
	<p>IVU Technical Report</p> <p>n. 03.2019</p>	
---	---	---

Comparing predictive models through visualizations: some proposals

Alessandra Legretto, Paolo Buono and Maria Francesca Costabile

Università degli Studi di Bari Aldo Moro

IVU Lab, Dipartimento di Informatica

Via Orabona, 4, Bari, Italy

{alessandra.legretto, paolo.buono, maria.costabile} @uniba.it

May 2019

Abstract

Decisions are affected by the information a person has at hand in a given moment. Predictive models can be used to analyze the available data and estimate, with a certain probability, future values. Different models produce different results. The choice of the best model is challenging and depends on the specific dataset. Several models should be compared in order to identify the one that it is more accurate for the specific analysis. The research presented in this report is part of a large project that aims at providing interactive visualizations for comparing predictive models in order to support analysts in choosing the model that best fit the data. Specifically, two visualizations are presented, which help the analyst in performing the first five tasks of the Keim's Visual Analytics Mantra. A third visualization, to support the last task *details on demand*, is still under development and will be presented in a successive report.

Contents

1	Introduction	1
2	Related work	2
3	Visualizations to compare predictive models	3
3.1	Dataset	3
3.2	Comparison Matrix	4
3.3	Pie-chart Matrix	9
4	Some examples of earlier prototypes	10

1 Introduction

Every day people have to make choices. Decisions range from trivial ones, like taking the umbrella before leaving home, to much more critical ones, like for a team of physicians to diagnose a specific disease. Technology may help in taking decisions by providing predictions.

In data science, a prediction is the *estimation of unknown values* that could occur in the future [10]. A predictive model provides estimates of future values of the variables characterizing a phenomenon.

Very important in predictive models is the *accuracy*, which is a value ranging from 0 to 1, determined by the number of correctly predicted items divided by the total number of items. However, different models applied on a same dataset may have the same accuracy but produce different outcomes. For example, let us consider *model A* and *model B* that, applied to the same dataset referring to one week period of time, predict whether to take the umbrella or not. If *model A* fails on predicting the rain only on Tuesday and *model B* fails only on Wednesday, they have the same accuracy because each fails just one day, but differ in the outcome.

Because of the increasing use of machine learning models to address several important problems, like predicting cancerous cells, researchers are more and more interested in understanding how models are trained and evaluated, in order to discover possible incorrect correlations and wrong generalizations. Understanding how models perform becomes very important to understand if and when a model is misused, in order to avoid wrong decisions. This issue is known as model interpretability, to which great attention is currently devoted (e.g. see [4]).

Understanding models helps selecting the model that works better for a given dataset. The literature reports the difficulties in selecting the model that provides correct predictions.

One reason could be that *“predictive modelers often only explore relatively few models when searching for predictive relationships [...] due to either modeler’s preference for, or knowledge of, or expertise in only a few models or the lack of available software that would enable them to explore a wide range of techniques”* [8]. The research work reported in this article provides a contribution towards creating a software system that enables the analyst to explore and compare a wider range of models. Two interactive visualizations are presented, whose aim is to support the analyst in comparing different predictive models, in order to select the best one for a given dataset.

2 Related work

Visual Analytics (VA) combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets [5]. Researchers working in VA have developed several interactive visualizations techniques. However most visualizations do not have the specific aim of comparing different predictive models applied to a dataset. In the following we briefly mention some works that use interactive visualizations to support the analysis of predictive models.

INFUSE [7] is a system in which interactive visualizations are primarily used to support the analysis of four feature selection algorithms. This helps the analyst to understand how the features are ranked by the different algorithms, and the analyst may improve a model by selecting the most significant features.

EnsembleMatrix [12] is an interactive visualization system for exploring the space of combinations of classifiers. It presents a visualization of the confusion matrices that help the user to improve the accuracy score of a model.

Prospect [9] provides a library of common visualizations, such as histograms, scatter plots, and confusion matrices. The paper describes how summary statistics and interactive visualizations can be used to address two problems: detecting classification errors and providing insights for generating new features. By using a scatter plot that compares the incorrectness of different models and the entropy, the user investigates regions in which instances are equally predicted by the models. This helps the user to detect errors in instance classifications and provide insights for generating new features.

Brooks et al. [3] propose FeatureInsight, a tool that interactively defines dictionary features for text classification. The system permits a feature-level comparison between the wrongly predicted instances and the correctly predicted instances, recommending features that could potentially be used to reduce erroneous predictions. This paper also reports that users preferred visual summaries, which led to significantly better classifier accuracy. Krause et al. [6] propose a visual analytic workflow to help data scientists and domain experts exploring, diagnosing, and understanding the outcome of a binary classifier. The workflow identifies a set of features that tend to influence the model outcome. However, support for model comparison is not provided.

Squares [11] is a system that uses histograms to show the prediction scores of a model in each class of a multi-class classification task. It helps analysts prioritize efforts in debugging

performance problems while supporting direct access to data.

