appear during the main part of the thesis, that are associated with unsupervised learning. In comparison to the supervised one, unsupervised learning deals only with the input data without target variables or outputs. In particular, it depicts a class of problems that

involves using a model to describe or extract relationships in data without relying on the training and testing procedures.

The methods we are going to apply, in order to solve the posed problems, deal with the classification and regression so that the main part of the thesis is connected with the supervised learning. Let us now introduce these algorithms, their characteristics and way of proceeding. We can divide them into three groups: methods used only for regression, methods used only for classification and methods that can be applied to both types of problems.

Regression methods

Definition 2.6 (Linear regression)**.** Linear regression is a collection of methods based on linear combinations of explanatory variables and parameters adjusting the model to the data. The fitted regression line or curve represents the estimated expected value of y given specific values of one variable x in case of classical regression, or variables x_i, which is then referred to as a multiple linear regression [36]. In particular, if there are n explanatory variables and m observations, the model can be of form:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in} + \epsilon_i, \qquad i = 1, \ldots, m, \tag{2.1}$$

where ϵ_i, is a random variable that represents the error. For the estimation, the most often used method is the classical method of least squares and its derivatives. Linear regression is the oldest and the easiest to apply among machine learning methods, although it has disadvantages (e.g. low resistance to outliers), which have been taken care of in other, less popularized methods. Besides that, linear regression analysis can be used to quantify the level of the relationship between the response and the explanatory variables, and to determine whether some variables have no clear linear relationship with the response, or to find out which subsets of explanatory parameters can contain redundant information about the response.

Classification methods

Definition 2.7 (Logistic regression)**.** Logistic regression is one of the regression methods used in statistics when the dependent variable is binary. Usually, the values of the response variable indicate the presence or absence of a certain event that we want to forecast. Logistic regression then allows the probability of this event to be calculated [16]. Formally, the logistic regression model is a part of General Linear Models family, where *logit* was used as a linking function. The model itself is not regarded as a classifier, but can be used to create one, for example by selecting the cut–off value and classifying the input data with the greater probability as one class, and the data with smaller probability as the second. The model uses a form of logistic function which is described as:

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}, \tag{2.2}$$

where β_0 is known as the intercept and β_1 as the inverse scale parameter. The parameters of a logistic regression are often estimated by maximum–likelihood estimation, and one of this method's assumptions is that the variables that the data consist of must be independent of each other. In our thesis, we will more likely be using the multinomial logistic regression, which deals with the case of a categorical dependent variable with

more than two possible outcomes. That type of regression can be a particular solution to the classification problems, which consist of the linear combination of variables and some problem–specific parameters to estimate the probability of each value of the dependent attribute. The optimal parameters' values are determined by training the model, and tested using the separate set of observations.

Definition 2.8 (K–Nearest neighbours)**.** K–Nearest neighbours algorithm is one of the non–parametric regression algorithms used in statistics to predict the value of a random variable or in classification problems. This method is especially useful when the relationship between the explanatory and explained variables is complex or unusual [2]. Its use for classification is based on the assigning to each data item the certain set of n values that characterize it, and then placing this item in n–dimensional space. To assign an observation to one of the existing groups, we need to find k nearest objects in the mentioned space and then select the most numerous group. K–NN is a type of algorithm, that in order to classify observations relies on the distance, so if variables represent different physical units or come in diverse numerical scales, then normalizing the data can significantly improve its accuracy. When it comes to the measured distance, different metrics can be used for different cases, with the most popular being Euclidean and Hamming distances. While implementing this method, it is also crucial to choose the k parameter properly. It generally depends on the data, while overall bigger values of this variable can decrease the effect of noise on the classification, but make the boundaries between classes less obvious.

Methods for both classification and regression

Definition 2.9 (Random forest)**.** Random forest is a machine learning method for classification, regression and other tasks, which consists of constructing multiple decision trees during training and generating a class that is the dominant of classes for classification or the predicted mean of individual trees for regression. A random forest eradicates the limitations of a decision tree algorithm, as well as reduces the overfitting of datasets and increases precision. The key of the algorithm's performance is low correlation between models because together they can produce ensemble predictions that are more accurate than any of the individual ones [18]. It also allows measuring the relative importance of each feature on the prediction by indicating how much the tree nodes that use that feature reduce impurity across all trees in the forest. Random forests are regarded as one of the most utile methods that despite their simplicity can provide very accurate predictions. The most crucial parameter of this algorithm is the number of base learners because as we increase its value the variance decreases, and the other way around, but the bias is constant. Typically this value can be found using cross–validation methods.

Definition 2.10 (Artificial neural network)**.** A neural network is a system designed to process information, the structure and principle of operation of which are to some extent modelled on the functioning of fragments of the real nervous system. Such network is based on a collection of connected units or nodes called artificial neurons, which reside in structures called layers. Each of them is connected to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network while the output of each neuron is computed by some non-linear function of the sum of its inputs. Signals travel from the input layer, to the output layer, possibly after moving through the layers multiple times. During this process of training, the weights are adjusted a couple

of times so that the error of output is successively smaller with each iteration. After an undetermined number of these adjustments the process can be terminated based upon certain criteria, then the network is suitable for prediction problems. The specific feature of the neural network is the possibility of computer–based solving of different problems without their prior mathematical formalization as well as the ability to learn from examples and automatic generalization of gained knowledge [40]. For that reason, they have a lot of scientific applications, particularly in pattern recognition and classification.

Definition 2.11 (Support vector machine)**.** Support vector machine is an abstract concept of a machine that acts as a prediction tool, the learning of which is aimed at determining a hyperplane separating examples belonging to two classes with a maximum margin. Being developed by Vladimir Vapnik in early 90s, SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. The mentioned hyperplanes are built in an n–dimensional space in a way to maximise the width of the gap between each category [13]. The algorithm chooses on its own the extreme points that allow to construct such plane, which are often referred to as support vectors. The effectiveness of SVM depends on the selection of kernel and hyperplane parameters, which are usually derived by solving the arising optimization problem. Their special property is that they can simultaneously minimize the classification error and maximise the geometric margin so that they are also known as maximum margin classifiers. When it comes to the multi–class problem, the most common approach is to use one–versus–rest method, which involves splitting the multi–class dataset into multiple binary classification problems.

Definition 2.12 (Catboost)**.** Catboost is a machine learning algorithm for gradient boosting on decision trees. It builds symmetrical trees and at each step the leaves from the previous tree are split according to the same condition. The pair of split and feature that account for the lowest loss is extracted and used for all level nodes. This balanced tree architecture helps to efficiently implement the CPU, reduces prediction time, ensures smooth model applications, and controls overfitting as the structure serves as regularization [35]. In addition to this, catboost uses the concept of an ordered boosting, a permutation–based approach to train a model on a data subset while computing residuals in another subset, thus preventing target overfitting. It also offers significant performance potential, as it works exceptionally well with the default parameters, greatly improving the results after tuning.

2.2.2 Metrics of model evaluation

In this section, we will introduce the statistical metrics that will allow us to evaluate the performance of created machine learning models. Due to the fact that the assessment of regression and classification models cannot be made in the same way, different metrics are used for these kinds of problems. We will divide them into two separate groups: the metrics for regression models and the metrics for classification ones.

Metrics for the evaluation of regression models

Definition 2.13 (Mean absolute error)**.** Mean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Usually used for assessing the performance of regression models, it provides arithmetic average of absolute

errors $|e_i| = |y_i - x_i|$, where y_i is the prediction and x_i the true value [39]. Then the MAE is calculated as follows:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}. \tag{2.3}$$

Overally, the smaller the error for the model, the better is its performance.

Definition 2.14 (Mean squared error)**.** Mean squared error (MSE) measures the average of the squares of the errors, in other words — the average squared difference between the estimated values and the actual value. The MSE is the second moment of the error, and thus incorporates both the variance of the estimator and its bias. It can assess the quality of a predictor as well as an estimator. For the purposes of prediction, its form is presented as:

$$\text{MSE} = \frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n}. \tag{2.4}$$

Similarly as before, the smaller the error, the better is the prediction.

Definition 2.15 (Coefficient of determination)**.** Coefficient of determination (R^2) is one of the metrics assessing the quality of the model fit to the training data. It provides a measure of how well observed outcomes are replicated by the model based on the proportion of total variation of outcomes explained by the model [44]. The fit is the better, the closer the value of R^2 is to one. In regression, this coefficient is a statistical measure of how well the regression predictions approximate the real data points. Basically it is expressed as:

$$R^2 := \frac{\sum\limits_{i=1}^{n} (\hat{y}_i - \overline{y})^2}{\sum\limits_{i=1}^{n} (y_i - \overline{y})^2} \geqslant 0, \tag{2.5}$$

where y_i is the predicted value, $\hat{y}_i$ is the actual value of a variable and $\overline{y}$ is an arithmetic mean of empirical values of the dependent variable.

Metrics for the evaluation of classification models

Definition 2.16 (Confusion matrix)**.** Confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. That way, we are able to analyse how many observations were properly classified and check how many predictions were inaccurate [34]. Calculating a confusion matrix can give a better idea of what a classification model is getting right and what types of errors it is making. There exists a large number of measures that are derived from the matrix which can provide the model's performance assessment, such as true positive rate, precision, recall, etc..

Definition 2.17 (Accuracy)**.** Accuracy is a measure describing the degree of compliance of the actual value with the arithmetic mean of the results obtained for the determined quantity. The more accurate the measurement method, the closer the results are to the true value. It is also used as a statistical measure of how well a classification test correctly identifies or excludes a condition. It shows the number of classifications a model correctly predicts divided by the total number of predictions made. Its value ranges between 0 and 1, and the higher the value, the more accurate the results are [26].

Definition 2.18 (Precision)**.** Precision is a performance metric that generally applies to classification assessment. It describes the fraction of relevant instances among the retrieved instances [34]. In binary problems, it is calculated as the number of true positives divided by the total number of true positives and false positives:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{2.6}$$

The result also varies between 0 for no precision and 1 for perfect precision. In an imbalanced classification problem with more than two classes, precision is calculated as the sum of true positives across all classes divided by the sum of true positives and false positives across all classes.

Definition 2.19 (Recall)**.** Recall is regarded as one of the classifying model performance metrics, which describes the fraction of relevant instances that were retrieved [34]. In a typical classification problem with two classes, recall is calculated as the number of true positives divided by the total number of true positives and false negatives:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{2.7}$$

Its value comes from a range from 0 to 1, where the bigger the result, the better the method's performance. In an imbalanced classification problem with more than two classes, recall is calculated as the sum of true positives across all classes divided by the sum of true positives and false negatives across all classes.

Definition 2.20 (F–1 score)**.** F–1 score is a metric of model's performance that combines precision and recall as their harmonic mean. It is considered as a more general measure, taking into account the two other metrics [34]. It is expressed as:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.8}$$

Similarly to other measures, its value varies between 0 and 1, where the values closer to 1 indicate better performance.

Definition 2.21 (Area under the ROC curve)**.** Area under the ROC curve (AUC) is one of the metrics of a model's performance evaluation. It is used for measuring the ability of a classifier to distinguish between classes. AUC varies between 0.5 and 1 — with an uninformative classifier yielding 0.5 and the higher is the value, the better is the performance of the model at distinguishing between the positive and negative classes. It is a robust overall measure to evaluate the performance of score classifiers because its calculation relies on the complete ROC curve and thus involves all possible classification thresholds [34].

2.2.3 Other definitions

Definition 2.22 (Principal component analysis)**.** Principal component analysis (PCA) is the process of computing the principal components and using them to perform a change of basis on the data, where the first principal component of a set of p variables, presumed to be normally distributed, is the derived variable formed as a linear combination of the original variables that explains the most variance. Each subsequent component explains

the most variance inthe remnants of the process [22]. In this way, a new observation space is constructed, in which the most variability is explained by the initial factors. PCA is often used to reduce the size of a statistical dataset by discarding the least significant factors, as well as for creating predictive models on uncorrelated variables with no loss of information from the original dataset.

Definition 2.23 (Cross–validation)**.** Cross–validation is a statistical method of dividing a statistical sample into subsets, and then carrying out any analysis on the so–called training set, while the others are used to confirm the reliability of its results, the so–called test set. It is mainly used in analysis aiming at prediction, and estimation of how accurate would be a predictive model in practice. The purpose of cross–validation is to not only test the model's ability to predict new data that was not used to estimate it, but also to mark problems such as overfitting or selection bias and to give insight into how the model generalizes to an independent dataset [34]. Cross–validation is implemented by dividing the data sample into subsets, performing the analysis on one, and validating the analysis on the other. To reduce variability, most methods perform multiple rounds with different partitions, and the results are averaged over the iterations to obtain an estimate of the model's predictive performance. There are a couple of types of this algorithm, with the most popular being the k–fold cross–validation and leave–one–out cross–validation.

Definition 2.24 (Normalization)**.** In statistics, data normalization is the process of putting different variables on the same scale and thus having the same importance. This process allows comparing scores between different types of attributes and improves the performance and training stability of the predictive model. It may also refer to more sophisticated adjustments where the intention is to bring the entire probability distributions of adjusted values into alignment. Usually, as the data is rescaled by applying a particular formula, each value is placed in a range between 0 and 1, so that the influence of each variable is the same. Such process is extremely useful when instead of the intervals, ratios of measurements are meaningful.

Definition 2.25 (One hot encoding)**.** One hot encoding is a process of converting categorical data variables by mapping them into integers, so they can be provided to machine learning algorithms to improve predictions. With this method, we convert each categorical value into a new categorical column and assign a binary value of 1 or 0 to those columns. Each integer value is represented as a binary vector and while all the values are zero, the index is marked with a 1. Such columns in the data are also called dummy variables. One hot encoding makes the training data more useful and expressive, and it can be rescaled easily. Besides that, some machine learning methods do not cope with categorical data and in order not to lose the information encoded in these variables, such approach is being applied. On the other hand, if the amount of data to process is too large it may produce multicollinearity among the various variables, lowering the model's accuracy.

Chapter 3

Exploratory data analysis

In this part, we would like to focus on presenting and understanding the data that we will have to deal with during the analysis. It is important to be aware of what factors the data is composed of and if there are any interactions between them. We would also like to find any unusual or missing observations and deal with them in order for the data to be clean and transparent. Thanks to the statistical insights, we will be able to simplify the data and find out the structure of the whole data set and of each of the parameters separately.

3.1 Structure of the data

The dataset which we will be using to perform the desired experiments and analysis can be found on the Kaggle website [1]. It contains photometric information about 100 000 astronomical objects that was collected by Sloan Digital Sky Survey (SDSS) in the 17th Data Release published in January 2022 [23]. The telescopes that were able to capture the light waves are placed in the USA and Chile and are one of the biggest instruments of this kind in the world. The purpose of that mission was to scan as deeply as possible every part of the sky and gather the data in the form of photographies. Every view containing any astronomical object was analysed and its features extracted. Usually, astronomical publications are of enormous size, so that the whole SDSS report includes over 222 TB of data. Despite that, it was possible to take out the most essential components and create the mentioned set. Each of these observations consists of 19 features describing its most important characteristics and the way they were observed. For better understanding of the data, we can present the first couple of entries as a table on Figure 3.1.

	objid	ra	dec	u	g	r	i	z	run	rerun	camcol	field	specobjid	class	redshift	plate	mjd	fiberid	velocity
0	1237645879551000764	348.841087	1.268802	25.92735	20.99570	19.24612	18.56461	18.43049	94	301	6	93	4825695603672766464	GALAXY	0.399661	4286	55499	322	0.066242
1	1237645879551066262	348.902530	1.271886	19.38905	18.24496	17.58728	17.20807	16.90905	94	301	6	94	430194949951088640	GALAXY	0.032125	382	51816	368	0.002570
2	1237645879562862699	15.896126	1.264845	20.14134	19.28787	19.04397	18.96897	18.79573	94	301	6	274	445958098402699264	STAR	-0.000821	396	51816	370	0.003667
3	1237645879562928144	16.004912	1.259423	21.50923	19.69340	18.47973	17.91998	17.55132	94	301	6	275	754455784200366080	GALAXY	0.312048	670	52520	374	0.005356
4	1237645879562928258	16.020244	1.267667	20.96947	20.29136	19.36779	18.86387	18.45551	94	301	6	275	754453860055017472	GALAXY	0.200468	670	52520	367	0.007651
5	1237645879562928805	16.026029	1.266772	25.27165	22.32081	21.33033	19.94852	19.47201	94	301	6	275	4853831831467087872	GALAXY	0.752731	4311	55506	281	0.015532

Figure 3.1: Table presenting the exemplary entries

Then it is crucial to explain the meaning of some of the parameters:

- **objid** — astronomical ID number of the observed object,
- **ra, dec** — angles of right ascension and declination — they are the astronomical coordinates that define the position of an object on the celestial sphere in the equatorial coordinate system,
- **u, g, r, i, z** — apparent magnitudes of the object, whose absorbed light was passed through various filters in photometric system. Each parameter represents a magnitude for a single one: **u** — ultraviolet filter, **g** — green light filter, **r** — red light filter, **i** — near infrared filter, **z** — infrared filter. These parameters are the most important ones, as they primarily define the chracteristics of an observed object,
- **run, rereun, field, specobjid** — identification numbers used to determine the specific part of sky scan and photo in which the particular object exists,
- **class** — categorical parameter that assigns the object one of three possible classes: galaxy, star or quasar,
- **redshift** — the photometric redshift of observed objects — indicates the increase in the length of emitted electromagnetic waves that reach the lenses,
- **plate, fiberid, camcol** — these factors describe the part of the telescope that spotted the particular object,
- **mjd** — Modified Julian Date, it is used to indicate when a given piece of SDSS data was taken,
- **velocity** — the approximate velocity dispersion of object.

In the data, there appear many numerical attributes that have to be considered as categorical variables because they represent the discrete values describing the particular telescope and the scan of the sky that captured an object. Apart from that, we can find out that the *run* and *rerun* parameters are redundant because for every row they take the same value. For this reason, we can drop them from our data. As a result in the dataset there remain 17 attributes out of which 8 are categorical and 9 continuous.

Next, we want to examine if there are any missing values in the data, and if there are some, we need to find out how they are denoted. It turns out that if there is a value equal to -9999.0, it means that the data was not gathered for this object [23]. Fortunately, there is only a small amount of such observations, therefore we can drop them from our dataset and proceed forward with 99 886 rows.

The next step will be handling the outliers. By definition, an outlier is a data point that is significantly different from other observations. In astronomy, it is extremely difficult to point out such deviation, as the diversity of the universe is truly enormous. No person can certainly claim that a particular object was incorrectly observed because the human race has so far known only a small part of the visible outer space. Due to that, we will take into account any appearing observation, considering it to be coming from a legitimate source and treat it similarly as every other. Such approach was taken, so that the models that we are going to build may be affected simultaneously by each object in the same way because in astronomy each one is equally important. The analysis should be conducted in a way that the results and conclusions could be applied to any observation, not only for those that fit the criteria of normality.