The work by Zhang et al. [13] presents various interactive visualizations, including some whose aim is similar to ours, i.e., they support the comparison of predictive models. Models are compared primarily through a scatterplot-based visual summary that overviews the models' outcome. However, there are some difficulties in interpreting these scatterplots. In our work, we are trying to provide visualizations that should be more easily understandable by users. Another interesting visualization proposed in [13] is a customizable tabular view that supports the analyst to visually discriminate features extracted from the subset of instances and to identify which features are more influential in the models' outcome.

3 Visualizations to compare predictive models

This paper presents two interactive visualizations whose aim is to support the analyst in comparing predictive models, so that the model that best fits the data can be selected. The two visualizations support the first tasks of the Keim's Visual Analytics Mantra: "*analyze first, show the important, zoom, filter and analyze further, details on demand*" [5]. A third visualization, to support the last task *details on demand*, it is still under development and will be presented in a successive report; it provides details on the features of a model, since this information is relevant for the analyst. Before describing the visualizations, the datasets used in the analysis are described.

3.1 Dataset

The proposed visualizations are applied to two different dataset. One is the *Genetic Variant Classification (Clinvar)* dataset, a public resource containing annotations about human genetic variants that report if a variant has been labeled as *consistent* or *conflicting* clinical classification. First, variants are (usually manually) classified by clinical laboratories on a categorical spectrum ranging from *benign*, *likely benign*, *uncertain significance*, *likely pathogenic*, and *pathogenic*. Not all laboratories provide the same classification; variants that get *conflicting* classifications by the laboratories can cause confusion when clinicians or researchers try to interpret whether the variant has an impact on the disease of a given patient [1]. It is therefore useful to distinguish two main classes: *class 1* represents *consistent* classifications, *class 2* represents *conflicting* classifications. It is a binary classification problem where each record in the dataset is a genetic variant assigned to *class 1* or *class 2*.

The other dataset used in the examples in this paper is the *Avila* dataset. It has been extracted from 800 images of the the "Avila Bible", a giant Latin copy of the whole Bible produced during the XII century in Italy and Spain [2]. The palaeographic analysis of the manuscript revealed that the document was written by 12 different copyists, so it is a multiclass classification problem. Each instance in the dataset is represented by 10 features and corresponds to a group of 4 consecutive rows of the Bible. The goal is to classify each instance in one of the 12 classes. Because the pages written by each copyist are not equally numerous, the 12 classes have been joined into 4 groups, in order to create a balanced dataset.

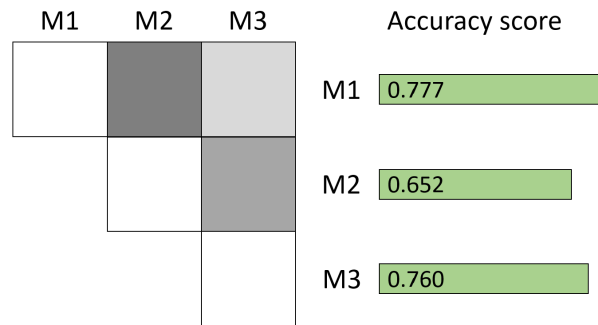


Figure 1: Comparison Matrix example

3.2 Comparison Matrix

The interactive visualization called *Comparison Matrix* addresses the comparison of the predictions of model pairs. It uses a triangular matrix where the models are reported on both, rows and columns. The initial idea was to represent in each cell the *prediction difference* between the two models corresponding to the cell row and column. The cell value is thus computed as the ratio between the number of instances of the test set that the two models predict differently and the total number of instances in the test set. The overall dataset is actually divided in two parts: the train set and test set. The former includes the instances used in the training phase of the model; the latter, i.e., the test set, includes the instances used to test the created model. The value in the matrix cell is close to 0 if the two models provide very similar predictions, otherwise is close to 1. The values in the interval $[0, 1]$ are represented in the matrix cell by gray levels going from white (corresponding to 0) to black (corresponding to 1). Figure 1 shows a Comparison Matrix with three models, namely M1, M2, M3. The cells on the diagonal of the Comparison Matrix are always white because each cell refers to the comparison of a model with itself. Thus, the prediction difference is always 0; The analyst immediately sees that the cell (1,2) is darker than cell (1,3), namely the prediction difference of M1 and M2 is greater than that of M1 and M3. On the right of the matrix, a horizontal histogram (each bar corresponds to a matrix row, i.e. to a model) shows the accuracy score of each model. For example, the M1 accuracy score is 0.777 and the M3 score is 0.760; the values are very close and we say that the two models have *similar* accuracy. M2 has a lower accuracy score (0.652), i.e. it correctly predicts a smaller number of instances of the dataset than the other two models.

Let us now to consider three classification methods: Random Forest (RF), K Nearest Neighbors (KN) and Logistic Regression (LR). By using the *Clinvar* dataset and selecting features with a specific correlation threshold, four models were created based on each one of the three classification methods. Specifically, considering Random Forest, we indicated as RF the model that used all features of the dataset; RF07 model was created using only the features with a correlation threshold less than 0.7; RF08 and RF09 used the features with correlation threshold less than 0.8 and 0.9, respectively. Similarly, four models were created considering K Nearest Neighbors (KN) and Logistic Regression (LR). These twelve models are used in the Comparison Matrix in Figure 2. The accuracy scores of each model is about 0.7, making it difficult the choice of the best model just using the accuracy score. At the bottom of Figure 2,

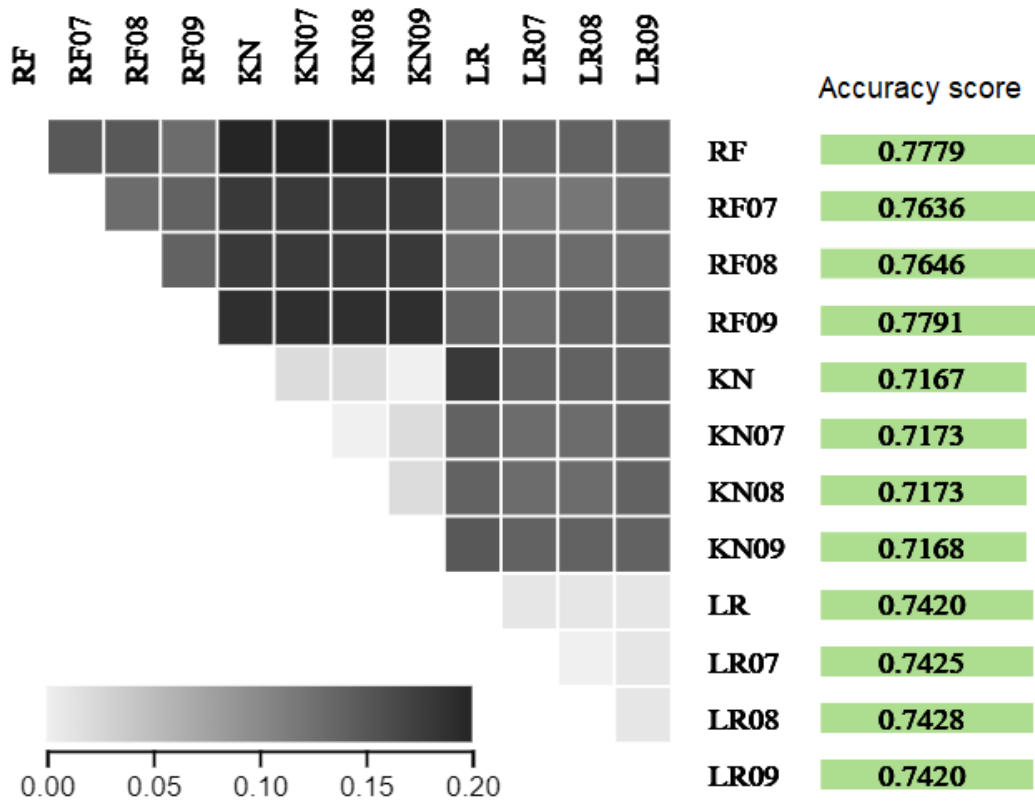


Figure 2: Comparison Matrix of twelve models of the Clinvar dataset.

a bar indicates the gray levels that represent the values of each cell. As we said before, in general the cell values may fall in the interval $[0, 1]$. Since the maximum value in the matrix of Figure 2 is 0.20, the gray levels used in the visualization would be very light, making difficult for the observer to get the differences among the values; thus, the range of the interval has been normalized to $[0, 0.20]$.

In Figure 2 some patterns that identify different models are easily recognizable; the lightest areas in the matrix represent similar models in terms of predictions of instances, i.e., the number of instances with different predictions provided by the two models is very low. The lighter regions are those that compare LR models with KN and RF models at different correlation thresholds. It is also evident that such similarity is invariant with respect to the choice of the correlation threshold. Conversely, the darkest region identifies models with the maximum difference in terms of predicted instances, such as those that compare RF and KN models at different correlation thresholds.

In this initial step of the analysis, which refers to the first two tasks of Keim's Visual Analytics Mantra, namely *analyze first* and *show the important*, the analyst goal is to quickly identify the most significant pairs of models to be later analyzed in more details. Thus, the analyst is interested in identifying the pairs of models that mostly differ in their prediction of instances, still being similar in their accuracy. Using the visualization of Figure 2, the analyst must mentally compare the accuracy score of the models, reported in the horizontal histogram.

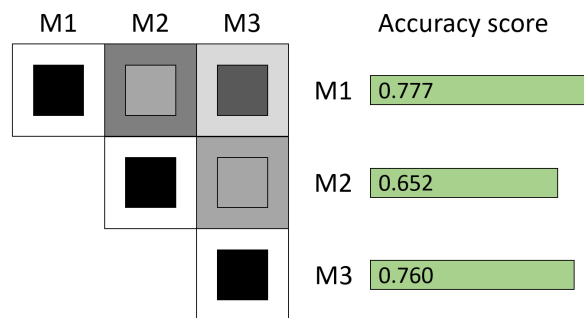


Figure 3: Comparison Matrix example with inner and outer boxes.