3.2 Statistical analysis of the data

First and foremost, we would like to check some of the statistical indicators in order to have a glimpse at the characteristics of the continuous attributes of the data. In the table on Figure 3.2 there are presented such statistics as the mean value, standard deviation, minimum, maximum and quartiles.

	ra	dec	u	g	r	i	z	redshift	velocity
mean	159.480019	12.790664	21.711688	20.161783	19.170447	18.632516	18.326497	0.463140	0.010046
std	59.729920	22.160536	2.310282	2.081673	1.847839	1.735493	1.734522	0.718084	0.020258
min	5.602667	-8.479532	11.960910	11.696170	11.277090	11.051390	10.616260	-0.010932	0.000000
25%	128.818839	-0.539464	19.863263	18.397170	17.586398	17.184025	16.912198	0.000431	0.002496
50%	167.062655	0.397702	21.757840	20.544225	19.527565	18.952515	18.590040	0.218945	0.005111
75%	201.496894	14.870773	23.463100	21.859318	20.684183	19.873667	19.494080	0.562615	0.011237
max	359.064559	68.102394	29.807250	28.075160	29.839250	28.179630	28.818630	7.021413	1.836817

Figure 3.2: Table presenting the basic statistics

We observe that for the majority of the attributes the mean is less than the median, meaning that the data is skewed to the left. The other thing that seems to be conspicuous is the fact that there are objects for which the redshift takes negative value. In fact, it is considered to be extraordinary because it would mean that these objects are approaching to the Earth, unlike the great majority of them that follow the principle rules of inflation theory. Also, the mean of the declination of observed objects is relatively higher than 0, so we might assume that the observations aren't evenly distributed across the sky. For other attributes, we cannot notice anything unusual.

Right now, we can analyse the characteristics of particular variables. As it was mentioned before, in the data there are almost 100000 photometric observations of astronomical objects that are divided into three classes – galaxies, stars and quasars. At first, we would like to check how many observations fall under each of these categories.

Figure 3.3: Figure presenting the size of classes

The Figure 3.3 shows that the data is slightly imbalanced, with the majority of the observations assumed to be galaxies. Despite that, there are too many observations for this issue to be problematic during the further analysis, so we will not deal with this kind of problem. Next, having the locations of objects on the sky, we can project a sample of them onto the map with Aitoff projection in order to find out their placement distribution.

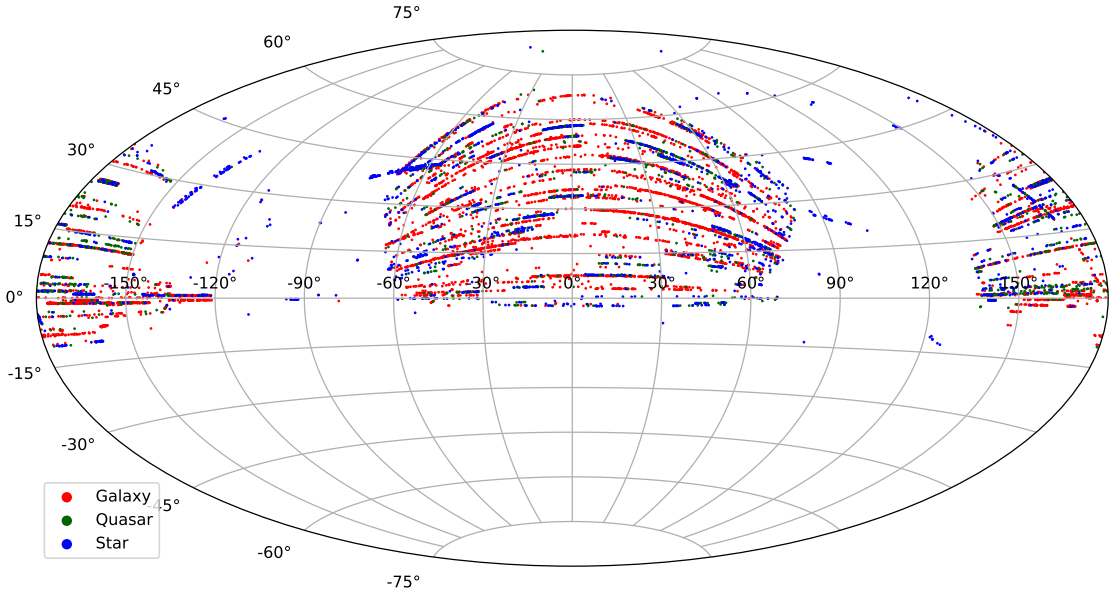

Figure 3.4: Figure presenting the location of objects on the firmament

On Figure 3.4 we can see a sample of 10% of all observations superimposed on the sky map. It appears that they are distributed in a shape of stripes across the whole firmament. It might seem that the same objects are close to each other, but it is caused by the telescope focusing on the one kind of objects during a couple of nights where the Earth's axial precession does not cause any disturbances. As the Earth revolves around, the telescopes don't change their angle of view, what results in curved lines of coordinates.

Next thing that we would like to analyse are the distributions of values across numerical variables. It would allow us to find out some details about their structure and examine the range of values they can take. In order to visualise that, we created the histograms together with probability density functions, which can be found on Figure 3.5. It appears that for each of the magnitudes, the values are in a similar range varying from about 10 M to 30 M. Each of the magnitude distributions is multimodal, with visible local maxima in density functions. It is noticeable especially for filters absorbing longer light waves as they have as much as three such maxima. A similar situation can be observed for the redshift histogram, as the majority of observations take values between 0 and 1 with apparent bimodal distribution.

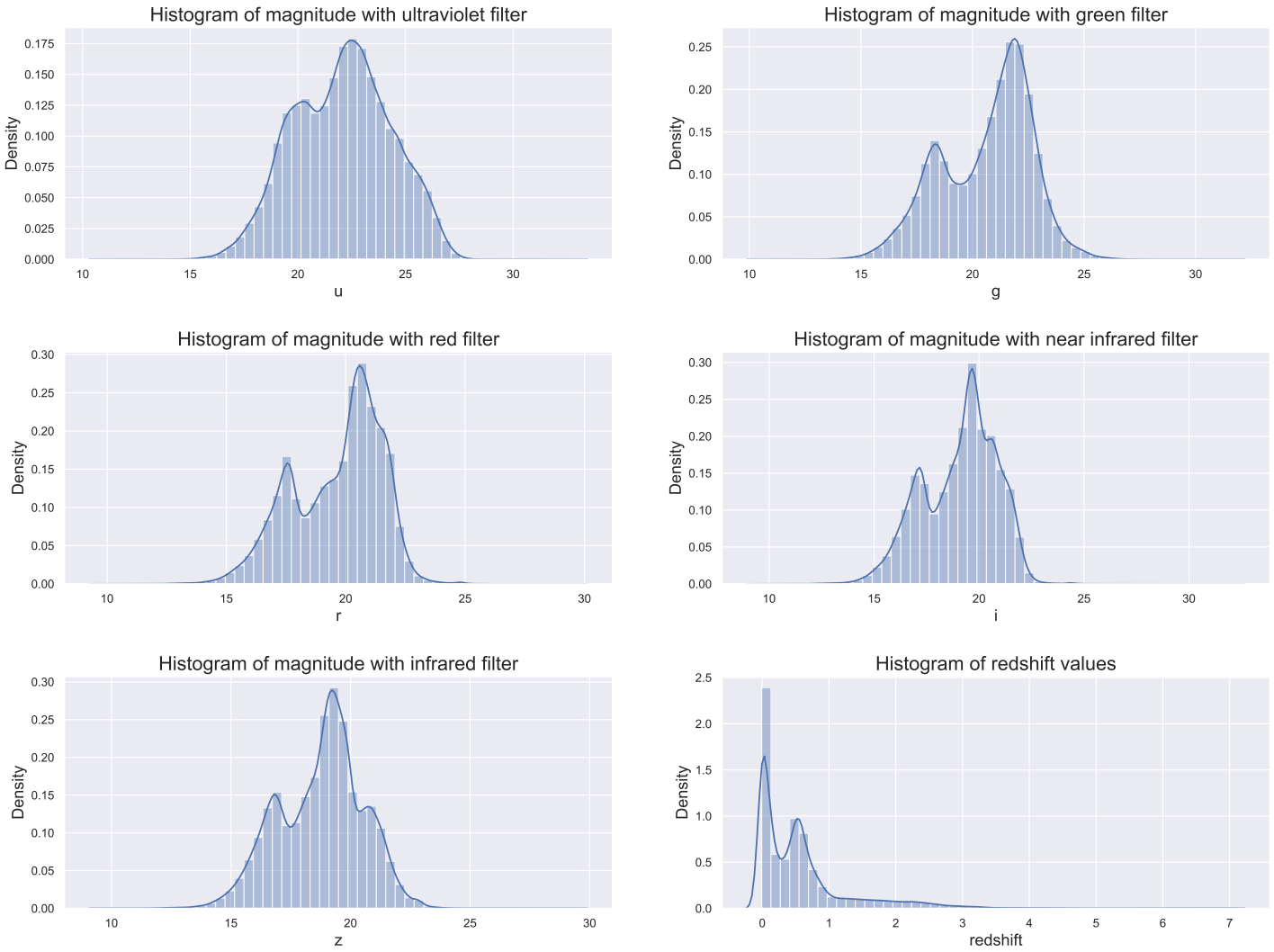

Figure 3.5: Figure presenting the histograms with density functions

The details of these visualisations can give us a hint that it may be caused by different mean values for different classes of objects. In fact, we are able to investigate that in the similar fashion. If we take into consideration separate probability density functions, we will be capable of determining the independent distributions for each class that the data consists of. Besides that, the analysis of them would enable us to understand the differences and similarities between them. Such graphs presented on Figure 3.6 can help us understand the details of these observations and may be essential in the further part of the thesis.

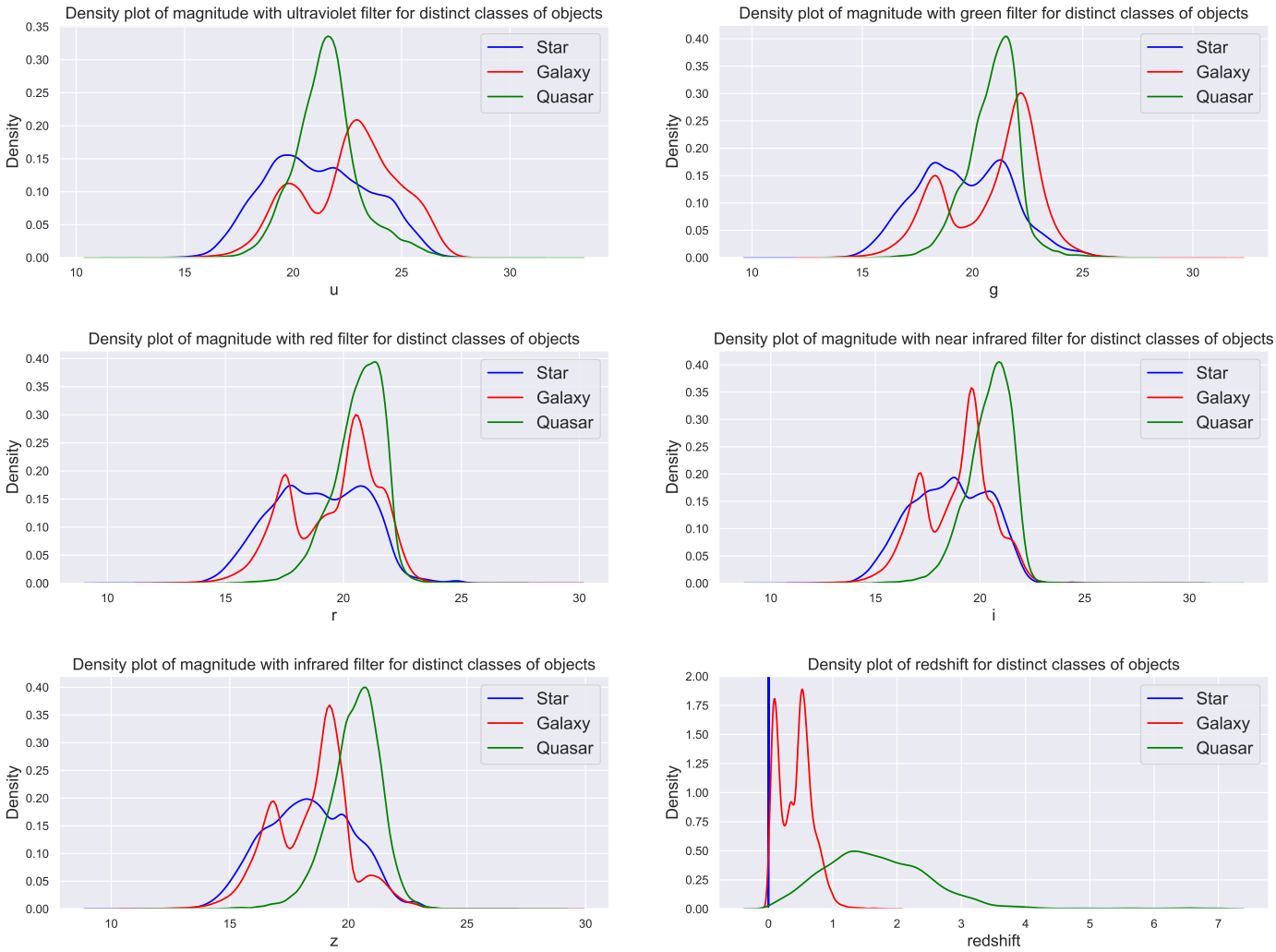

Figure 3.6: Figure presenting probability density functions for each class of objects

It turns out that, the distributions are quite different from each other. For the magnitude cases, we can notice that the range of values for quasars is narrower than for the remaining classes. The distributions of galaxies are multimodal in all of the graphs, what seems to be quite unusual regarding their apparent similarity. If it comes to stars, the distribution of magnitude most closely resembles the normal distribution which the most commonly used in modelling naturally occurring phenomena. What is the most standing out is the fact that across each magnitude filter, the distributions for quasars do not vary at all, including the shape and the point of local maximum. It may be caused by the unification of the absorbed light waves due to the immense distance between them and the Earth. Regarding the redshifts' distribution, the situation is quite clear. Its values for stars are relatively small in comparison to the other objects, however it seems to be logical, as they are relatively close to the observer.

Another statistical issue that needs to be taken into consideration is the correlation of numerical attributes. It is important to check whether some of them are interrelated linearly because it can have an effect on the conducted further analysis. The Figure 3.7 presents the heat map of correlation matrix between the variables. At a first glance, we can see a strong positive connection between the magnitudes. It should not seem peculiar, as they all have the same origin in the electromagnetic waves emitted by objects under study.

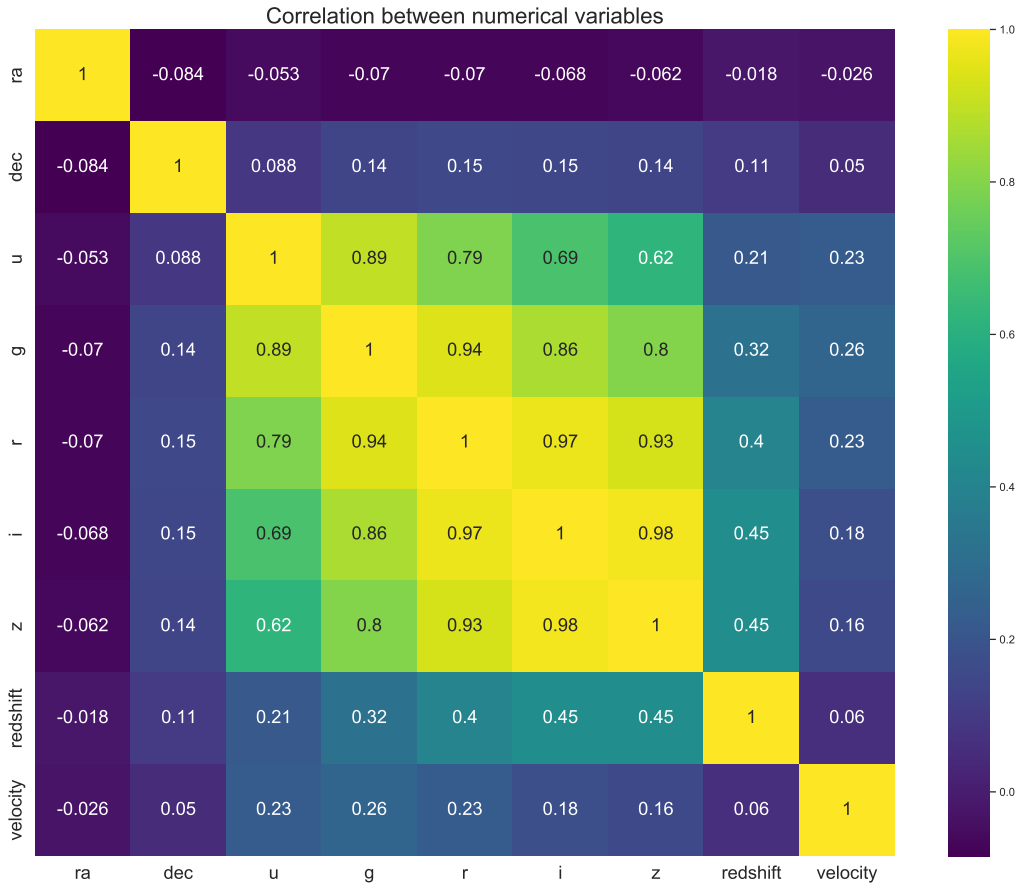

Figure 3.7: Heat map of correlation matrix

Apart from the magnitudes, the rest of attributes are not correlated at all or the correlation is negligible, especially when it comes to the location coordinates. In the next parts of the thesis we will focus to a greater extent on the correlation of the variables, what comes out from this matter and whether it is possible to reduce the correlation without any loss of information. At this point, we concluded all the most important information about our data, including its structure, size, origin and various statistics. We managed to get to know all the details that can be found inside, as well as analyse the relations between the variables. Such exploration of the data might come in handy while in next chapters we will deal with the thesis' main aspects and problems.