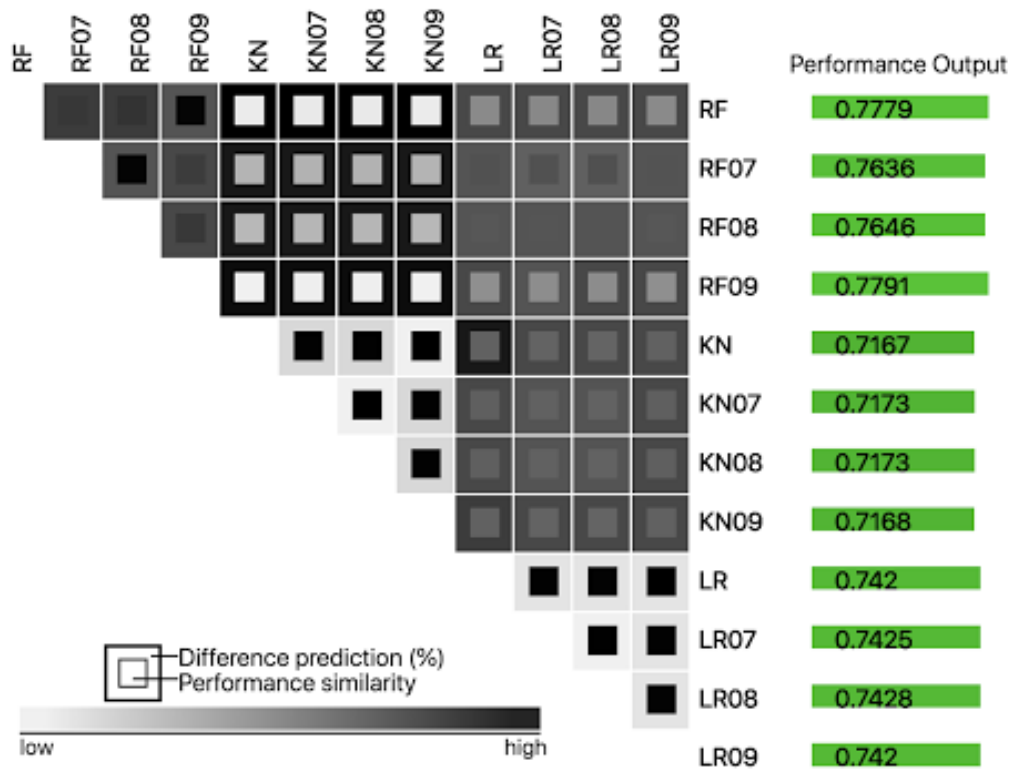


Figure 4: The new version of the Comparison Matrix in Figure 2.

Some informal test with users, performed with the thinking aloud protocol, revealed that a mental comparison overloads the analyst, it is time consuming and prone to errors. In order to reduce the cognitive workload and save the analyst time, the Comparison Matrix in Figure 2 has been modified by adding an inner box in each cell, as shown in Figure 3. Comparing

this figure with Figure 2, the value represented in each cell in Figure 2 i.e., the difference in prediction of the two models, is now represented in the outer box of each cell in Figure 3, while the inner box shows the accuracy similarity, i.e., it is a gray level that represents the difference of the accuracy scores of the two models; a greater difference is represented by a darker gray level. In order to make more evident the inner box with respect to the outer box, the inner box is visualized with a black border.

Figure 4 reports the the same models of Figure 2 with the new Comparison Matrix. Since the analyst is primarily interested in pairs of models with similar accuracy but different in their prediction of instances, the most interesting cells have both inner and outer boxes darker. An example is the cell comparing KN and LR models.

According to the Human-Centred Design, formative tests have been performed with three data analysts with medium experience on predicting models. In these tests, some users observed that they would expect the labels of the matrix rows on the left of the matrix rather than on the right. Thus, we implemented two different versions of the Comparison Matrix, which are reported in figures 5 and 6. We are going to perform other tests with users, in order

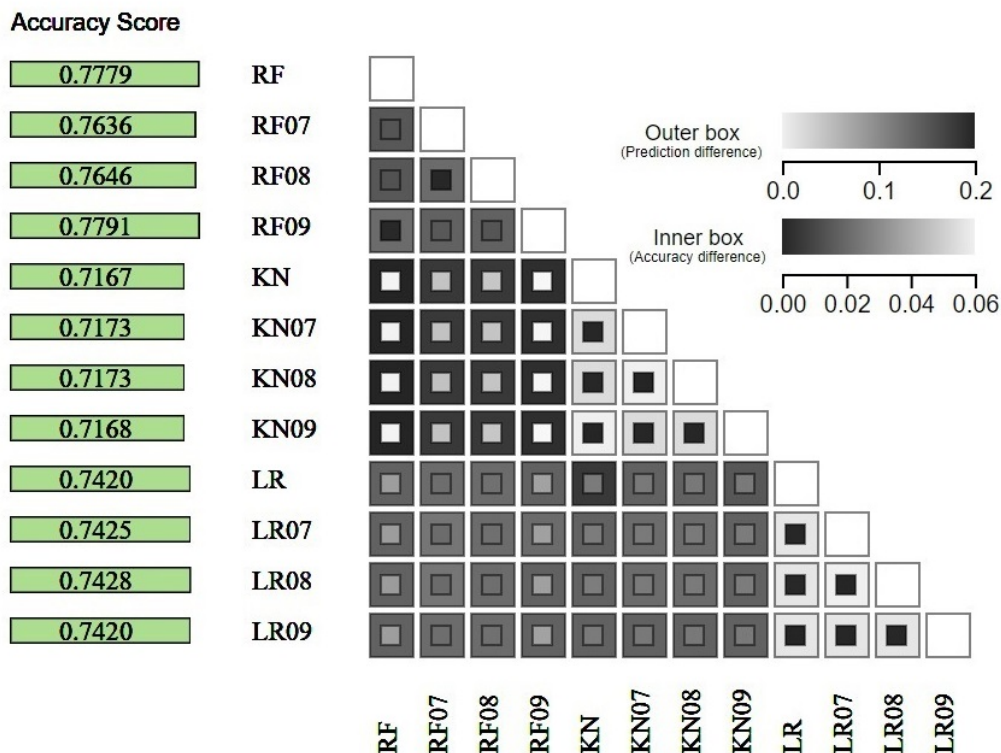


Figure 5: A new version of the Comparison Matrix in Figure 4

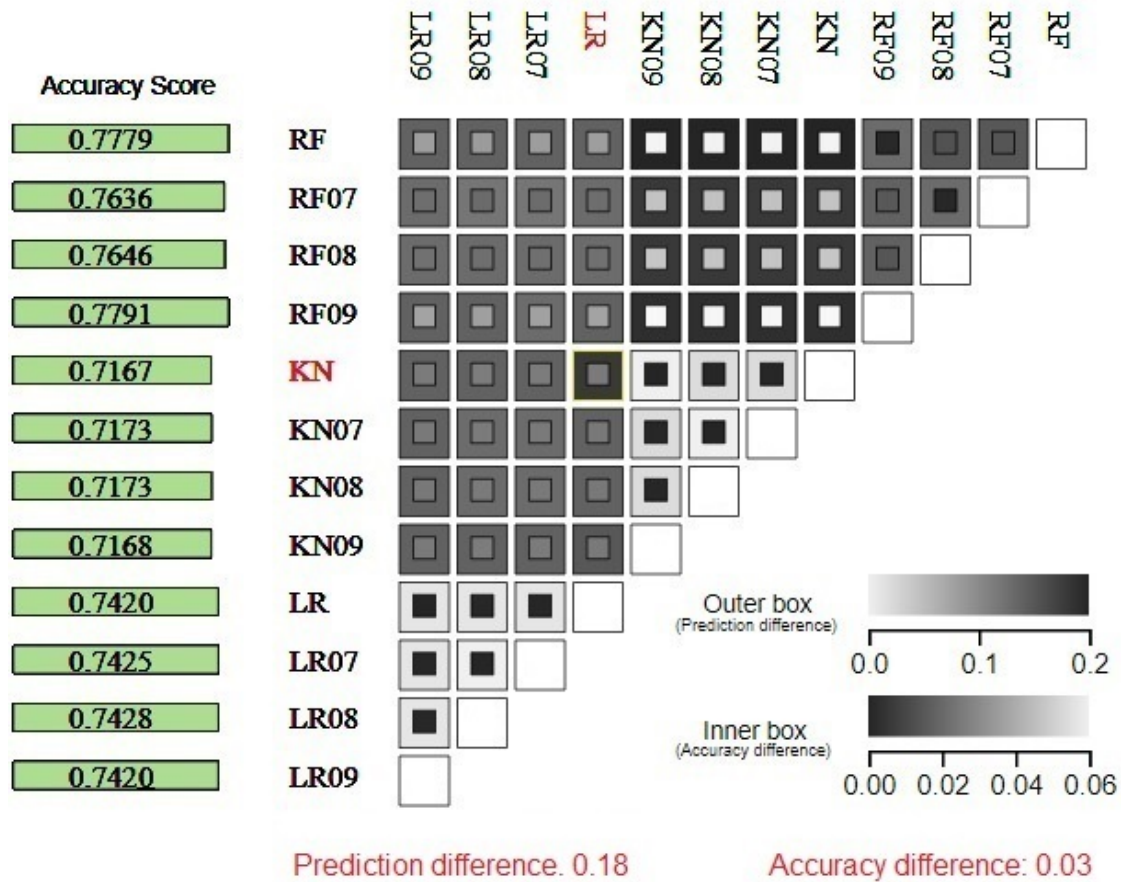


Figure 6: Another version of the Comparison Matrix in Figure 5.

to get indication of which one of the three visualizations better fit their mental model. Notice that, in these versions of the Comparison Matrix, the elements on the matrix diagonal are better visible since they are indicated by only the border of the outer box, whose internal value is white because the prediction difference is zero when comparing a model with itself. As a consequence, there is no reason to show the inner box.