Chapter 4

Prediction of the redshift

The previous analysis of the variables gave us some insight into the data and allowed to find out different dependencies. During this part, we will try to make a use of them in the prediction problems. Some of them are contained in one of the most important questions that causes many sleepless nights to scientists, namely how to accurately estimate the distance of various celestial bodies basing only on the photographies and information captured by the ground–based telescopes. The solution to that problem could help understand many open questions dealing with large–scale structures and issues, such as evolution or galaxy formations. The parameter that comes in handy in this kind of situations is redshift, also known as photometric redshift. Thanks to the developed cosmological rules, knowing the value of this factor, we are able to estimate such distances quite accurately. On the other hand, many methods trying to calculate redshift were using expensive and inefficient spectroscopy. This was the foundation for the development of other approaches with the involvement of machine learning. It resulted in the possibility to pursue a precision in the cosmological calculations by combining different tools and deep photometry for big number of objects with an incomplete spectroscopic familiarity [9]. Each of the objects is supposed to have different photometric signatures, but mapping them into the computation of the redshift parameter is slightly complex. Primarily, the images captured by telescopes need to be processed in order to extract the crucial data, then machine learning tools and data–driven methods are applied for the estimation. Before that, the redshift had to be calculated using previously mentioned spectroscopic methods, which in the long run, with more and more data flowing in, no longer made sense. This introduced process needed to be automated while utilizing regression tools that allowed to predict this factor using the photometric variables. Such solutions might become essential for handling extensive tasks dealing with astronomical data processing and analysis. That is why in this part we will try to implement some of the supervised machine learning methods to deal with prediction problem. Naturally, this is one of the regression problems, as the redshift is a continuous variable. Before that, we will need to introduce a couple of issues that would make the prediction viable. By creating different models using feature selection, we will be able to analyse which variables are the most influential and which method performs the best. Obviously, we would like to achieve as good results as possible so that we will assess the outcome using various regression metrics.

4.1 Selection of features

For the purpose of the results to be non–homogeneous and meaningful, we have decided to use five different subsets of features. To select them, at first, we would like to focus on attributes that are not supposed to affect the regression in any way and try to remove them. In particular, any IDs of objects are of no use in the prediction process, therefore factors *fiberid*, *objid* and *specobjid* are redundant. The date of observation also shouldn't have any influence on the results, so we can drop *mjd* from the data set. It's also crucial to treat $redshift$ attribute as a target for the prediction, so obviously we will not take it into consideration while choosing the parameters. For the remaining categorical attributes we will try to find a way to make a use of them implementing a method called one hot encoding [41]. This way we can preserve the information as "dummy variables". By those means, we have created the very first of our subsets, where all factors can contribute to the model, what makes it the most natural. The drawback of this approach is the creation of an excess number of columns, what can affect the computational efficiency and provoke unnecessary trouble.

In order to reduce the computation time and the overall variance, we can propose a second subset, where we take into consideration only numerical variables and remove any attributes that define the observational specifications. Such attitude to the matter will certainly be optimal, as there are not as many columns as before, but does not provide answers to all doubts because the correlation between a couple of parameters mentioned in the previous chapter is still high what can affect the performance.

Having a look at the Figure 3.7, we may claim, that one of the biggest issues is the strong connection of apparent magnitude variables manifested by a high correlation. If we were able to reduce it without losing any details, we would have an even greater set of uncorrelated variables, which in machine learning tasks is supposed to perform better. The way of proceeding with that type of problem was introduced in articles [28] [46] dealing with the analysis of quasars and galaxies variability based on photometric data. Authors persuade that it's possible to transform the variables in a following way: we take the magnitude attributes and use their differences from the adjacent bands instead. To be precise, rather than parameters u, g, r, i, z, we would like to consider $z-i$, $i-r$, $r-g$, $g-u$, $u-z$. This would very likely reduce the correlation at the cost of a higher variance of these variables. Their distribution right now would presumably resemble the exponential rather than Gaussian. Apart from that, in the transformation clearly no information is lost and due to the fact that any photometric errors are assumed to be independent of magnitude, it might be a great manner to obtain more accurate results. What comes up with that approach is not only the reduced correlation but also a possibility to distinguish between different types of the same class of object (e.g. morphological types of galaxies or types of stars in Morgan–Keenan system). The heat map of correlation matrix with the modified attributes can be seen on Figure 4.1. We can notice that the values of correlation have dropped to the level where it would not be considered as an insurmountable obstacle. That way, including only continuous variables,this collection may be regarded as a next subset of features.

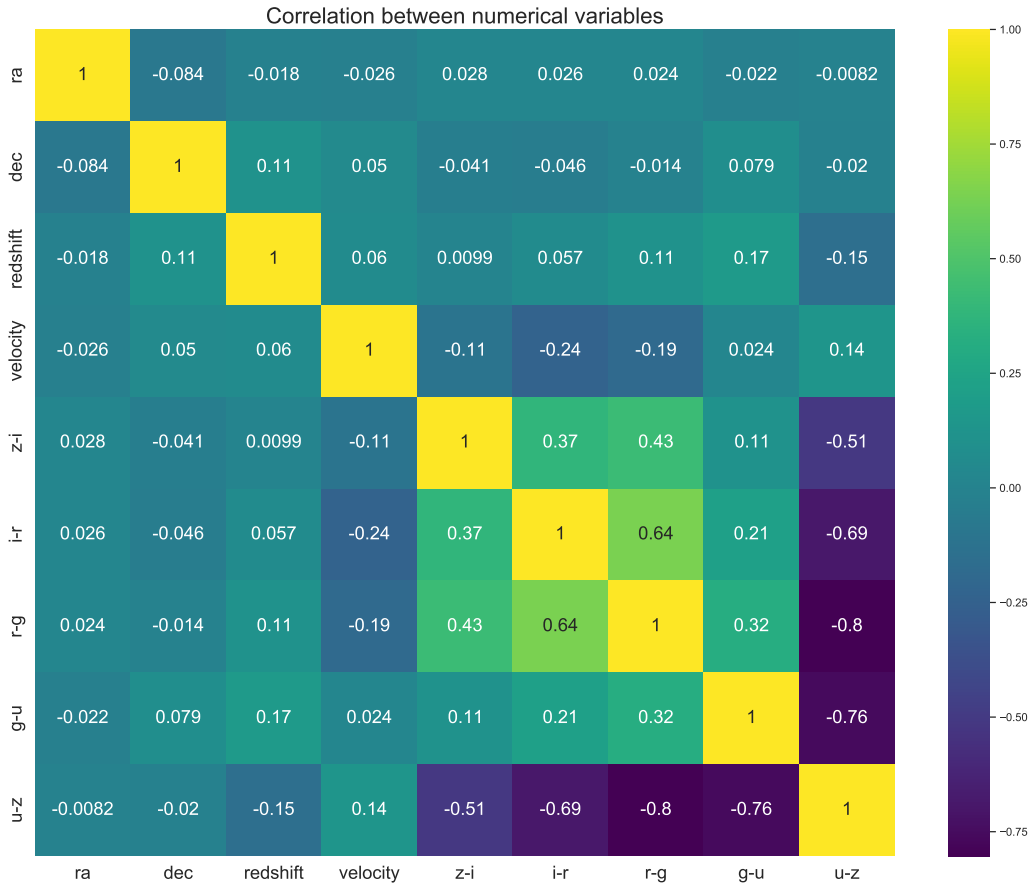

Figure 4.1: Heat map of correlation matrix with transformed attributes

The next subset used for machine learning methods will not be chosen manually. We wanted to check which choice of features would be the most optimal if such process was done by the machine. For this reason, we took advantage of the Python's function *SelectKBest* available in sklearn library. Its purpose is to select the k best variables for the model, where k is a parameter set by user. The function performs univariate linear regression tests that consist of two steps:

1. The cross correlation between each regressor and the target is calculated,

2. The result is converted to an F score and then to a p-value.

Having the scores for each parameter, the program can pick k best among them as an output. That is the way we are able to obtain the variables that will make up for our next subset. Let us assume that we want k to be equal to 5, so that the size of each subset is different, what can help us decide which one will perform the best. After imputing the data, the computer analysis returned the attributes that the fourth subset consists of — g, r, i, z and *velocity*.

Another automatic way to choose proper data for the analysis is through the use of some kind of feature extractor. In our thesis, we decided that the most suitable would be principal component analysis (PCA). Such approach was seen in different publications and has usually affected the prediction in positively [27]. This unsupervised method reduces the dimension and deals with multicollinearity transforming the data so that it can help us achieve better results. Implementing it and choosing $n = 5$ first components allows us to create the last of considered subsets. Finally, we have selected five subsets of features, which will be used for prediction of the redshift. To be precise, they are created as follows:

- first subset consists of all features, where categorical features are treated as "dummy variables", we will refer to it as a subset number 1,
- the second includes only numerical attributes for computational purposes, it will be named a subset number 2,
- third subset is formed by the transformation of magnitude variables into the difference of values between the adjacent bands, it will be regarded as subset number 3,
- fourth subset is established computationally by choosing the highest scoring variables in the conducted tests, it will be the subset number 4,
- last subset is a result of imposing PCA method on our dataset and selecting 5 principal components — subset 5.

As soon as we obtain the selected subsets, it is also useful to normalize them in order to avoid the influence of variables that take values from broader ranges. We do not want to deal with the situation when a couple of attributes would contribute more to the model just because of their higher magnitudes.

4.2 Machine learining methods

When it comes to the selection of methods used to predict the redshift, the amount of them to choose from is really overwhelming. The scientists from all over the world are using different techniques, and it is for us to decide which ones we will incorporate into the analysis. The first one we can consider is multiple linear regression. Generally, it is used to describe quantitatively a noticeable correlation between properties of a sample of objects. Before any of the calculations were done by the machine, it was the most popular tool in astronomy to predict any relationships resulting from different posted theories, as well as for reduction of observational errors. Nowadays, as the astronomical data deals more often with the intrinsic distribution of the independent variables, it allows implementing linear regression with a very good effect [20]. These models cope very well with problems regarding the large–scale structure of the universe, which the prediction of redshift is a part of. It might seem that it is a simple statistical procedure, but the complexity arises when it is utilized for variables that may not have Gaussian distribution or are heteroscedastic. Anyway, its implementation should be fairly suitable and the interpretation of its results ought not to be troublesome.

Another method that will help us in regression problem will be random forest algorithm. It is also regarded as one of the most known machine learning tools that by constructing many classical decision trees at training time, averages the prediction of each single one of

them, what reduces overfitting. Their advantage for astronomical data is that they can perform well with complex structures, and they don't impose any model on the underlying data [12]. In previous research, they were used to detect quasars and estimate the distances from objects. For our problem, the training algorithm creates a set of optimized decision trees on subsets of the obtained spectroscopic sample, which provide separate constraints on the redshift of each object. For this reason, the choice of this method is reasonable and justified. It historically has always provided quite accurate results, in most cases better than linear regression, and we wouldn't be surprised if that was the matter here too. The only drawback could be the computational complexity for some subsets of data, but if we were able to optimize its hyperparameters, it would not turn out problematic.

The third method frequently taken into consideration in that kind of tasks is support vector machine (SVMs). Being one of the most robust prediction methods, it is very well known for its efficiency for astronomical data, even though over the years its amount has increased in quality and quantity. It was used for mainly classification problems, but it had a couple of applications in regression, such as stars' effective temperature prediction or mass to light ratio estimation [51]. Its clear advantage is good generalization ability and excellent performance, as well as easy model parameter adjustment. Knowing that, we are able to effortlessly implement it in the regression model using the radial kernel. As for the drawbacks, this method barely has any, one of them being sensitivity to the noise, which is present in that type of photometric sets of data and might have been caused by vibrations in the Earth's atmosphere.

The next viable method for the photometric redshift prediction is artificial neural network (ANN). It is assumed to be the most common tool used in astronomy. Due to its universal nature, it can be utilized in practically any situation. For the supervised category, having a priori and accurate knowledge of the desired property, the networks consisting of neurons and layers can easily be trained by adjusting their weighted associations, so they fit perfectly to the data used in the thesis. Their previous applications in the field of astronomy included prediction of solar activity and phenomena or the study of the interplanetary magnetic field [47]. Thanks to their versatility, they have a truly great potential for further development and adoption in any tasks regarding the systematizations of the rules and patterns of the universe. These networks are indeed superior when it comes to the increase in quantity and the distributing complexity of astronomical data, but on the other hand ANNs are prone to overfitting on the training data, and by choosing suboptimal hyperparameters, the training process can take a long time. Nevertheless, the use of them for the introduced problem might give unsurpassed results when compared to other methods.

The last considered method is going to be the one most recently developed — catboost. Being a gradient boosting algorithm, it applies boosting method to create strong classifiers through learning multiple weak classifiers or regressors. It's widely used for different types of data due to the fact that it provides great results with default hyperparameters and is computationally efficient. In astronomy, there are a few applications of the method, especially in the field of prediction of quasar candidates and multiclass classification [24]. As the technological development fairly easily allows the implementaton of more and more advanced and accurate state–of–the–art software to the data, and catboost being the most recent one among the tools that will be used to predict the redshift, we expect that it can handle this task best.

To sum up this part, the machine learning methods, which will be used for prediction of the redshift are:

- Multiple linear regression,
- Random forest,
- Support vector machines,
- Artificial neural network,
- Catboost.

4.3 Metrics of model accuracy evaluation

Before we head to the prediction itself, we need to be aware of how to assess the methods' performance. In order to do that, we have to introduce a couple of accuracy evaluation metrics which will indicate the errors of regression. It is essential for the errors to be as small as possible so that the model would be considered as a good prediction tool. These metrics not only allow to understand the model better, but also may be helpful in figuring out what needs to be adjusted or improved. In contrast to the classification problem, we cannot explicitly claim that the prediction is good or bad because the target variable is continuous, so if we were to evaluate the model on accuracy parameters, we would end up overfitting the model. To avoid that, we are going to use other evaluation metrics where our model can be considered as good even if the predictions are near the actual value. The first of them is mean absolute error (MAE), which measures the average difference between the prediction and target quantity. It is the simplest and most commonly used measure for regression problems, but it can give a good assessment of a model's quality. Another metric that will be used to value the methods' performance is mean squared error (MSE), which is very similar to MAE, but measures the average magnitude of a squared error. Due to the process of raising this value to the second power, it is more sensitive to the larger errors that might appear during the analysis, as well as it is prone to omitting small ones. The last measure that we will introduce is a parameter called R Squared (R^2) also named as a coefficient of determination. It measures how much variability in the dependent attribute can be described by the model. It normally ranges between 0 and 1, so the value it provides might be more informative in comparison to the previously mentioned metrics. The bigger value it takes, the better is the fit between prediction and target value. Additionally, for some methods we will be able to tune their hyperparameters, which are responsible for the way the method works. Adjusting them might improve the results, but it generally is referred as a lengthy process, so one of the issues will also be to find the trade–off between the best model performance and the most advantageous optimization technique. As we have established the metrics, let us proceed further.

4.4 Splitting the data

In machine learning, it is essential to divide the data into the subsets of observations. Usually, there are three of them, one for training, one for validation and one for testing. The training data is used to fit the machine learning model and is generally bigger, while the test and validation split are utilized to evaluate the fit of the model. It is done this

way to avoid the high bias and simply to examine the pattern on the data not used during the training phase. In our problem we would like to use the split with the proportion of 4:1, meaning that 80% of observations will be allocated to training split, and the rest to the test (with a negligible number of validation observations). Besides that, we would like to use cross–validation method so that the results are more generalized. In particular, we would like to focus on the k–fold cross–validation type, where the data is divided into k equal sized samples and one of them is used as a test data, while the remaining ones are treated as training data. This process is repeated k times so that every subset is used exactly once as a test sample, then the average of all runs can be computed what can result in overall more authentic results. Connecting these two approaches, we can notice that the optimal value of k is 5 because it would mean that the data is divided into 5 parts out of which one is treated as a test sample, what makes up the split of 4:1. Carrying it out this way, we would obtain 5 results which can then be averaged to produce a single estimation. Obviously, it is possible to modify this parameter in order to achieve more accurate outcomes, but in this part we wanted to pay attention to the overall specification of the way we are going to proceed.

4.5 Results

At first, the selection of subsets of features and machine learning tools allowed us to determine the rules which we will follow during the implementation. Then, as we have gone through the presentation of accuracy evaluation methods and methodic of testing, we were able to establish the details of how to proceed with obtained results. Right now, we are finally ready to apply mentioned methods and create models predicting the redshift. Besides, due to the fact that the actual values of acquired redshift data might be a little inexact, we must take into account that the obtained results may deviate from the true quantity by a significant percentage, but still, we will try to create the most accurate models possible. The main goal of this regression analysis is to choose the best solution to deal with redshift prediction, in particular we want to choose the best performing combination of features subset and machine learning method among all presented models.

4.5.1 Multiple linear regression

Beginning with multiple linear regression, as mentioned before, we want to assess its performance using three metrics: mean absolute error, mean squared error and R squared statistic. After implementing it to the data, in Table 4.1 we can find the appropriate values rounded to 4 decimal places for each subset of features.

Table 4.1: Table of metrics' values for multiple linear regression

	MAE	MSE	R^2
subset 1	0.5742	0.6244	0.0103
subset 2	0.4382	0.4838	0.0563
subset 3	0.4397	0.4994	0.0270
subset 4	0.2212	0.2675	0.5701
subset 5	0.2470	0.3051	0.5185

Source: own study

What can we notice is a really poor performance of this method. Considering the fact that the mean value of redshift for all objects is at about 0.46 and the errors for first three subsets exceed this value, we can claim that it is not the optimal method of prediction for that kind of problems. Also, the inequality of MAE<MSE indicates that there are many large errors, which are magnified by the taken square of them. Apart from that, the metrics' errors values for fourth and fifth subset are much smaller than for the remaining ones and the R^2 parameter indicates that there seems to be some connection between the predicted and actual values. Being unsatisfied with obtained values, we will use the method of hyperparameter tuning so that we will be able to check whether any changes made to the default model can provide better performance. As linear regression is a simple tool, the number of adjustments is limited, but by implementing a cross–validated grid–search method we were able to perform the following adjustments:

- forcing the coefficients to be positive,
- normalizing the regressors.

Implementing models with such parameters once again allowed us to assess their performance by using the introduced metrics. The results of such evaluation may be seen in Table 4.2.

Table 4.2: Table of metrics' values for multiple linear regression with tuned hyperparameters

	MAE	MSE	R^2
subset 1	0.5723	0.5549	0.0201
subset 2	0.4522	0.4192	0.0836
subset 3	0.4663	0.4256	0.0740
subset 4	0.2004	0.2565	0.5918
subset 5	0.2368	0.2964	0.5233

Source: own study

Having a look at these values, we can notice that the improvement is insignificant and the tuning did not affect the overall impression. After all, we can claim that this method

is not particularly suitable for both the data and the problem. Despite decent result for last subset, multiple linear regression have not fulfilled the general conditions of being an effective tool, what may have been caused by its simplicity and mismatch to the data.

4.5.2 Random Forest

As for such a universal and pinpoint method, we previously mentioned to had expected better results than for linear regression, and after all we were right. By evaluating the outcome of the same metrics, we present the results in Table 4.3 with similarly rounded values.