In order to provide the analyst with some details about data, the interactive visualization in Figure 6 provides a mouse-over feature that, when the mouse is on the cell, highlights the cell, shows in red the name of the models of the cell row and column and highlights their accuracy scores in the horizontal histogram on the left of the figure; moreover, the values of prediction difference and accuracy difference of the two models are shown in red below the matrix.

The analyst is now interested in knowing more about such models. To this aim, a new visualization based on pie charts is shown once the analyst clicks on a cell of interest, as described in the next section. Of course, the cells on the diagonal are not clickable. It is worth noticing that this visualization, called Pie-chart Matrix actually supports the Keim's Mantra, specifically *zoom*, *filter* and *analyze further*.

3.3 Pie-chart Matrix

The analyst goal is now to further analyze what is represented in a cell of the Comparison Matrix. More specifically, the interest is to understand the prediction error of the two models when classifying the instances in the available classes. Once the analyst clicks on a cell of the Comparison Matrix because he/she wants to compare in details the two models A and B corresponding to the cell row and the cell column, a new visualization appears: it is a Pie-chart Matrix. Rows and columns represent the prediction classes available in the dataset. Therefore, also this matrix is always squared. More specifically, the cell (i,j) provides information about instances classified by model A and B with respect to the classes corresponding to row i and column j (see Figure 7).

Indeed, at this step of the analysis process, the analyst is primarily interested in finding out how many instances are not correctly classified by a model. Considering the cell (i,j) , four situations are possible: 1) both models correctly predict a number of instances in the classes corresponding to row i and column j (in short, *Both correct*); 2) model A is wrong in predicting a number of instances in the class corresponding to row i (in short, *A incorrect*); 3) model B is wrong in predicting a number of instances in the class corresponding to column j (in short, *B incorrect*); 4) both models are wrong in predicting a number of instances in the classes corresponding to row i and column j (in short, *Both incorrect*). Such four situations are coded with the following colors: gray for *Both correct*; blue for *A incorrect*; red for *B incorrect*; red-blue (striped) for *Both incorrect*.

The Pie-chart Matrix in Figure 7 presents the comparison between KN and LR models, whose accuracy values are 0.716 and 0.742, respectively. Because for the Clinvar dataset the classification problem is binary, there are two classes on both rows and columns of the Pie-chart Matrix. As an example, the pie chart in the cell (2,1) has a blue sector and a red sector. As we said, it provides in the blue sector information about the instances that are incorrectly predicted by model KN to be in class 2 and it provides in the red sector information about the instances that are incorrectly predicted by model LR to be in class 1. The radius of each pie-chart is proportional to the number of instances predicted by the two models. In Figure 7, it is evident that the biggest number of predicted instances is in the cell (1,1) and the smallest number is in the cell (2,2).

It is worth remarking that the pie charts in the cells of the matrix diagonal always show two sectors, one indicating the instances correctly classified by both models and the other indicating the the instances incorrectly classified by both models. In the case of a binary classification problem, like the one in Figure 7, the other pie charts only have two sectors. However, when the number of classes is greater than two (and thus the index of the matrix is greater than two), each pie chart may have three sectors at most: one indicating the instances incorrectly classified by the first model as belonging to the class corresponding to row i ; the second indicating the instances incorrectly classified by the second model as belonging to the class corresponding to column j ; the third indicating the instances incorrectly classified by both models.

As said at the beginning, the analyst goal is to identify the best predictive model for a given dataset. For this goal, it is certainly important to know more about the classification of