Table 4.3: Table of metrics' values for random forest algorithm

	MAE	MSE	R^2
subset 1	0.1332	0.1495	0.7112
subset 2	0.2870	0.3807	0.2588
subset 3	0.2561	0.3457	0.3281
subset 4	0.1298	0.1556	0.7002
subset 5	0.1817	0.1852	0.6456

Source: own study

As it was foreseeable, the application of random forest method resulted in much smaller errors and better fit. Surprisingly, this time the metrics for the first sample are as good as for the ones generated by computer, while their values of R^2 mark the relatively high correlation between the predicted and target values. As for the other subsets the errors are a bit larger, but overall the method performs much better than linear regression because there is at least a partial matching of the prediction for each of the subsets. We are even able to boost the results by tuning the hyperparameters ,but due to already high efficiency, we do not expect any significant improvements. Utilizing the same process as previously, we can find the best values of method's hyperparameters. The changes made are:

- increasing the number of trees in the forest,
- using bootstrap samples when building trees,
- pruning the tree with the appropriate value of complexity parameter.

Such adjustments helped us to construct another models, for which the evaluation metrics are presented in the Table 4.4.

Table 4.4: Table of metrics' values for random forest algorithm with tuned hyperparamters

	MAE	MSE	R^2
subset 1	0.1620	0.1714	0.6859
subset 2	0.3667	0.2974	0.2672
subset 3	0.3353	0.2645	0.3486
subset 4	0.1681	0.1537	0.6939
subset 5	0.1660	0.1832	0.6766

Source: own study

It seems that the adjustment provided a slight change in the values, making the models for second, third and fifth subsets perform better, but for the remaining subsets the match to the data has negligibly decreased. Despite the lack of meaningful improvement in the evaluations, we can still state that random forest is a suitable method for this problem, providing satisfactory results.

4.5.3 Support vector machine

The next method presented was the support vector machine with radial kernel. It is one of the most common tools used in prediction in astronomy, and we wanted to check if it performs that well also in our issue. After the implementation and having a look at Table 4.5 we may claim otherwise.

Table 4.5: Table of metrics' values for SVM

	MAE	MSE	R^2
subset 1	0.8312	0.9698	0.0062
subset 2	0.6907	0.9354	0.0098
subset 3	0.7428	1.0758	0.0732
subset 4	0.3779	0.4625	0.2549
subset 5	0.2500	0.3359	0.5159

Source: own study

This algorithm completely did not cope with the regression task. The measured errors exceed the average value of redshift multiple times. The only reasonable ones are for subset 5, but the fit is only apparent. If we consider the R^2 parameter, the predictions of redshift are almost totally randomized, with no clear relationship with the target values. The only resource that could make the models viable is the hyperparameter adjustment. By implementation of the grid–search methods, we are able to inspect whether such tunings can improve the evaluations.

The proposed changes are as follows:

- using different type of kernel,
- switching the kernel coefficient,
- decreasing the regularization parameter.

These adjustments allowed us to find the most suitable values for some of the parameters, what resulted in the new models, whose metric evaluations are in Table 4.6.

Table 4.6: Table of metrics' values for SVM with tuned hyperparameters

	MAE	MSE	R^2
subset 1	0.8247	0.9332	0.0081
subset 2	0.7012	0.9326	0.0093
subset 3	0.7755	1.0239	0.0771
subset 4	0.4506	0.3685	0.2683
subset 5	0.3217	0.2443	0.5276

Source: own study

This time, similarly as before, the tuning process did not affect the evaluations significantly. The changes in metrics' values are negligible, what makes the tuned models underperform. As SVMs are not entirely suitable for imbalanced and large datasets, it could have been expected that the results would be inaccurate but not to that extent. Even by trying to implement different kernels and tune the parameters, which decreased the errors by a small margin, all of these models failed miserably.

4.5.4 Artificial neural networks

Experiencing a glimmer of disappointment about the last method, we arrive at the most commonly used one in astronomy. Among all of the tools, neural networks are the most complex ones, so for the results presented in Table 4.7, no matter how accurate they are there should be a room for improvement, but it turns out that created models are quite precise.

Table 4.7: Table of metrics' values for neural network

	MAE	MSE	R^2
subset 1	0.2544	0.2120	0.5884
subset 2	0.4419	0.4859	0.0524
subset 3	0.4362	0.4951	0.0348
subset 4	0.1874	0.1669	0.6773
subset 5	0.2203	0.1778	0.6553

Source: own study

Out of five different subsets, for three of them the prediction is slightly accurate while for the remaining actually it does not work at all. What catches an eye is the fact that unlike for other methods, MAEs for subsets 1, 4 and 5 are quite larger than MSEs. It may be for the reason that the bigger values of redshift were precisely predicted and the majority of errors are smaller than one. The values of statistic R^2 also may indicate that for these subsets there exist at least partial correlation between predictions and true values. Thanks to the complexity of neural network initialization and the variety of hyperparameters, we are able to tune some of them in order to improve the results. Changing the default values and using exhaustive cross–validated grid–search over a parameter grid allowed to slightly decrease the errors and increase the R^2. The adjustment in hyperparameters included:

- changing the activation function to 'ReLU',
- decreasing the parameter of applied $L2$ penalty,
- increasing the initial value of learning rate,
- increasing the exponential decay rate for estimates of first moment vector in a stochastic gradient-based optimizer.

Thanks to these operations, we were able to affect the learning process and obtain better results, which are presented in Table 4.8.

Table 4.8: Table of metrics' values for neural network with tuned hyperparameters

	MAE	MSE	R^2
subset 1	0.2141	0.1782	0.6463
subset 2	0.3603	0.4125	0.1671
subset 3	0.3091	0.3750	0.2126
subset 4	0.1392	0.1321	0.7264
subset 5	0.1981	0.1696	0.6706

Source: own study

As we can notice, the tuning of parameters brought about the accuracy enhancement. It seems to be truly essential to be aware of this possibility because such process can straightforwardly allow to solve the machine learning problems optimally, and can drastically improve the precision. The values of R^2 parameter now indicate a good fit of models to the data, and with further development, can result in even better prediction.

4.5.5 Catboost

For the last method, we proceeded in the same way as we did previously. We managed to measure the errors and R^2 statistic, and it appears that catboost provided quite accurate results but did not completely live up to expectations. The appropriate metrics are presented in Table 4.9.

Table 4.9: Table of metrics' values for catboost algorithm

	MAE	MSE	R^2
subset 1	0.2599	0.2216	0.5700
subset 2	0.3996	0.4687	0.0860
subset 3	0.3464	0.3848	0.2510
subset 4	0.2401	0.2057	0.6011
subset 5	0.2627	0.2161	0.5805

Source: own study

This time also the best results are achieved for computer–generated subsets, where the predictions might be considered as convincing. Similarly, the values of MSE for them are larger than MAE, what may be the result of the higher number of errors whose value don't exceed 1. Overall, the regression is not perfect, but the catboost tool is computationally very optimal and works very well for bigger datasets. As a consequence of this tool's advancement, we are also able to adjust its hyperparameters so that the outcomes are even more precise. Correspondingly to other methods, we used grid–search method to find the most optimal values of these factors and tuned them in order to improve the accuracy. In particular, the steps we took are the following:

- increasing the maximum depth of created trees,
- extending the number of iterations,
- adjusting the learning rate.

After all the changes we have made, the implementation of an enhanced models resulted in the succesive values of metrics, described in Table 4.10.

Table 4.10: Table of metrics' values for catboost algorithm with tuned hyperparameters

	MAE	MSE	R^2
subset 1	0.1456	0.1484	0.6939
subset 2	0.2881	0.3802	0.2417
subset 3	0.2641	0.3626	0.2844
subset 4	0.1434	0.1534	0.6953
subset 5	0.1977	0.2035	0.6021

Source: own study

Similarly, as before, the values of metrics occurred to be more precise, what resulted in a better fit of the models. The values of R^2 at about 0.7 allow us to state that there is a moderate correlation between the predicted and actual values. Similarly to neural networks, the introduced adjustments made the evaluations much better, increasing the models' performance by a significant margin.

4.6 Summary

As we have presented every issue that deals with the regression problem, we can now head to the summary of what we have done in this chapter. Primarily, introducing the machine learning methods allowed us to find out about their most common applications and their general purpose in astronomy, then by selecting different features we were able to diversify the results and get to know which ones are the most crucial. After that, we proposed the way to handle the prediction task and the means to assess the results. Admittedly, it seems to be an easy problem to tackle, but the truth is, that due to the little amount of knowledge we have about the measured objects, it is really difficult to estimate the real value of redshift basing only on the data captured by sky images. It has turned out that some methods are not entirely suitable for that sort of problems, dealing with various disturbances in data. The results of multiple linear regression and SVMs are far from being accurate, but the methods themselves can be viable in the future when the data obtaining processes would be even more advanced and would result in a cleaner and more transparent data. Apart from that, the remaining methods handled the regression issue satisfactorily. Also, the parameter tuning operations factually helped in improving the models and obtaining more thorough effect. When it comes to the correlation of variables, we can observe that for transformed attributes from subset 3, achieved results are slightly more accurate than for subset 2, but only by a small margin. Nevertheless, we can state that this process was rather useful. As we mentioned in the introduction to the problem, we would like to analyse the results in general and choose the best performing combination of method and subset. In order to do that, we have to pick the optimal subset across each of machine learning methods and present their results in a form of a table. To select the proper ones, we decided to treat R^2 parameter as the most important and decisive. After that, we can show the final evaluations in Table 4.11, where the number in parentheses describes the number of chosen subset.

Table 4.11: Table of metrics' values for best subset of each method

Method	MAE	MSE	R^2
Multiple linear regression (4)	0.2004	0.2565	0.5918
Random forest (1)	0.1332	0.1495	0.7112
Support vector machine (5)	0.3217	0.2443	0.5276
Neural network (4)	0.1392	0.1321	0.7264
Catboost (4)	0.1434	0.1534	0.6953

Source: own study

Just as we can see, the best model is described by the subset 4 and neural network, where the R^2 parameter takes the highest value. Despite the fact that mean absolute error for random forest algorithm is lower, the comparison of MSE values is decisive. The exceeding of 0.7 threshold for R^2 parameter, which is considered as a level where the regression model performs well [31] makes the best models viable for prediction and further development. Frankly speaking, it appears that the best subsets of variables were the ones marked as number four and five, as they stood out from the rest positively so that we can state that computer–generated ones perform better than the ones chosen by human. Another thing we can present is how those best evaluations compare to averaged values of metrics for all subsets, across all methods.

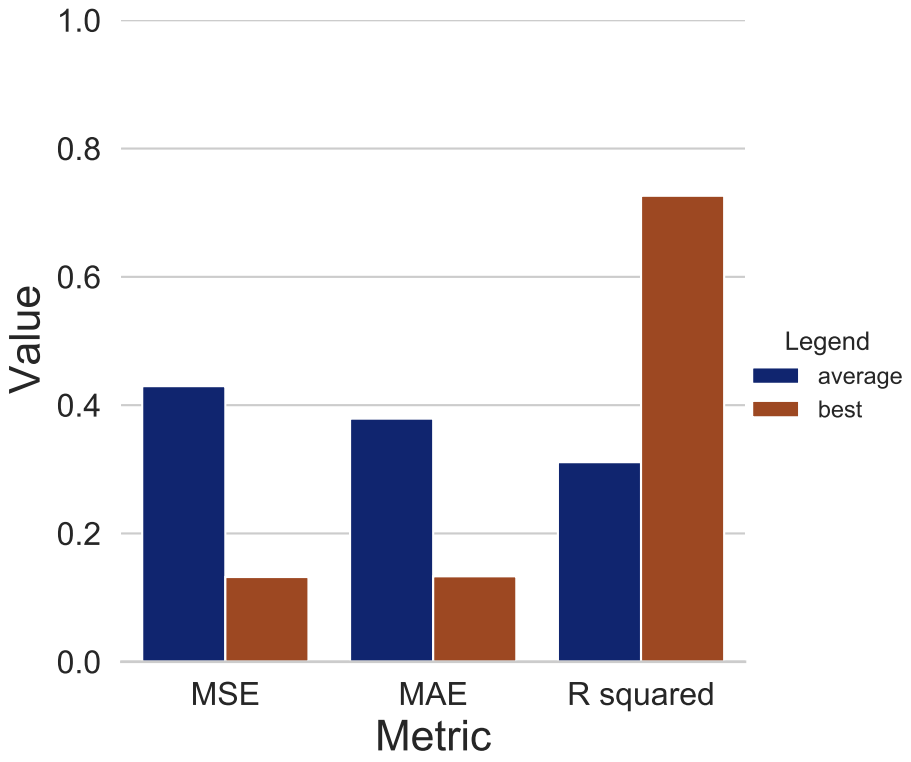

Figure 4.2: Comparison of the best achieved evaluation results to average ones

Looking at Figure 4.2 we can notice how the optimal values stand out in a positve way. They are considered as results of a better performance of selected subset–method combinations. The difference between them and the average is quite significant, what proves our statement about the efficiency of neural networks. Predicting the redshift itself turned out to be a quite difficult task, but the way we handled the problem seems to be optimal and as the observation technology processes advance even more, it could give even more accurate results. Having finished the matter of redshift, we are ready to move forward and reflect about how to distinguish different objects based on the photometric data, as it is one of the most relevant issues nowadays in astronomy.

Chapter 5

Classification of objects

In astronomy, actually many branches of science and physics are intertwined. Starting with the most classical mechanics and motion of objects, through theorems on gravity and electromagnetism, and ending with various unproven theories derived from quantum mechanics. However, all those scientific issues have one thing in common — celestial bodies and objects. They create everything what we can call the universe and without them any theory, law, or relationship wouldn't make sense. From earth, it is really difficult to grasp how huge the universe is. One bright point observed in the dark of night with the naked eye might seem to be no different from the others surrounding it. An amateur might even say that since these points in the sky are so close to each other, the actual distance between them cannot be too great. Nothing could be further from the truth. Each of these objects is one of a kind, has its own characteristics and the way it interacts with other bodies and particles. That is why in today's dynamic nature of the outer space distinguishing between these objects is really indispensable to understand the rules that govern the universe. Since the early days astronomers tried to get to know their essence, whereas the discoveries of planets in solar system were the seeds of the development of technologies that allowed people to look deeper in the darkness of the firmament. They started to catalogue brighter stars, galaxies and nebulas. Nowadays, we already know that there are many more types of bodies, including quasars, black holes and supernovas. What is even more important, all of this information comes from the images captured by telescopes that scan the sky seeking for previously unknown objects. Additionally, in recent times there was a shift towards photometric surveys, where data from only a couple frequency bands are collected, but for a big number of objects at once [29]. Due to the huge amount of data that flows in each night, there was a need for automation of the detection and classification processes, that would allow indexing them much faster. The conversion of photographies into the data was not considered troublesome, but determining the type of object from this data posed many problems. At that moment, scientists realized that different statistical and machine learning methods may facilitate their task and started implementing them on the daily basis. One of the most known classification schemes is the one proposed by Edwin Hubble for morphological classification of galaxies based on their visual features. For that purpose, he created 7 different subclasses, each for different shape of a galaxy. In those cases, the accuracy plays a significant role, as the assignment of celestial bodies to individual classes allows researchers to analyse their physical features. Having the photometric data with explained types of objects we will be able to implement some supervised classification methods so that we will be able to distinguish between the models we will create. We have previously mentioned that our dataset consists of

three separate classes — stars, galaxies and quasars, that means we will have to deal with multinomial classification. Similarly as before, we will introduce the subsequent steps that need to be taken into consideration before the whole process, and explain the way of dealing with that kind of problem.

5.1 Selection of features

This time, remembering how important for the accuracy is the selection of particular subsets, we would like to introduce the same amount of them so that the results might be diverse, what could allow us to assess which attributes are crucial for each method's performance. This time, the target variable will certainly be $class$, as it retains information about the type of objects. Similarly as before, at first, we drop any variables that refer to the IDs and date of discovery. We will also proceed in similar fashion if it comes to categorical variables — we will use one hot encoding to save all the stored information. This time though, we will take the introduced differences of apparent magnitudes as a base of the dataset. It will look very similar as for regression, but now we are going to keep the r attribute and instead of variables u, g, i and z, we introduce $u-g$, $g-r$, $r-i$ and $i-z$, following examples presented in the book [21]. Thanks to that, the values are not likely to be negative in comparison to the situation when we had to deal with parameter $z-u$. That way, the first subset will consist of these transformed variables and the categorical ones.

For the second one, we will once again let computer decide which factors would be the best while building the model. We will use the same functionality, but this time also within qualitative attributes. Using this approach, we can even derive two different subsets by setting the choice parameter to $k_1 = 3$ and $k_2 = 6$. That way in subset number 2 there will be included three best variables chosen automatically and in subset number 3 there will be six of them. The selected variables are: $redshift$, $r-i$, $g-r$ for number 2 and $redshift$, $r-i$, $g-r$, r, $u-g$, $i-z$ for number 3. We can notice that none of the categorical variables were chosen, so one might start wondering whether their contribution is significant.

For that reason, as fourth subset we can choose only the transformed magnitude variables so that we can figure out their influence on the prediction as well.

Similarly, as for regression problem we can also introduce a feature extraction method, that could create last of the subsets. Once again, the optimal way is to use the unsupervised method — PCA, as no information is lost during this transformation process. We can select $n = 5$ best components and construct the subset. To sum up, we have chosen five subsets of features which will be used as an input to machine learning methods that will be presented in the next section. The details of these subsets are as follows:

- as first subset we take transformed magnitude variables together with all the remaining attributes — continuous and categorical,
- second subset consists of 3 attributes selected automatically,
- for third subset we also have chosen automatically 6 best features,
- the fourth one includes only transformed magnitudes,
- fifth subset is a result of feature extraction conducted by PCA.

5.2 Machine learning methods

In general, we would like to implement similar methods that were used during redshift prediction with the except that not all of them can be used for classification due to their characteristics. Therefore, we will abandon linear regression, and instead we will introduce multinomial logistic regression. It differs from a classical one in that it can generalize this algorithm to multiclass problems. It is widely used in astronomy to classify different kind of stars and for detecting objects with unusual gravitational field [8]. Furthermore, it is a really viable tool because of its resistance to overfitting for low–dimensional data and that it makes no assumptions about distributions of classes in feature space. Its universality also underlines the fact that it can be easily extended to even more target classes.

Another method not previously used will be the k–nearest neighbours algorithm. As the family of nearest neighbours classifiers is among the simplest and yet most efficient classification rules and widely used in practical aspects of astronomy, the choice of it is really justified. It has been used for morphological classification of galaxies and prediction of stellar atmospheric parameters [25]. This method relies on assigning observations to an already existing group, where the algorithm finds k nearest objects in multidimensional space and selects the most numerous group. Its big merit is that the parameter k can be easily changed so that it's possible to experimentally determine it in order to achieve the best results. Despite being computationally intensive for large training sets, it guarantees to yield a low error rate, and it is fairly simple to improve the algorithm's performance.