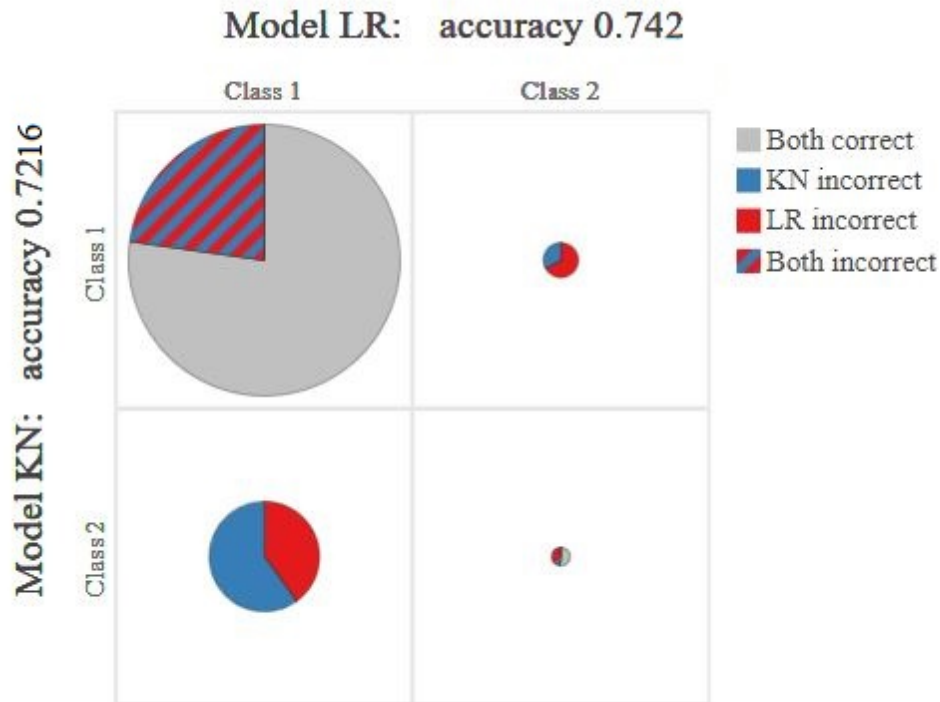


Figure 7: Pie-chart Matrix comparing KN and LR models on Clinvar dataset

instances by the two models. Indeed, with reference to the Pie-chart Matrix, the analyst might want to better investigate on the reasons of the classification errors and also to know more about the features used by the two models. This need actually refers to the last task of the Keim's Visual Analytics Mantra, namely *details on demand*.

Clicking on a cell of the pie-chart matrix, a new visualization appears showing details that may satisfy the analyst's needs. This new visualization is currently under development and will be presented in a future report, in which also examples of both the Comparison Matrix and the Pie-chart Matrices on the Avila dataset, which is a multi-class classification problem, will be illustrated.

4 Some examples of earlier prototypes

Figures 8 and 9 show some examples of earlier prototypes created to compare models. These two visualizations have been replaced by the *Comparison Matrix* visualization. Figures 10, 11 and 12 show some examples of prototypes created to compare two selected models using the predict probability value, i.e., the value that defines the probability of an instance to belong to the class predicted by the model. For various reasons these three visualizations have been replaced by the *Pie-chart Matrix* visualization reported in Figure 7.

Acknowledgement

Part of the research reported in this paper was carried out when Alessandra Legretto, who is a PhD student at the University of Bari, was working at the VIDA lab of the New York University, under the supervision of Prof. Enrico Bertini. The authors are very grateful to Enrico Bertini for his support and comments. Alessandra discussed some prototypes of the visualizations with the students of the research group coordinated by Prof. Bertini. In particular, the suggestion of Sonia Castelo about representing the difference of the accuracy values of two models in the inner square of a cell of the triangular matrix is acknowledged.