The remaining methods used to predict the types of objects include random forest and neural network, as they appeared to be not only very efficient regressors, but also they are considered one of the best tools dealing with multiclass problems so that incorporating them in classification task will certainly be beneficial. Then we will also implement support vector machine to compare its performance and whether it is possible to improve its results since the achieved accuracy for regression did not meet the expectations. The last of the machine learning methods will also be previously used catboost. Its advancement and proficiency will enable us the possibility to pull off even better scores in comparison to the rest. Finally, we can list out all the tools which will be used in the classification of objects:

- Multinomial logistic regression,
- K–nearest neighbours,
- Random forest,
- Neural networks,
- Support vector machines,
- Catboost.

What's more, we are going to use the same splits of observations, which are explained during the prediction of redshift in section 4.4. Basically, it consists of the cross–validation method implied on the observations that are divided into 5 separate samples. Then we can simply average the results of testing the model on each sample and treat it as an outcome.

5.3 Metrics of model accuracy evaluation

In classification problem it is not possible to use the same metrics as for the regression, and for that reason, in this part we will introduce a couple of measures which will allow us to thoroughly evaluate the results. Just because the target is a discrete variable, the mean errors statistics are not valid factors, especially in case when we are dealing with multiclass problem. The metrics that would do better in that matter should be the ones that will tell us how many objects are misclassified or what fraction of each class was correctly assigned. Such approach can help us better understand the distribution of classification errors and what should be taken into consideration in order to improve the model. First and most common tool used for such problem is a confusion matrix. It allows visualization of the performance of an algorithm in the form of a table, where rows and columns represent actual and predicted classes respectively. Such object might be helpful in detecting the location of most significant errors and can be a base from which different measures can be derived. In case of our work, matrix will take the form of a table with 3 rows and 3 columns, corresponding to 3 classes of objects. Based on that we will be able to calculate the balanced accuracy parameter which describes the fraction of observations that were correctly classified to their respective classes, we will also regard to it as simply accuracy. Obviously, it can only take the value from range [0,1] and the bigger the accuracy, the better is the model. The next metric used to assess the method's performance will be precision. It presents the fraction of correctly assigned objects to a particular class over all objects allocated to that class. Precision can help us assess which class would be more problematic for prediction and is useful in determining the evaluation. Another similar to precision metric is recall. It measures the factor of the correctly assigned objects of one class over all objects that were originally of this kind. Its applications are in fact similar, as the ones for precision, where it can be used as one of metrics evaluating the model. These two measures combined properly can create another metric called F–measure. Basically being a harmonic mean of those two factors it is widely used for multiclass prediction and by a macro–averaging formula it allows to unbiasedly evaluate the prediction. All three of these parameters derived from a confusion matrices also take values from range from 0 to 1, in a way where values close to 1 are considered better. The last metric which we'll be using is area under the ROC curve (AUC). It's mainly used for the measure of a probability that a classifier will be able to tell the difference between one randomly selected positive instance and one negative instance. Besides that, AUC allows to compare different models in case of their efficiency despite its quite noisy estimation [17]. To sum up, we have 5 metrics that will allow us to evaluate the performance of different models and methods as well as compare them and check which ones are suitable for classification problem of that kind. Similarly, as in prediction of redshift, we will be able to tune hyperparameters of each model in order to improve the obtained results. Doing that, we will assess whether for all subsets the improvement was equal or for some of them it was bigger. Having presented that and all details of the analysis, we are ready to implement models into the data and perform the classification task.

5.4 Results

In previous sections, we have presented the tools and methods needed for the classification process. In this part, we will implement all of them, and we will try to interpret the obtained results. It is essential to keep in mind that accuracy for binary classification is assumed to be good for values above 0.9, but for multiclass problems it often is less strict and a good model is described by values at the level of 0.7. Despite small disturbances in the data, due to more or less precise values of redshift, we might expect a good performance of models on some of the used subsets. What is more, we can assume that the bigger the values of mentioned metrics, the better the classification model is. Similarly, as for the prediction of redshift, the main goal is to find the combination of subset and machine learning method that would be the most accurate among all presented models involved in classification.

5.4.1 Multinomial logistic regression

The first of mentioned models was constructed using multinomial logistic regression. In previous part we mentioned confusion matrix as a base of statistics for models' evaluation and on Figure 5.1 we can observe such matrices presenting the distribution of predictions for each subset of features.

Figure 5.1: Confusion matrices for multinomial logistic regression

We can notice that for first subset, the majority of observations from each of the classes were predicted to be galaxies, resulting in a bigger dispersion of the data, whereas for subset 2, the accuracy of prediction seems to be really high, as almost every object was correctly assigned to its proper class and only three stars are assumed to be of different type. The situation looks similar for subset 3, where a great part of observations was relocated to a corresponding class. The results for subsets 4 and 5 resemble the ones obtained for subset 1, as galaxy objects were accurately classified, but the remaining objects were allocated wrongly. The main issue in this method seems to be the excessive assignment of galaxy class for other objects, what may be caused by a big number of observations originally belonging to this class. Having read this table, we can move on to the presentation of metric results, which may give a broader overview of the solution.

Table 5.1: Table of metrics' values for multinomial logistic regression

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.4950	0.6553	0.4950	0.5133	0.7570
subset 2	0.9612	0.9675	0.9612	0.9639	0.9894
subset 3	0.8567	0.8838	0.8567	0.8692	0.9564
subset 4	0.5754	0.6872	0.5754	0.5918	0.8029
subset 5	0.4538	0.6872	0.6273	0.4414	0.8145

Source: own study

In the Table 5.1 our suspicions are confirmed. For subsets 1,4 and 5 the accuracy as well as the values of recall are quite poor because of the tendency to classify objects as galaxies. Together with modest score of precision, the F1–score also imply the inaccuracies in prediction. On the other hand, for remaining subsets, the model seems to be performing very well. All the metrics' values for subset 2 exceed the level of 0.9, so we are able to claim that such combination of features and model would be viable for this task. Similarly, for subset 3, the results are slightly worse, but the prediction is precise. The lack of misclassification errors may have caused such behaviour, but still, if we could average the accuracy parameter for all subsets we would find out that about 33% of objects are assigned to a wrong class. We can try to tune some of method's hyperparameters in order to improve the results and take a look at them once more. We will use the same method for searching for the optimal values, which is a cross–validated grid–search. The adjustments include:

- changing the used algorithm in the optimization problem,
- increasing the number of iterations taken for the solvers to converge,
- increasing the parameter of inverse of regularization strength.

Thanks to those operations, we were able to improve the performance for some subsets, as the evaluation metrics are presented in Table 5.2.

Table 5.2: Table of metrics' values for tuned multinomial logistic regression

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.8882	0.8923	0.8882	0.8893	0.9509
subset 2	0.9647	0.9740	0.9647	0.9690	0.9902
subset 3	0.9617	0.9733	0.9617	0.9671	0.9929
subset 4	0.5759	0.6874	0.5759	0.5923	0.8030
subset 5	0.7062	0.7927	0.7062	0.7120	0.8706

Source: own study

It appears that we managed to improve only statistics for first, third and last subset. Nevertheless, the process of hyperparameter tuning was successful and for three of feature subsets the results indicate the fit of the model into the data with a great level of accuracy. Unfortunately, the adjustment did not affect the metrics for fourth subset, so we are forced to leave it in such state and assume that method performs better for automatically chosen variables. After all, multiple linear regression dealt with this issue very well and the created models are precise, so we can move to the next method.

5.4.2 K–Nearest neigbours

Similarly as before, we will evaluate this method's performance by presenting the confusion matrices and then by showing the values of metrics that can define the goodness of fit of each model. By default, we stated that the value of k in this algorithm will be equal to 5, but it will be possible to adjust it during the hyperparameter tuning. We can now take a look at the matrices on the Figure 5.2.

The things worth noticing are the values in the upper right and lower left corners of the matrix for the first subset. About 30 000 objects that were originally of one type were classified as the other and vice versa. The same situation occurs for other pairs of different objects, but with a lower intensity. From that, we can assume that prediction for this subset was not successful. Apart from that, we notice that for second and third subsets there are only a few wrongly assigned observations, so we are able to claim that the prediction was auspicious. When it comes to two last subsets, there exists a dispersion of objects but not to that extent as in the first matrix, but for more details we will have to immerse ourselves in numerical statistics of metrics.

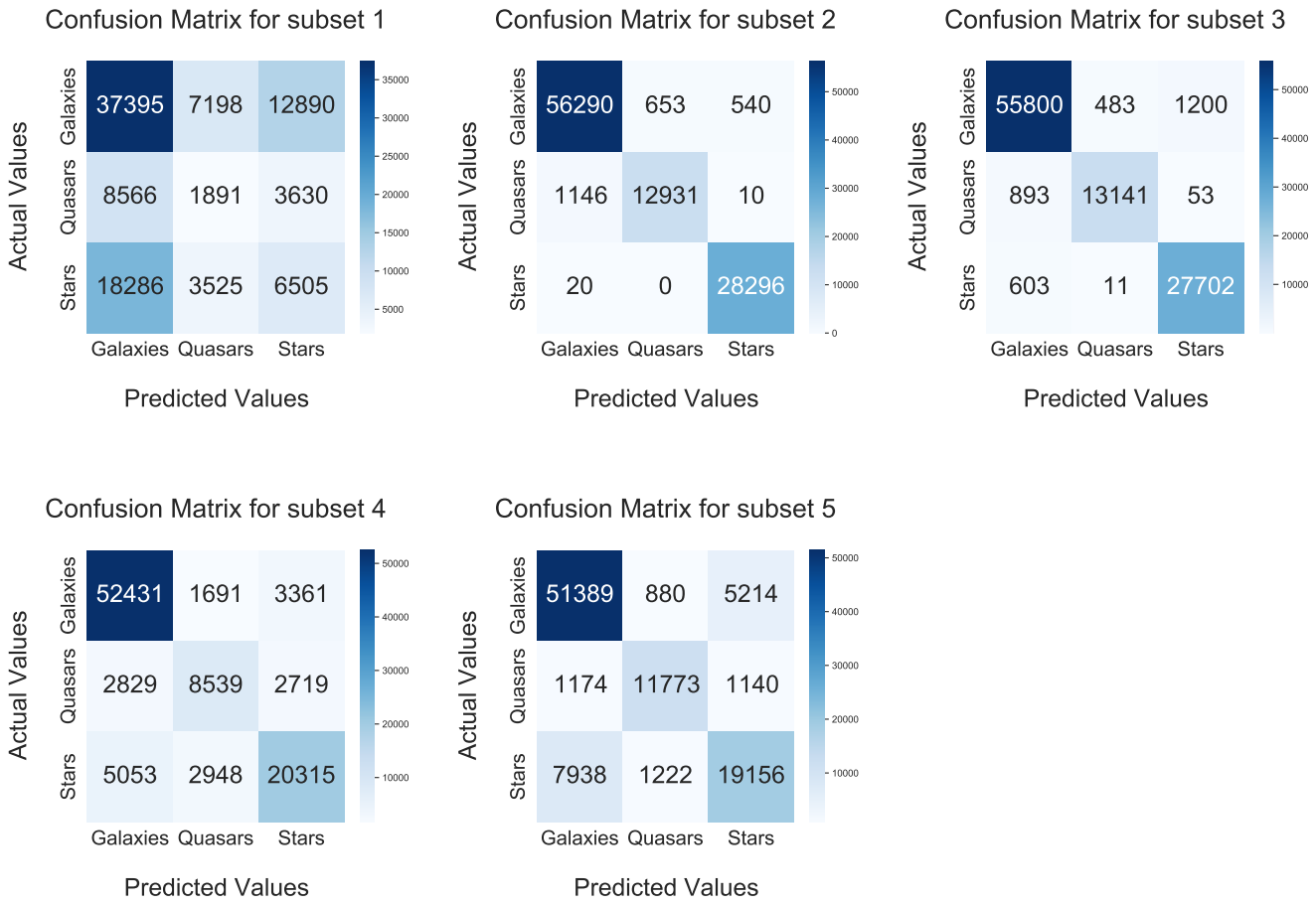

Figure 5.2: Confusion matrices for k–NN algorithm

Considering these matrices, we can also comment that the classification errors occur mainly between pairs: galaxy — quasar and galaxy — star. It may have been caused by the apparent lack of similarity between stars and quasars in case of magnitude and redshift. To analyse that further, let us take a look at evaluation metrics presented in Table 5.3.

Table 5.3: Table of metrics' values for 5–NN algorithm

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.3381	0.3532	0.3381	0.3346	0.5168
subset 2	0.9654	0.9710	0.9654	0.9680	0.9836
subset 3	0.9606	0.9648	0.9606	0.9625	0.9857
subset 4	0.7452	0.7629	0.7452	0.7529	0.8942
subset 5	0.8020	0.8189	0.8020	0.8078	0.9063

Source: own study

We see that for the first subset the classification seems to be totally random, as the accuracy is at the level of $\frac{1}{3}$, also the AUC parameter is close to 0.5 what indicates the lack of any pattern. The situation changes drastically for two next subsets, where all of the statistics take values between 0.9 and 1. It marks a great performance of the model, when the features are chosen by the computer. If we look at the classification based on the differences of magnitude, the accuracy exceeding the threshold of 0.7 might be considered as good, especially when there can be found some unusual objects that do not fit the definition of ordinariness. The evaluations for the subset being a result of imposing PCA also are quite good, as these values exceed 0.8, what also is a considerable prediction. We may try to choose different parameter k for the algorithm and check whether there would be any improvement in evaluation, especially considering first subset. Analysing a few different values of k, the best performance of mehod was achieved for $k = 20$. It may seem a little large, but when there are so many objects in the data, it is likely that such value would enhance the performance. The evaluations' statistics for the augmented model are presented in the Table 5.4.

Table 5.4: Table of metrics' values for 20–NN algorithm

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.4680	0.4709	0.4680	0.4422	0.6391
subset 2	0.9665	0.9719	0.9665	0.9690	0.9875
subset 3	0.9566	0.9628	0.9566	0.9595	0.9904
subset 4	0.8127	0.8328	0.8127	0.8214	0.9412
subset 5	0.7976	0.8247	0.7976	0.8061	0.9219

Source: own study

We may spot that there exists some improvement for first and fourth subset, where the values of metrics increased slightly. One may say, that we got rid of the randomness appearing in the first subset, but due to the really low accuracy we cannot be too certain about that. The values below 0.5 do not guarantee any fit, even though we have to deal with multiclass problem. For the remaining subsets, the improvement is negligible, partially due to some existing outliers which, no matter how precise the model would be, could not be properly classified. For two last subsets the metrics' values have reached the level of 0.8, so at last one may claim that it is a well–performed multiclass classification. Similarly to logistic regression, the evaluation methods clearly imply that the best results are being achieved when the subsets used are computer–generated.

5.4.3 Random forest classifier

When it comes to the random forest algorithm, we may remind ourselves that it was the best performing method for the problem of regression, so we would expect its results to also be meaningful. As it was mainly used for objects' classification, we can now check its viability. The main part of it will be the presentation of confusion matrices for different subsets of variables, which can be seen on Figure 5.3.

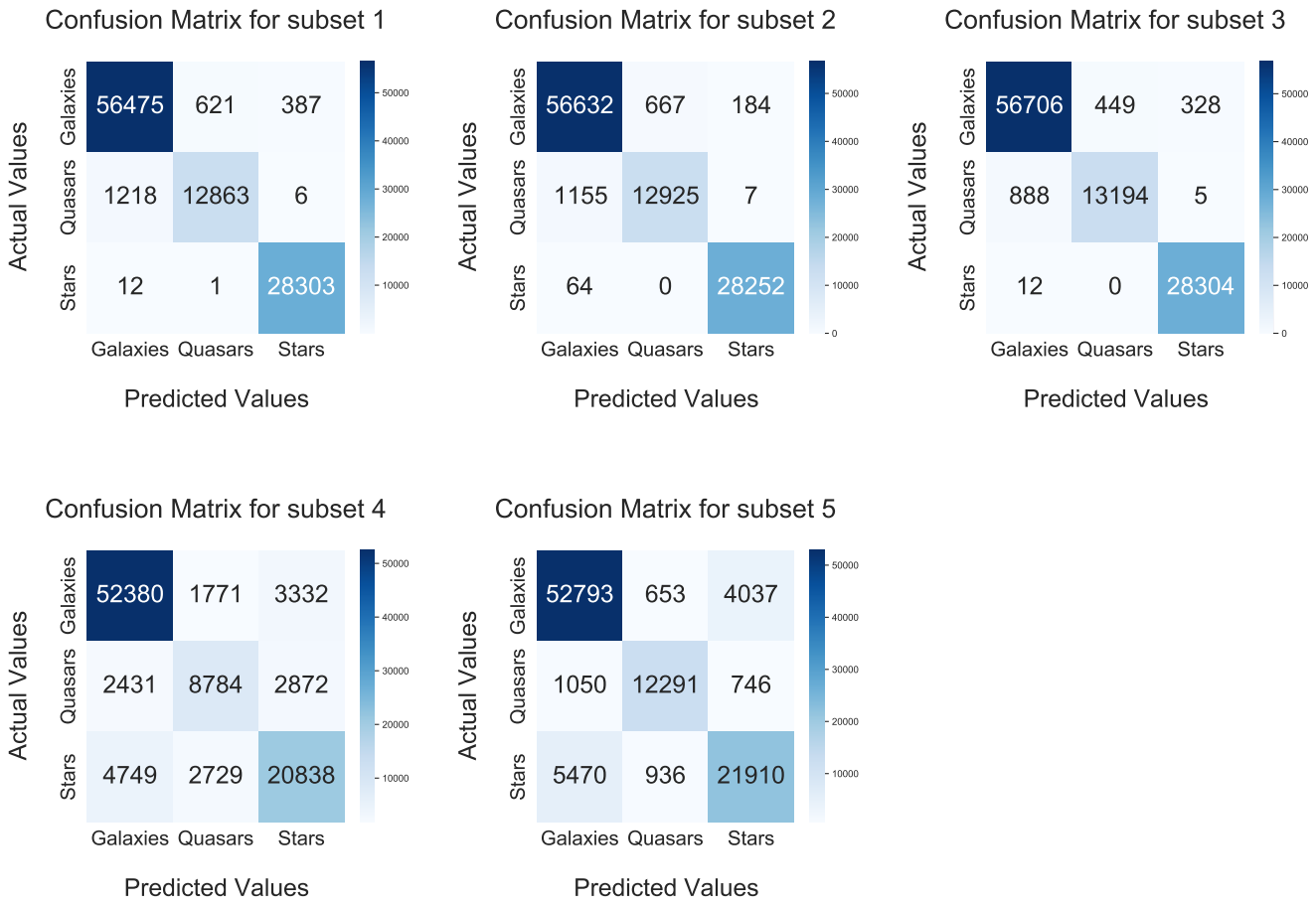

Figure 5.3: Confusion matrices for random forest algorithm

At a first glance, we can notice that for every subset, the vast majority of objects were correctly classified. The only significant errors are visible for subsets 4 and 5, but for the remaining ones there exist cells in the matrix with a single digit values. Saying that, we may even expect that the accuracy of these models will exceed 90%, what can be considered as very as a great performance. Additionally, the distinction between stars and quasars is on an exceptionally high level. The predictions seem to be accurate and by taking control of overfitting, the models may be regarded as a great classifiers. Before we jump to conclusions, let us have a look at the values of evaluation metrics presented in Table 5.5.