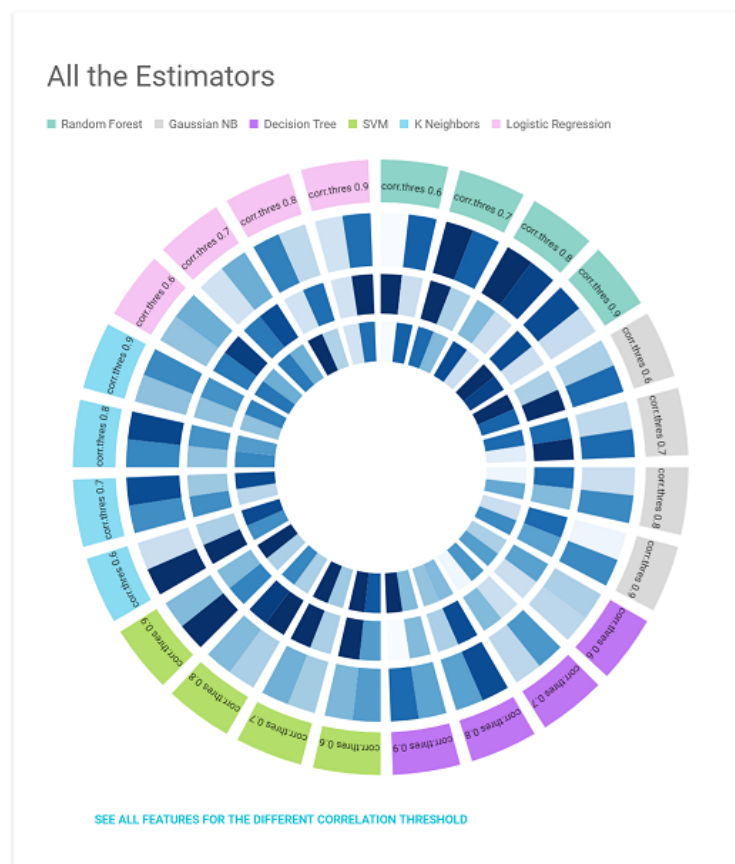


Figure 8: First circular view to compare models



Figure 9: New circular view to compare models

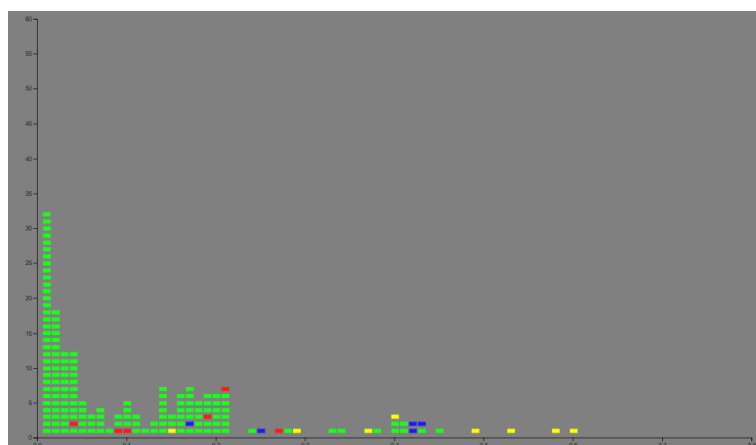


Figure 10: Pixel view of predict probability

Comparison between K Neighbors and Random Forest

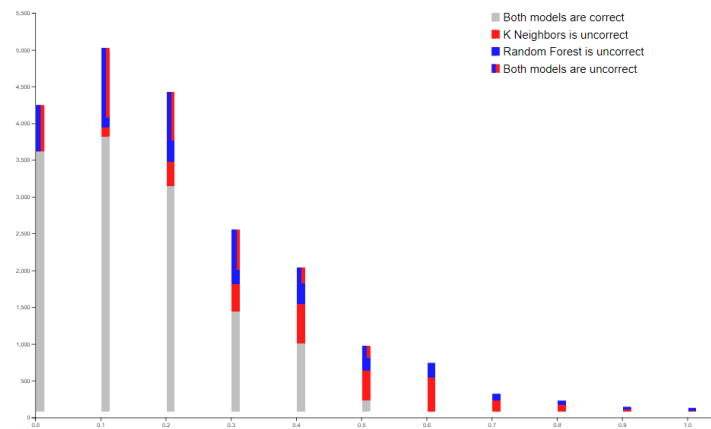


Figure 11: Histogram view of predict probability

Comparison between K Neighbors and Random Forest

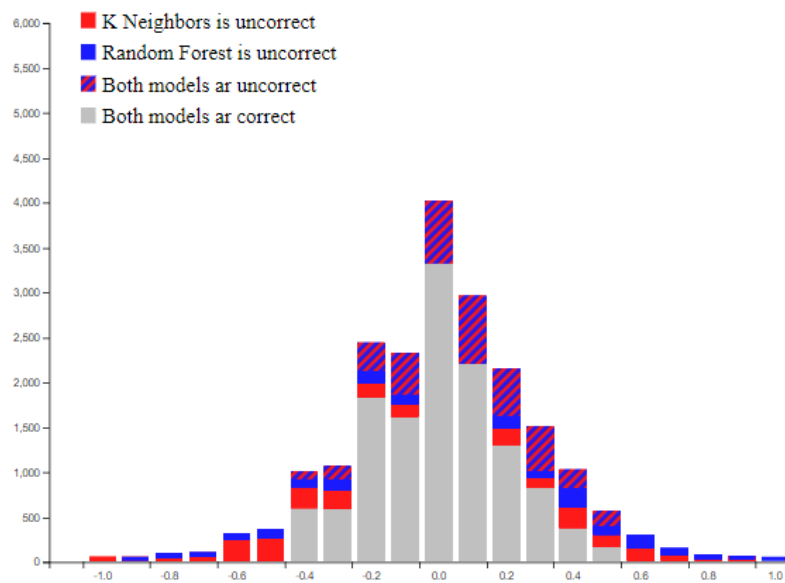


Figure 12: New histogram view of predict probability

Bibliography

- [1] Genetic Variant Classifications dataset. Retrieved from <https://www.kaggle.com/kevinarvai/clinvar-conflicting>. Last access: March 2019.
- [2] Avila dataset. Retrieved from <https://archive.ics.uci.edu/ml/datasets/Avila>. Last access: March 2019.
- [3] Michael Brooks, Saleema Amershi, Bongshin Lee, Steven M Drucker, Ashish Kapoor, and Patrice Simard. Featureinsight: Visual support for error-driven feature ideation in text classification. In *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*, pages 105–112. IEEE, 2015.
- [4] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, Oct 2018.
- [5] Daniel A Keim, Florian Mansmann, and Jim Thomas. Visual analytics: how much visualization and how much analytics? *ACM SIGKDD Explorations Newsletter*, 11(2):5–8, 2010.
- [6] Josua Krause, Aritra Dasgupta, Jordan Swartz, Yindalon Aphinyanaphongs, and Enrico Bertini. A workflow for visual diagnostics of binary classifiers using instance-level explanations. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 162–172. IEEE, 2017.
- [7] Josua Krause, Adam Perer, and Enrico Bertini. Infuse: interactive feature selection for predictive modeling of high dimensional data. *IEEE transactions on visualization and computer graphics*, 20(12):1614–1623, 2014.
- [8] Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.
- [9] Kayur Patel, Steven M Drucker, James Fogarty, Ashish Kapoor, and Desney S Tan. Using multiple models to understand data. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1723, 2011.
- [10] Foster Provost and Tom Fawcett. *Data Science for Business: What you need to know about data mining and data-analytic thinking*. " O'Reilly Media, Inc.", 2013.

- [11] Donghao Ren, Saleema Amershi, Bongshin Lee, Jina Suh, and Jason D Williams. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE transactions on visualization and computer graphics*, 23(1):61–70, 2017.
- [12] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S Tan. Ensemblematrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1283–1292. ACM, 2009.
- [13] Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S Ebert. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE transactions on visualization and computer graphics*, 2018.