Table 5.5: Table of metrics' values for random forest classifier

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.9647	0.9751	0.9631	0.9687	0.9913
subset 2	0.9669	0.9746	0.9669	0.9706	0.9897
subset 3	0.9743	0.9800	0.9743	0.9770	0.9958
subset 4	0.7571	0.7713	0.7571	0.7633	0.9192
subset 5	0.8543	0.8657	0.8543	0.8590	0.9498

Source: own study

This time even for three of our subsets we achieved superior results. The values of these statistics surpassing the level of 0.95 indicate almost perfect performance of created model. The only values that stand out from the rest are the metrics for the fourth subset, but despite that, they are greater than 0.7, so we can assume a partial goodness of fit. What is also worth mentioning is the fact, that this time we achieved highly accurate results without even adjusting the hyperparameters of random forest method, especially for the first subset, where the previous models could not cope well with it. Either way, it would be unnecessary to try to tune the parameters of the method because the results are already satisfying, but we can seek improvement for two last subsets. Using the same tools as before, we analysed the adjustments, which include:

- increasing the number of decision trees,
- introducing the weighted classes, where the algorithm automatically adjusts weights inversely proportional to class frequencies in the input data,
- pruning the tree.

That way we managed to slightly increase the evaluation metrics, whose values are placed in the Table 5.6.

Table 5.6: Table of metrics' values for random forest classifier with tuned hyperparameters

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.9477	0.9289	0.9477	0.9369	0.9863
subset 2	0.9696	0.9620	0.9696	0.9656	0.9917
subset 3	0.9754	0.9633	0.9754	0.9690	0.9937
subset 4	0.7969	0.7669	0.7969	0.7744	0.9369
subset 5	0.8416	0.8240	0.8416	0.8312	0.9345

Source: own study

In fact, we succeded in improving the accuracy for fourth subset, but the change is inconsiderable. The differences in evaluations for other metrics also are negligible, but

we can claim that the whole process was useful. That way, we have proved that random forests are in particular a very utile and thorough tool for multiclass classification problems and even without any adjustments can provide a precise model.

5.4.4 Neural network

Another model that we have constructed for the purpose of object classification is the neural network classifier. By a quite complex process of learning, it has a habit of being really adaptable and efficient. By implementing it to the data and performing the classification process, we have obtained the following confusion matrices presented on a Figure 5.4.

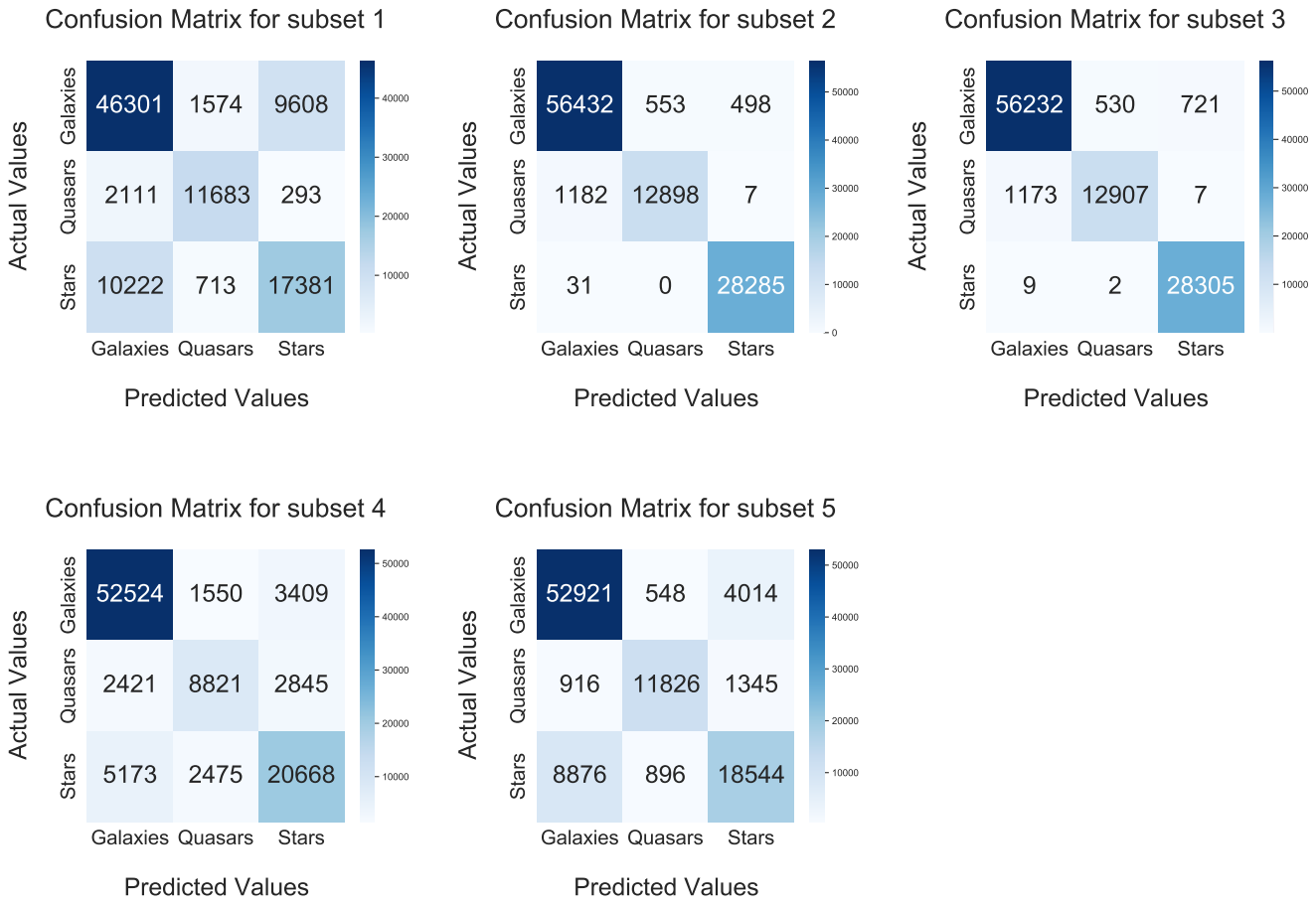

Figure 5.4: Confusion matrices for neural network algorithm

As we observe the matrices, we can see that for the first subset there are many stars that were, in fact, mistakenly assigned to galaxy class and the other way around — galaxies classified as stars, while a big part of quasars was classified correctly. Similarly, as for previously mentioned methods, subsets 2 and 3 are characterized by a significant accuracy, with only a small portion of quasars being assigned the class of a galaxy. For the last subset there are also some inaccuracies, but it may be possible to get rid of them by adjusting hyperparameters of the model, as neural networks have quite a few of them.

Another thing concerning these confusion matrices is that once again, the distinction between stars and quasars is quite clearn but it blurs among other pairs of objects. In order to get to know even more detailsn we can have a look at the metrics presenting the assessment of the neural network model which are shown in Table 5.7.

Table 5.7: Table of metrics' values for neural network classifier

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.7550	0.7621	0.7550	0.7568	0.8961
subset 2	0.9646	0.9737	0.9646	0.9688	0.9915
subset 3	0.9660	0.9707	0.9660	0.9681	0.9937
subset 4	0.7548	0.7736	0.7548	0.7628	0.9161
subset 5	0.8040	0.8381	0.8040	0.8157	0.9125

Source: own study

Quite similarly to other methods, the best performance is achieved for middle subsets, as their accuracy exceeds 0.95, what makes them great classifiers. What we have seen on confusion matrices, values of other metrics such as precision and recall are also large due to the low number of misclassified objects. For first, fourth and last subsets the values are on a level indicating a partially good prediction. This time, all the evaluations show that neural network is truly a reliable and flexible tool. We can even claim that a regular unadjusted model provided solid outcomes, but we are able to tune some hyperparameters in order to find out whether it is possible to improve achieved results, especially when it comes to boundary subsets. Because of the network's complexity and versatility we can adjust a couple of parameters, such as the number of layers, number of neurons in a layer, learning constant etc. By implementing previously mentioned searching methods, we are capable of creating an enhanced model which can be considered as an even better classifier. The adjustment of paremeters include:

- switching activation function to 'ReLU',
- introducing the non–constant learning rate, which gradually decreases at each time step,
- increasing the initial value of learning rate in order to adjust it to its behaviour.

That way, we obtained an improved model, whose numerical statistics are presented in Table 5.8.

Table 5.8: Table of metrics' values for neural network classifier with tuned hyperparemeters

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.8023	0.7992	0.7863	0.7917	0.9366
subset 2	0.9658	0.9734	0.9658	0.9693	0.9918
subset 3	0.9690	0.9723	0.9690	0.9704	0.9943
subset 4	0.7991	0.8059	0.7791	0.7860	0.9274
subset 5	0.8094	0.8379	0.8094	0.8197	0.9130

Source: own study

As we may notice, the evaluations of model have slightly increased for first and fourth subsets, reaching about 0.8. For the remaining ones, the improvement seems almost impossible to achieve, as there are too many uncertainties in the data. Nevertheless, the hyperparameter adjustment provided better results than the default model so that we may state that there is always a room for improvement when utilizing machine learning for such dataset. Overall, despite their complexity, the implementation of neural network models seems to be optimal, as they performed very well — their accuracy is at a level, where they can be regarded as one of the best tools for classification problems in astronomy.

5.4.5 Support vector machine

Another method used in prediction problem was not very efficient during regression. SVMs performed very poorly, and we may have some doubts regarding their usefulness. Hopefully, the tables will turn, and we will be able to implement the method without unnecessary complications. As we have done that using this time linear kernel as a default one, the confusion matrices are presented on Figure 5.5.

We may notice really unusual results provided by this method. For the first subset almost all objects were classified as galaxies, so despite apparently high accuracy resulting from the overall number of galaxies, it should be relatively easy to discard such model. For the second and third subsets most of the stars and quasars were classified correctly, but this time many galaxies were assigned to the wrong classes, as quasar and star respectively. The biggest dispersion in prediction is observed for the last subsets of variables, where in each cell there are assigned over 3000 objects, what makes the models likely unreliable. After analysing these tables our first impression leaves a seed of scepticism about the performance, but before we jump to conclusions let us have a look at numerical values of the evaluation metrics, which can be found in Table 5.9.

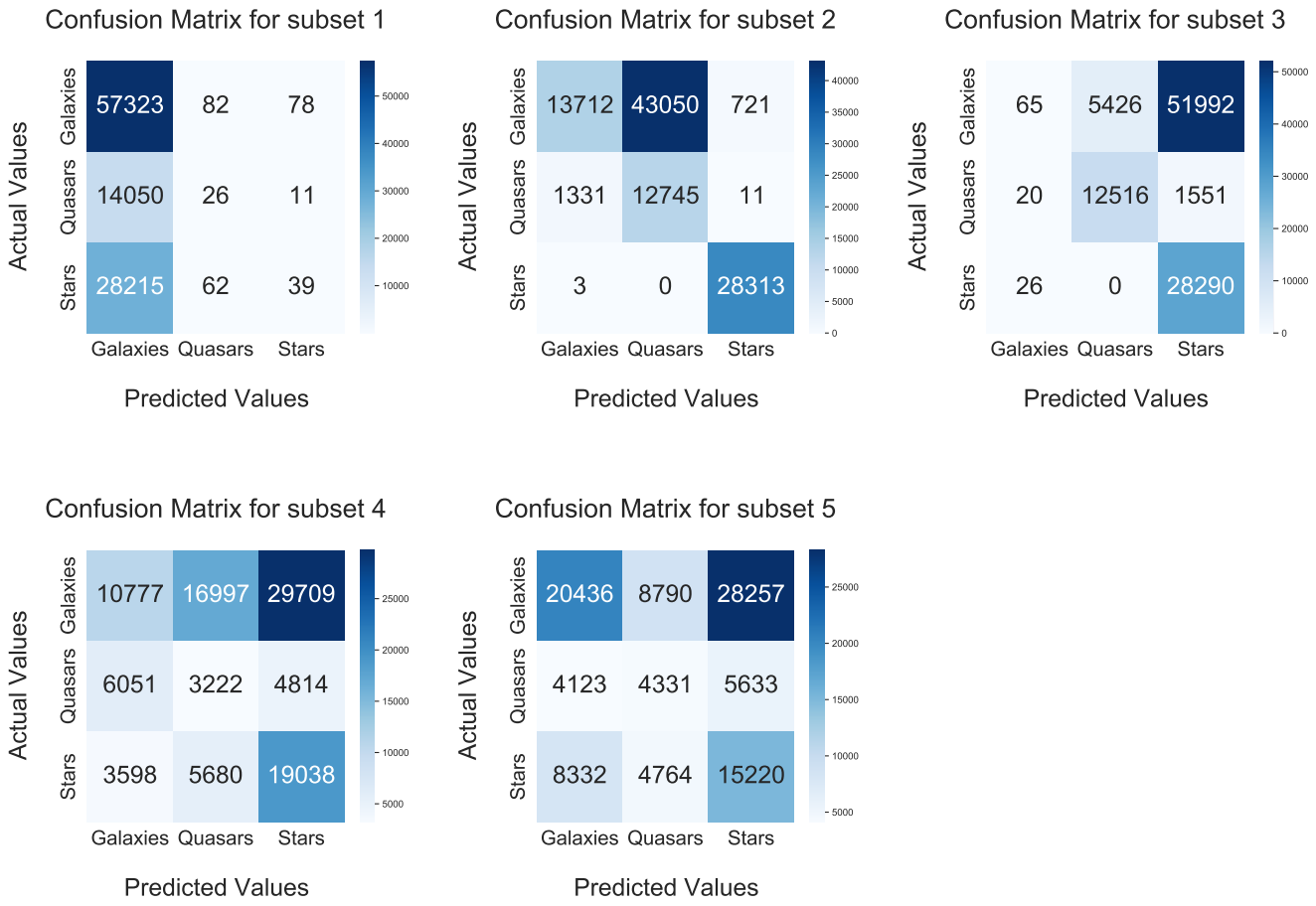

Figure 5.5: Confusion matrices for SVM algorithm

Table 5.9: Table of metrics' values for SVM

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.3334	0.3550	0.3334	0.2453	0.5525
subset 2	0.7143	0.7282	0.7143	0.5655	0.8736
subset 3	0.6295	0.5414	0.6295	0.4329	0.7781
subset 4	0.3628	0.3604	0.3628	0.2820	0.3772
subset 5	0.4001	0.4404	0.4001	0.3098	0.7136

Source: own study

Like we have expected, the metrics' values are really low, especially the accuracy parameter, where in the majority of subsets the prediction seems to be totally random. None of these models can be considered as a viable classifier, even if the accuracy for them is approximately at the level of 0.7. Values of F1–score take all the chances out of the way

of being precise. The approach of a linear support vector machine did not result in desired effects, but we will be able to adjust some of the hyperparameters and try to implement even models with another kernel. As this method is really flexible, we can try to affect the results in the most possible way because there is a big room for improvement. Once again, we use cross–validated grid–search in order to find the optimal values of parameters, the adjustments are composed of:

- increasing the number of iterations of algorithm,
- switching the kernel to radial,
- adjusting the class weights as inversely proportional to class frequencies in the input data.

Such operations allowed us to substantially improve the models, whose statistics of assessment can be seen in Table 5.10.

Table 5.10: Table of metrics' values for SVM with adjusted hyperparameters

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.4486	0.5472	0.4486	0.4134	0.6945
subset 2	0.9634	0.9722	0.9634	0.9675	0.9898
subset 3	0.7825	0.8028	0.7825	0.7334	0.8818
subset 4	0.4354	0.4307	0.4354	0.3626	0.7831
subset 5	0.3634	0.6117	0.3634	0.2165	0.6372

Source: own study

It turns out that the biggest increase in values is seen for the second and third subsets, where the accuracy reaches levels where the models can be viable predictors. Overall, besides the last subset, the tuning of hyperparameters permitted us to improve the performance and for one subset 2 we achieved the outcomes comparable to other methods. Unfortunately, we cannot entirely claim that we got rid of randomness appearing in the default model, but definitely there exists a pattern by which objects are classified. Despite the good performance for second and third subsets, SVM once again was exposed as not completely fitting tool for such type of problems, and we can objectively state that there could exist much more precise methods to deal with this issue.

5.4.6 Catboost

We have already mentioned that caboost classifier is the most recent tool destined for object prediction, and for analysis of its implementation and outcomes we will proceed very similarly as for other methods. At first, we can present the confusion matrices of predictions for each subset, which can be seen on Figure 5.6.

Figure 5.6: Confusion matrices for Catboost classifier

If we consider the first three subsets, the results seem to be superior, as the vast majority of objects for all of them is correctly assigned and the distinction between stars and quasars is on a high level. Besides that, only a couple of thousands of objects are misclassified, what causes the accuracy to be presumably over 90%. Only for two last subsets the prediction stands out negatively, but still, the great part of objects is placed on the diagonal of the matrix, which means an apparently good fit of the model. To confirm our assumptions, we can take a look at the values of evaluation metrics which can be seen in Table 5.11.

Table 5.11: Table of metrics' values for Catboost classifier

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.9713	0.9780	0.9713	0.9745	0.9960
subset 2	0.9674	0.9738	0.9674	0.9704	0.9942
subset 3	0.9739	0.9799	0.9739	0.9768	0.9972
subset 4	0.7662	0.7846	0.7662	0.7741	0.9290
subset 5	0.8576	0.8679	0.8576	0.8619	0.9546

Source: own study

Having a look at these metrics, we can clearly prove that the performance of this method was exceptional, as for three out of five subsets the accuracy and other metrics exceeded 0.96. Such results being very close to 1 indicate almost prefect classification. Even if we consider the two last subsets, looking at the metrics, we can assume a good fit to the data. Being given the possibility of improving the results posed by default model, we could try to adjust some of the method's hyperparameters and check whether there would be any improvement. To be honest, we don't expect any higher values for metrics describing first three subsets, but there are opportunities for the last ones. We proceed similarly using grid–search methods to find the best possible parameters of catboost classifier and the adjustments include:

- increasing the maximum depth of created trees,
- extending the number of iterations,
- adjusting the learning rate.

One may notice, that for regression problem we performed the same type of tuning, what resulted in fitting the model into the data even better. This time, for classification problem, such adjustment caused the formation of even more enhanced models regarding the last subsets. Numerical values of models' assessment can be found in Table 5.12.

Table 5.12: Table of metrics' values for Catboost classifier with tuned hyperparameters

	Accuracy	Precision	Recall	F1–score	AUC
subset 1	0.9716	0.9779	0.9716	0.9746	0.9981
subset 2	0.9671	0.9733	0.9671	0.9700	0.9951
subset 3	0.9735	0.9788	0.9735	0.9760	0.9980
subset 4	0.8053	0.8225	0.8053	0.8127	0.9381
subset 5	0.8554	0.8653	0.8554	0.8595	0.9534

Source: own study

The only improvement was deteced for fourth subset, where all parameteres reached 0.8. Out of all methods used in the classification problem, catboost seems to be the most

consistent, where the average value for all metrics is the highest. Its superiority may be caused by the advancement and recent development, but we have to remember that the most crucial is a single combination of method and subset, so that in summary we can take into account only one model proposed by this tool. Nevertheless, the results are more than satisfying, as the models were well–matched to the data, and such large values for model evaluations are not that common in three–class problems.

5.5 Summary

After presenting and evaluating all the models and predictions, we can now summarize what we have done in this part. By changing the parameter base from default magnitudes, to the difference of them, we were able to remove high correlation, what may have positively impacted the outcomes. Letting the computer decide which variables would be the most suitable for classification, we obtained more accurate results, and thanks to implemented various machine learning methods we could investigate which model would be the most suitable for the problem concerning the classification of astronomical objects. This task did not seem easy at the beginning as the data wasn't entirely exact and the introduction of three–class problem appeared troublesome. After all, most of the created models dealt very well with this issue and provided results, thanks to which they can be regarded as good classifiers. In particular, as we have mentioned before, we would like to focus on choosing the combination of method and subset which performed the best. For that reason, we picked the best subset of features for each method and created a table where we can compare them, and by the values of evaluation metrics we will be able to pick one. We have decided that the way we will select the best subset is the following:

1. The subset is chosen if 5 or 4 of its evaluation metrics are the best among analysed parameters.

2. If all the highest values of metrics are divided into more subsets, the higher value of accuracy parameter is decisive.

After a small analysis and chosing the proper subsets, the results can be seen in Table 5.13, where the number in parentheses indicates the number of chosen subset.

Table 5.13: Table of metrics' values for best subset for each method

Method	Accuracy	Precision	Recall	F1–score	AUC
Multinomial logistic regression (2)	0.9647	0.9740	0.9647	0.9690	0.9902
K–Nearest neighbours (2)	0.9665	0.9719	0.9665	0.9690	0.9875
Random forest (3)	0.9754	0.9633	0.9754	0.9690	0.9937
Neural network (3)	0.9690	0.9723	0.9690	0.9704	0.9943
Support vector machine (2)	0.9634	0.9722	0.9634	0.9675	0.9898
Catboost (3)	0.9735	0.9788	0.9735	0.9760	0.9980

Source: own study

It's worth mentioning that the subsets 2 and 3 appeared the same number of times in table with best models and they both were computer–generated. We can notice that the accuracy of over 95% for each of these combinations is considered as exceptional. If we looked closely and compared these values we would observe that random forest algorithm provided the highest values for accuracy and recall, whereas the best values for precision, F1–score and AUC were achieved by catboost. It is really difficult to decide which one of them performed better, but we will utilize the tiebreaker of majority so that the best combination of subset and method is the catboost for subset 2, as it assured the largest values in 3 of 5 considered evaluations. In order to visualise these differences, we can show a graph, where there are indicated the values of metrics for the most efficient subset–method combination and the average values across all the analysed models. Such comparison can be found on Figure 5.7.

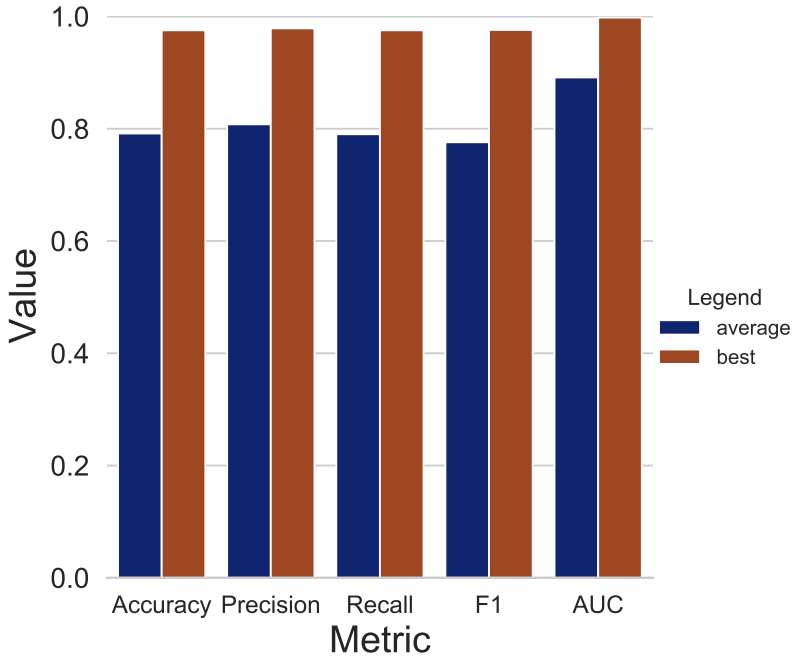

Figure 5.7: Comparison of the best achieved evaluation results to average ones

We notice that the difference is not as significant as it was for regression problem, but still, the best results are better by about 0.2. It shows the superiority of classification performed by catboost as these values are nearly perfect. In the end the classification turned out to be successful and the vast majority of objects were properly classified across every method. Partially it may be linked with redshift parameter because as we have seen on Figure 3.6, it has a big impact of how the object is perceived. That connection will be the issue of the next part of the thesis, where we will try to unify these two presented astronomical approaches and create a model predicting the objects' classes based on the predicted value of redshift from the photometric data.

Chapter 6

Unified models

In previous chapters, we have addressed the prediction problems connected with redshift regression and classification of celestial objects. Possession of the photometric data can give us even more opportunities. Usually, as the data becomes larger and larger, the primarily captured raw information about the observed object contains only the most basic attributes, such as its filtered magnitudes or placement on the night sky. Generally, other variables can be found later on, and are derivated from initial values. Such process can take a lot of time for terabytes of inputs. Apart from that, the data generated by telescopes almost always has measurement inaccuracies inside, so calculating thoroughly every single value of variable can be impossible and overall does not make sense, especially when the analysed object is far away from Earth. What is more, it might be a little problematic, when some of these already calculated variables were intertwined with each other, what would distort any further predictions. That is exactly what could have happened in our previous chapter, where the redshift values had a big impact on the predicted class of an object and the other way around. The metrics for regression, as well as the accuracies for three–class classification were unexpectedly superior for most of the results so that in this part of the thesis we would like to introduce the unified prediction models, where on the basis of the objects' locations and magnitudes extracted from the dataset we would like to foresee the class and redshift of each particular observation at the same time. Such approach seems to be more natural and has already been proposed for the analysis of astronomical images [33], where from a single photo author tried to extract the class of object and its luminosity. In our thesis, we will try to implement two separate ways to achieve our goal. In the first of them we will try to implement multi–output neural network. We will construct the network in a way that it will predict and output both attributes simultaneously. Such usage of this method seems to be computationally optimal and will allow us to compare the performance with single label prediction. Another way to solve this problem is to at first perform the regression, and based on its results we will try to classify objects — achieving the evaluations for classification. Such process is called the multi–stage prediction and works also in reverse order, primarily trying to classify objects, and based on that, introducing regression method to predict the redshift and evaluate it. We are conscious of the fact that such approaches are assumed to be not as efficient as single predictors, so that the accuracy and fit to the data will take a limited range of values. In fact, the main goal is to implement these two ways of dealing with the problem using most efficient machine learning methods and compare created models with each other and the original approach. We need to be aware that for some types of data, not the most precise solution is the best, but the one for which the trade–off between computation

time and accuracy is optimal. That is implied in particular for huge datasets with many uncertainties. Before we head to the implementation and results, let us present the details of proceeding with the problem.

6.1 Properties of the problem

To begin with, we need to determine some of the details that will enable us to perform the prediction. Similarly, as in previous chapters, the first thing we are going to establish are the subsets of features. We would mainly like to focus on the ones that have been more relevant before. For that reason, we would like to use the differences of magnitudes and the locations of objects on the sky as the base of our data and the first subset. In order to create the second one, we will once again use a feature extractor – principal component analysis. Its implementation allows reducing the muticollinearity and thus the variance. This time, though, we will take all the components into consideration while creating the subset. The two remaining ones will be constructed using the previously introduced method, namely the *Select K Best* function, which performs prediction tests and returns the most suitable variables for the problem. That way we can set the K parameter to 4 and extract the best features for classification and regression separately, which will be our third and last subsets, respectively. We take such approach so that we will be able to tell if there would be any difference in the metric evaluations while analysing the results of prediction. For such comparison, we don't need more samples so that we can move on and show the machine learning methods used for solving this problem.

The previously done predictions gave us some essential insights which tools are more useful and which ones are not suitable. In order to achieve as good results as possible, we would like to focus on the implementation of random forest, neural network and catboost. As we mentioned during the introduction, there are two ways to unify the models. In the first one we'll be using only one instance of the neural network, in which we will be able to simultaneously obtain both the class of object and its value of redshift. It is possible to build such network architecture using Keras library, which is known for its high–level neural network application programming interface. For the purpose of our work we introduced two loss functions, sparse categorical cross entropy for classification and mean squared error for regression, and in addition to that, we implemented two hidden layers of neurons that allow for the optimal prediction.

The situation is a little bit more complicated when it comes to the second manner. We need to implement two methods to primarily predict one of the labels and then based on its value joined with the proper subset predict the second one. In order to perform it, we can introduce two pairs of methods. Each first method of the pair will be used for the first part of the prediction process and similar operation applies to the second. The pairs include:

- random forest as method 1 + neural network as method 2,
- catboost as method 1 + random forest as method 2.

That way we are able to optimize the whole procedure and conserve some flexibility concerning these methods.

When it comes to the evaluation of models, we will use the same metrics as previously, but this time we will connect them together. Doing it, we will be able to assess the regression and classification at once. During the analysis, we would like to focus primarily

on the numerical values so that we will not present the confusion matrices, and we will drop the AUC parameter. Obtaining these metrics for the multi–output neural network does not pose any challenge, but for the joined methods it might be a little problematic and confusing. The detailed way we are going to assess the prediction for them on the example of one unified model is the following:

1. Predict the value of redshift using the first method and based on that implement another method to predict the class — we obtain the metrics for classification.

2. Predict the class using the first method and on its basis implement another machine learning method to predict the value of redshift — we get the metrics for regression.

To be more precise, we can take a look at the Figure 6.1 presenting the implemented procedures, where the steps for obtaining specific metrics are shown in the appropriate order.

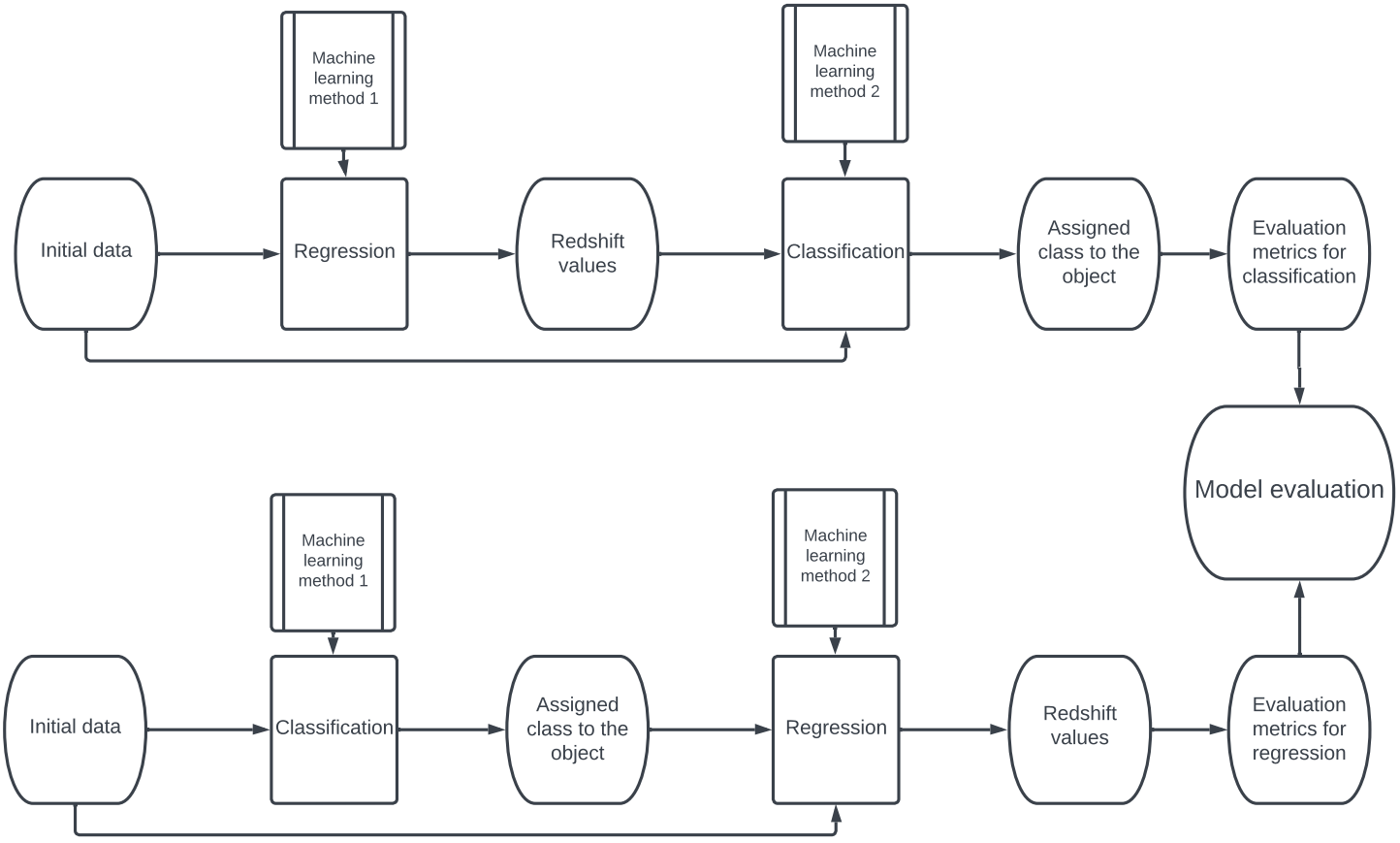

Figure 6.1: Scheme for obtaining the evaluation of a unified model

By such approach, we will be able to get all 7 measures needed to evaluate the performance. Similarly as before, we'd like to split the data and use 5–fold cross–validation method so that we can average the outputs and obtain as many predictions as there are objects in the dataset. This process was covered earlier in more detail, so we will not go any further into it, but now we can head to the implementation and present the results of performed prediction.

6.2 Results

As we have presented the way we deal with this problem, we are now prepared to implement the mentioned models and assess their performance. Before we do so, it is essential to keep in mind that although we are using the most suitable methods, their efficiency will not be as good as in a single label prediction, because of the errors made in the first stage of this process. We need to be aware, that both redshift and the class of object are strongly correlated, so that foreseeing one of them without the other might be more troublesome. Apart from that, we need to comprehend the fact that while using only neural network for multi–output prediction, we have no prior information about any of these variables, what makes the problem even more complicated. The main goal of this analysis is also to compare the generated models with each other by picking the best performing subset for each type of way of dealing with the matter. Having said, that we are ready to present the results and evaluations we achieved for respective methods and models.

6.2.1 Multi–output neural network

Even though the implementation of such method seems to be more difficult than the default one, in fact, it is based on the same assumptions and principles. Among analysed methods, only neural networks allow the simultaneous prediction of both variables, what makes them a really useful tool. The only obstacle we had to overcome, was the generated output, where we had to select the parameters of the model so that it could return categorical and numerical variables at once. Such approach turned out to be quite successful, we were even able to check what number of layers and neurons can amplify the performance so that we adjusted these parameters that way. We can see the results if we look at the evaluations of models presented in Table 6.1.

Table 6.1: Table of metrics' values for multi–output neural network

	MSE	MAE	R^2	Accuracy	Precision	Recall	F1–score
subset 1	0.3675	0.3105	0.3013	0.7923	0.8356	0.7923	0.8105
subset 2	0.3679	0.2832	0.3006	0.8092	0.8500	0.8092	0.8255
subset 3	0.3884	0.3240	0.2617	0.7261	0.7598	0.7261	0.7402
subset 4	0.3996	0.3385	0.2402	0.7426	0.7725	0.7426	0.7545

Source: own study

We can notice that for each subset, the results are quite similar, with a really low deviation. It appears that the accuracy is on a level where the classification is considered to be good, while the regression shows signs of inaccuracy. The errors assessing the performance of redshift prediction are pretty high in comparison to the best models analysed in previous parts, but still smaller than the average, and we need to remember that the whole process is computationally superior. Apparently the best evaluations were achieved for the second subset, which is composed of the data transformed by PCA. Overall, the approach of implementation of multi–output neural network was more successful when it comes to the classification. The calculated values of metrics allow us to state that the

performance is comparable with simpler approaches and can be viable for large datasets, where it is important for the algorithm to be not time–consuming.

6.2.2 Models with joined methods

When it comes to the models using connected methods, their computation time is obviously longer than for the first method, but they can use the previously predicted label to foresee the second one. One may expect that the values of metrics should be better, but when the error for first prediction is high, it can even be multiplied in the second stage of this process. Having said that, let us have a look at the evaluations for the first pair of connected methods with already adjusted hyperparameters: random forest + neural network, which are presented in Table 6.2

Table 6.2: Table of metrics' values for joined random forest and neural network

	MSE	MAE	R^2	Accuracy	Precision	Recall	F1–score
subset 1	0.3227	0.2477	0.3740	0.7984	0.8235	0.7984	0.8083
subset 2	0.3344	0.2547	0.3511	0.7978	0.8217	0.7978	0.8072
subset 3	0.3639	0.2803	0.2925	0.7043	0.7237	0.7043	0.7045
subset 4	0.3646	0.2834	0.2921	0.7173	0.7500	0.7173	0.7270

Source: own study

By the values of metrics, we see that this approach results in similar evaluations as the multi–output one. In terms of regression, the fit to the data turns out to be even better, but on the other hand the classification is not as accuracte as previously. This time, the best performing subset is the first one, whose all the measured parameters are the best among the remaining ones. Overally, the two–method approach does not differ that much, as the values are not distant and are placed in a fairly narrow range. Similarly as before, the values of R^2 parameter imply that the regression was not particularly relevant, whereas the classification can be considered viable.

Let us now move to analysis of the second pair of methods: catboost + random forest. The algorithms have already been implemented with their optimal hyperparameters so that these values are the best performing for this particular pair. The situation looks the same as before, as 7 general metrics are evaluated for the performance assessment, they can be seen in Table 6.3.

Table 6.3: Table of metrics' values for joined catboost and random forest

	MSE	MAE	R^2	Accuracy	Precision	Recall	F1–score
subset 1	0.3007	0.2331	0.4174	0.8343	0.8235	0.8343	0.8210
subset 2	0.3356	0.2547	0.3483	0.8300	0.8144	0.8300	0.8171
subset 3	0.3620	0.2835	0.2958	0.7822	0.7237	0.7822	0.7574
subset 4	0.3701	0.2890	0.2815	0.7645	0.7424	0.7645	0.7502

Source: own study

As we may see, once more, the values of metrics are placed in a similar range. This time though, for the first subset the R^2 parameter exceeded the level of 0.4, so that we can claim that the regression is at least partially suitable for the data. Besides that, the values of accuracy for all subsets indicate good fit, and as for the connection of two different methods for multiclass classification these models can be viewed as a really good predictors.

6.3 Summary

In this part, we have presented the alternative approach to the issue, where we tried to unify the problems of regression and classification. By introducing the evaluation metrics concerning both types of problems, we were able to assess the performance of each model. We have constructed three separate processes that allowed us to achieve the desired effects, which can now be seen in Table 6.4, where for each manner we selected the best performing subset. In order to be able to compare this approach with single–label prediction, we added also the most efficient method/subset combinations mentioned in previous parts of thesis.

Table 6.4: Table of metrics' values for best subset for each method

Method	MSE	MAE	R^2	Accuracy	Precision	Recall	F1
Multi–output neural network (2)	0.3679	0.2832	0.3006	0.8092	0.8500	0.8092	0.8255
Random forest + neural network (1)	0.3227	0.2477	0.3740	0.7984	0.8235	0.7984	0.8083
catboost + random forest (1)	0.3007	0.2331	0.4174	0.8343	0.8235	0.8343	0.8210

Source: own study

As we observe, the results achieved during this part of prediction diverge a bit from the ones proposed earlier. The biggest difference is spotted for the regression part, where errors are about two times larger and R^2 parameter two times smaller. We see, one more time, that the metrics responsible for models evaluation indicate very weak correlation of predicted values of redshift and the actual ones. On the other side, the disparity between measures for classification is not that distinct, additionally, regarding the values of accuracy at about 0.8 we might confirm the suitability of proposed models to the

analysed data. Among the models analysed in this part, the best results are obtained by the connected methods of catboost and random forests, as they show the highest level of suitability. We can partially conclude, that the prediction of redshift doesn't affect as much the class of object as the predicted class can impact the redshift. In particular, the errors of misclassification may have a big influence on the regression, what can be the reason why these values are so poorly predicted.

In order to visualize these differences, we can create a similar graph showing the average value of metrics among all subsets and methods, values from the best performing subsets and the optimal values of evaluations from unified models. Such a picture could help us understand how far are achieved results from the best ones, and whether they are even above average for a particular metric. Such comparison is shown on Figure 6.2.

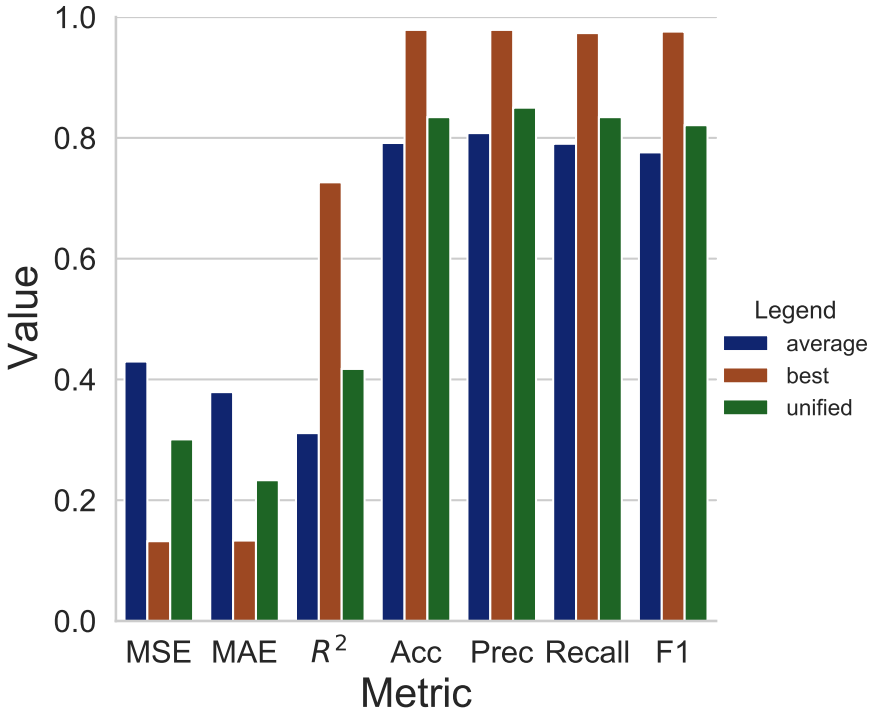

Figure 6.2: Comparison of evaluation results for unified models to average and best ones from single label prediction

We observe that each of these values is between the average and the best achieved so that they might be considered as tools of moderate performance and with some development and enhancement could be applicable to deal with the problem. To sum up, the effect of multi–label approach had both sides of the coin. On one hand, the accuracy and other metrics regarding the class allowed, in fact, claim that these models could be assumed as good predictors, but on the other hand, substantial values of errors could easily discard them. Despite that, we got rid of the depencencies between redshift and class, what allowed us to create more natural predictors and check how they would behave in described situation. Our goal was to achieve as good predictions as possible for both types of problems, but models created would be seen as viable only if for both variants the results were viable. As a result, we can state, that the models for single–label prediction are more practical due to their high accuracy, despite their time–consuming nature, but we need to remember that the results can easily be affected and distorted.

Chapter 7

Summary

During our work, we managed to present a couple of machine learning problems and applications existing in astronomy. This branch of science, however, is still rapidly developing and we as a human race know and understand only a little part of it. In order to comprehend the phenomena occurring in the universe, we have to climb to the heights of technological possibilities in the future, whereas in today's world the most information about the outer space is gathered by telescopes. These objects due to their various measurements can give us some insights about different objects that are placed in an enormous distance from the Earth. Having collected one of the datasets captured by one of them, we are able to apply numerous machine learning methods in order to predict some of the most crucial parameters of these objects. We proposed that the most influential in astronomical analysis are redshift, which is a continuous variable and the type of object, which is a categorical one. That way, we have posed two prediction problems, one for regression and one for classification.

When it comes to the prediction of redshift, we encountered a couple of issues concerning the quality of measurements, which in photometric data might occur persistently [19]. The regression itself in the majority of models appeared to be imprecise, but the other part of them were able to deal with this issue. We introduced 5 different machine learning methods and each one was used for 5 created subsets. By that means we managed to implement 25 combinations of subset–method out of which the best performing one turned out to be a neural network with subset (4) of features which was chosen by computer–generated algorithm. The assessment of each model was carried out using three metrics that allowed us to check how big the prediction errors were and how the predicted values were correlated with the actual ones. The best evaluations indicated that still, even the best model among the constructed reached only the level of 0.7 regarding the R^2 metric. We may confirm, that such results do not entirely prove the overall suitability of these models, but are a significant foundation for the further works in the topic of prediction of photometric redshift.

Regarding the matter of classification, the results were better than expected, although we had to deal with a multiclass problem. For a great part of models, the evaluations turned out to be superior and models themselves managed to be precise. Incorporating machine learning, we proceeded in a similar fashion, using this time as many as 6 methods and the same number of subsets. That allowed us to generate 30 pairs of method–subset from which the best performance was achieved by catboost algorithm implemented on automatically selected variables from subset 2. For the assessment we introduced even more evaluation metrics, beginning with visualising the predictions using confusion matrices,

through the accuracy parameter and ending at the statistics derived from the mentioned matrix. Each one of them allowed us to rate the efficiency of models, while some of them reached the accuracy level of 0.95 what makes them extremely viable. The average values of each of proposed metrics are really high, what got us to wonder whether the values of redshift aren't too influential in that type of prediction.

That suggestion lead us to introduce the idea of unified models, where we decided to predict both attributes simultaneously. One way to achieve that was to use neural network, as it's the only one among analysed methods allowing the multi–label prediction. The other manner was to predict one variable and based on these values try to predict the other using two different methods. Such approaches did not stand a chance if it comes to efficiency, but we wanted to compare them, and their performance with single–label predictions. It appeared that the evaluations for classification they did not differ that much from the values achieved earlier, but the metrics for regression implied the lack of models' fit to the data. All in all, we managed to examine the influence of both variables on each other, and apparently they are somehow intertwined.

To sum the thesis up, we were able to present a couple of different approaches to the common problems encountered in astronomy. Prediction of mentioned variables allows scientists to save a lot of time and in the days to come, if the conditions of development allow it, can be fully automated. Even though some of constructed models were not completely precise, along with the increase in the accuracy of the obtained data, there is a genuine possibility of improvement and expansion of these models into newly discovered phenomena. Some of created classification models can even nowadays be useful to foresee the class of an object due to their high accuracy. In further works on this topic it is possible to immerse even deeper into prediction processes with the application of unsupervised machine learning methods. In particular it is admissible to recognize specific types of stars or even predict their physical characteristics. Furthermore, in recent days, the freshly installed James Webb Space Telescope will allow to gather data about how different objects were formed and to search for light from the first stars and galaxies that originated in the universe after the Big Bang. Such breakthrough discoveries will certainly push humanity forward and together with the development of astronomical technology, the evolution of machine learning methods is practically inevitable, and the sooner it happens, the more opportunities we will get to learn about the universal principles governing the cosmos.

Bibliography

[1] Stellar classification dataset - sdss17. `https://www.kaggle.com/fedesoriano/stellar-classification-dataset-sdss17`. Accessed: Jan 2022.

[2] Altman, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician 46* (1992), 175–185.

[3] BALL, N. M., BRUNNER, R. J. Data mining and machine learning in astronomy. *International Journal of Modern Physics 19*, 07 (jul 2000), 1049–1106.

[4] Ballantyne, D. R., Shi, Y., Rieke, G. H., Donley, J. L., Papovich, C., Rigby, J. R. Does the AGN unified model evolve with redshift? using the x-ray background to predict the mid-infrared emission of agns. *The Astrophysical Journal 653*, 2 (dec 2006), 1070–1088.

[5] Barazza, F. D., Jogee, S., Rix, H.-W., Barden, M., Bell, E. F., Caldwell, J. A. R., McIntosh, D. H., Meisenheimer, K., Peng, C. Y., Wolf, C. Color, structure, and star formation history of dwarf galaxies over the last 3 gyr with gems and sdss. *The Astrophysical Journal 643*, 1 (may 2006), 162–172.

[6] Barkana, R., Loeb, A. High-redshift galaxies: Their predicted size and surface brightness distributions and their gravitational lensing probability. *The Astrophysical Journal 531*, 2 (March 2000), 613–623.

[7] Bartelmann, M., White, S. D. M. Cluster detection from surface-brightness fluctuations in sdss data. *Astronomy & Astrophysics 388*, 2 (may 2002), 732–740.

[8] Beitia-Antero, L., Yanez, J., de Castro, A. I. G. On the use of logistic regression for stellar classification. *Experimental Astronomy 45*, 3 (jun 2018), 379–395.

[9] Brescia, M., Cavuoti, S., Razim, O., Amaro, V., Riccio, G., Longo, G. Photometric redshifts with machine learning, lights and shadows on a complex data science use case. *Frontiers in Astronomy and Space Sciences 8* (2021).

[10] Brinchmann, J., Charlot, S., Heckman, T. M., Kauffmann, G., Tremonti, C., White, S. D. M. Stellar masses, star formation rates, metallicities and agn properties for 200,000 galaxies in the sdss data release two (dr2).

[11] Campagne, J.-E. Adversarial training applied to convolutional neural network for photometric redshift predictions.

[12] Carliles, S., Budavari, T., Heinis, S., Priebe, C., Szalay, A. S. RANDOM FORESTS FOR PHOTOMETRIC REDSHIFTS. *The Astrophysical Journal 712*, 1 (mar 2010), 511–515.

[13] Cortes, C., Vapnik, V. N. Support-vector networks. *Machine Learning 20* (2004), 273–297.

[14] de Sitter, W. On distance, magnitude, and related quantities in an expanding universe. *bain 7* (jul 1934), 205.

[15] Frank, J., King, A., Raine, D. J. *Accretion Power in Astrophysics: Third Edition*. 2002.

[16] Gray, N., Ferson, S. Logistic regression through the veil of imprecise data.

[17] Hanley, J. A., McNeil, B. J. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology 148*, 3 (1983), 839–843.

[18] Ho, T. K. Random decision forests. 278–282.

[19] Hudelot, P., Goranova, Y., Mellier, Y. The final cfhtls release, chapter 7: Photometric accuracy. *Canadian Astronomy Data Centre* (2012).

[20] Isobe, T., Feigelson, E. D., Akritas, M. G., Babu, G. J. Linear regression in astronomy. i. *The Astrophysical Journal 364* (nov 1990), 104.

[21] Ivezic, Z., Connolly, A. J., VanderPlas, J. T., Gray, A. *Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data, Updated Edition*. Princeton University Press, 2019.

[22] Jolliffe, I. T. *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, 2002.

[23] Katherine, A., Conny, A., Aguirre, S. The seventeenth data release of the sloan digital sky surveys: Complete release of manga, mastar, and apogee-2 data.

[24] Li, C., Zhang, Y., Cui, C., Fan, D., Zhao, Y., Wu, X.-B., Zhang, J.-Y., Han, J., Xu, Y., Tao, Y., Li, S., He, B. Photometric redshift estimation of bass dr3 quasars by machine learning. *Monthly Notices of the Royal Astronomical Society* (nov 2021).

[25] Li, L., Zhang, Y., Zhao, Y. K-nearest neighbors for automated classification of celestial objects. *Science in China Series G: Physics, Mechanics and Astronomy 51* (07 2008), 916–922.

[26] Liu, B., Udell, M. Impact of accuracy on model interpretations.

[27] Lu, H., Zhuang, Z., Dong, X. Photometric redshift estimation based on pca. *Journal of University of Science and Technology of China 8* (01 2006).

[28] MacLeod, C. L., Ivezic, Z., Sesar, B., de Vries, W., Kochanek, C. S., Kelly, B. C., Becker, A. C., Lupton, R. H., Hall, P. B., Richards, G. T., Anderson, S. F., Schneider, D. P. A description of quasar variability measured using repeated sdss and poss imaging. *The Astrophysical Journal 753*, 2 (jun 2012), 106.

[29] Martinazzo, A., Espadoto, M., Hirata, N. Deep learning for astronomical object classification: A case study. 87–95.

[30] Mengel, G., Sweigart, V., Demarque, P., Gross, P. Stellar evolution from the zero-age main sequence. *apjs 40* (aug 1979), 733–791.

[31] Moore, D. S. *The Basic Practice of Statistics with Cdrom*, 2nd ed. W. H. Freeman & Co., USA, 1999.

[32] Nichol, R. C. Clusters of galaxies in the sdss.

[33] Odewahn, S. C. Automated classification of astronomical images. *Publications of the Astronomical Society of the Pacific 107*, 714 (1995), 770–775.

[34] Powers, D. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *Machine Learning Technologies 2* (01 2008).

[35] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., Gulin, A. Catboost: unbiased boosting with categorical features.

[36] Puntanen, S. Linear regression analysis: Theory and computing by xin yan, xiao gang su. *International Statistical Review 78* (04 2010), 144–144.

[37] Qian, K., Shen, J. *Measurement of Apparent Magnitude and Effective Temperature with Amateur Telescopes*. arXiv, 2019.

[38] Rachen, J. P. Bayesian classification of astronomical objects - and what is behind it. In *AIP Conference Proceedings* (2013), AIP.

[39] Sammut, C., Webb, G. I., Eds. *Mean Absolute Error*. Springer US, Boston, MA, 2010.

[40] Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks 61* (jan 2015), 85–117.

[41] Seger, C. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing. 34.

[42] Sheynin, O. B. On the history of the statistical method in astronomy. *Archive for History of Exact Sciences 29*, 2 (1984), 151–199.

[43] Shy, S., Tak, H., Feigelson, E. D., Timlin, J. D., Babu, G. J. Incorporating measurement error in astronomical object classification, 2021.

[44] Skovgaard, L. Applied regression analysis. 3rd edn. n. r. draper and h. smith, wiley. *Statistics in Medicine - STAT MED 19* (11 2000), 3136–3139.

[45] Sparke, L., Gallagher, J. *Galaxies in the Universe: An Introduction*. Galaxies in the Universe: An Introduction. Cambridge University Press, 2000.

[46] Strauss, M. A., Weinberg, D. H., Lupton, R. H., Narayanan, V. K., Annis, J., Bernardi, M., Blanton, M., Burles, S., Connolly, A. J., Dalcanton, J., Doi, M., Eisenstein, D., Frieman, J. A., Fukugita, M., Gunn, J. E., Ivezić, Ž., Kent, S., Kim, R. S. J., Knapp, G. R., Kron, R. G., Munn, J. A.,

Newberg, H. J., Nichol, R. C., Okamura, S., Quinn, T. R., Richmond, M. W., Schlegel, D. J., Shimasaku, K., SubbaRao, M., Szalay, A. S., Vanden Berk, D., Vogeley, M. S., Yanny, B., Yasuda, N., York, D. G., Zehavi, I. Spectroscopic target selection in the sloan digital sky survey: The main galaxy sample. *The Astrophysical Journal 124*, 3 (Sept. 2002), 1810–1824.

[47] Tagliaferri, R., Longo, G., Milano. Neural networks in astronomy. *Neural Networks 16*, 3–4 (apr 2003), 297–319.

[48] Way, M. J., Foster, L. V., Gazis, P. R., Srivastava, A. N. New approaches to photometric redshift prediction via gaussian process regression in the sloan digital sky survey. *The Astrophysical Journal 706*, 1 (nov 2009), 623–636.

[49] Way, M. J., Klose, C. D. Can self-organizing maps accurately predict photometric redshifts? *Publications of the Astronomical Society of the Pacific 124*, 913 (mar 2012), 274–279.

[50] Wolf, C., Meisenheimer, K., Röser, H.-J. Object classification in astronomical multi-color surveys. *Astronomy & Astrophysics 365*, 3 (jan 2001), 660–680.

[51] Zhang, Y., Zhao, Y. Applications of support vector machines in astronomy. In *Astronomical Data Analysis Software and Systems XXIII* (may 2014), N. Manset and P. Forshay, Eds., vol. 485 of *Astronomical Society of the Pacific Conference Series*, p. 239.