PERFORMANCE COMPARISON BETWEEN STANDARD AND ENSEMBLED MACHINE LEARNING TECHNIQUES IN SUPERVISED LEARNING

INVESTIGATION AND COMMUNICATION EXAM

20.06.2020

1067666@UCN.DKL

Bartosz Piotr Lachowicz[1]

[1]University College of Northern Denmark, Aalborg 9000, Denmark;

1067666@ucn.dk

Keywords: Machine learning, Ensembled learning, Feature scaling, Overfitting

Abstract: Machine learning is part of computer science that intends to perform given task using software that was not explicitly programmed and is divided into three main paradigms: Supervised Learning, Unsupervised Learning and Reinforcement Learning.
Supervised learning addresses problems where assuming that having input data x and its output data y there is function f such that $f(x) = y$. The job of machine learning model is to find function h(the hypothesis) that best reflects true function f. Unsupervised learning deals with clustering and grouping data. It allows to discover underlying patterns between the data and makes it possible to distinguish both the similarities and dissimilarities of the input space. The last branch of machine learning – Reinforcement Learning, attempts to maximize a reward of certain environment by taking right actions e.g. traffic light control. This paper asses supervised part of machine learning and intends to compare performance difference between solutions that combine multiple machine learning algorithms into a model (ensembled learning) and the standard ones in area of classification. The benchmark will be performed using dataset from online machine learning community and competition provider Kaggle.com

At the beginning of this paper the general problem as well as aim of this paper are going to be introduced. part of this paper used dataset is going to be described. Afterwards all of the used machine learning models are going to be introduced and described. After that

Results of multiple datasets have be repeatedly computed and then examined using quantitative methods and analysis of variance. The results have proved that ensembled techniques are of significant performance gain for some of the machine learning models. At the same time showing that not all of the machine learning models work when assembled together.

Introduction

Problem description:

With number of algorithms that are present in machine learning community, one might wonder what kind of strategy to use when constructing machine learning model and if there is any benefit in combining machine learning models together. This research is going to select a number of models, explain the idea behind them and then validate them on real data.

Research questions:

This research aims to answer following questions:

1. What kind of machine learning algorithms works with classification?
2. Is there a performance gain using ensembled machine learning techniques?
3. How to ensure that the results are not random and that they can be trusted?

Summary:

The research is going to analyse learning data set using variety of machine learning methods. Performance of each of them is going to be computed using randomized runs. Afterwards the results of are going to be represented and evaluated. Furthermore after exercising the dataset the conclusion will be drawn.

Research

The aim:

For there is already many papers about the advantages of ensembled methodologies, the aim of this research is deductive. It attempts to directly show the performance gain using ensembled machine learning techniques over randomized runs of multiple datasets. [1]As many researchers have investigated the technique of combining the predictions of multiple classifiers to produce a single one. The resulting classifier (hereafter referred to as an ensemble) is generally more accurate than any of the individual classifiers making up the ensemble. Both theoretical and empirical research has demonstrated that a good ensemble is one where the individual classifiers in the ensemble are both accurate and make their errors on different parts of the input space.

Data:

As there are already many well-known and properly documented machine learning data sets the research is going to use secondary type of data. Meaning that one of the already well known dataset is going to be selected and evaluated instead of collecting, cleaning and evaluating self-collected data which would be of a primary type. During my analysis choice to use community resources for data scientist and machine learning engineers – Kaggle.com has been made. The selected dataset-titanic cruise ship crash addresses classification problem. In the dataset the goal is to predict whether a certain person survived the crash or not given certain input space about the passengers.[3]Titanic cruise dataset: Is a dataset of 2224 entries each representing a passenger of the

Titanic ship. In this dataset information such as cabin type, number of spouses, sex, etc. The machine learning goal is find out what.

Method:

For the selected dataset following actions are going to be performed: Before feeding the machine learning models the data is going to be accordingly pre-pre-processed. Then the data is going to be divided into training and validation set. That enforces the model to validate its results on previously unknown scenario and directly prevent overfitting(false results that have been created because of learning the noise of the data that is not connected with the real pattern/function).

First multiple machine learning models are going to be initialized. Then training data is going to be used to "feed" the models. Afterwards already trained models are going to be evaluated using the test set. This is going to be repeated into multiple randomized runs.

Aftewards the same datasets are going to be used to create ensembles and validate the results again. Having already pre-trained model L_1 and L_2 new model L_{12}, is going to be created as a combination of those two. Then the probability of each class is going to be extracted and added together in order to combine the results. Then the probability is going to be divided accordingly. That creates mean of the probability and that mean is going to be used in order to predict the labels of each passenger. That ensures variety between training separate models combined in ensemble. By doing that we satisfy condition of difference between models, that previously have determined 'good ensemble'. It allows the models to find errors and make mistakes in different input spaces. Therefore diversity the model is of general benefit for the prediction.

Hereafter the results are going to be saved and examined using analysis of variance. In order to determine whether there exist any performance difference between the models the analysis of variance is going to be used. Specifically the *p-value.* [9]The *P* value means the probability, for a given statistical model that, when the null hypothesis is true, the statistical summary would be equal to or more extreme than the actual observed results.

Machine learning models selection

Limitations:

Due to the fact that the computational power available for this research is limited to one machine it has to been decided to comprise within the amount of computational power necessary for testing purposed and the time it takes to test using available machine. Therefore 3 separate machine learning models are going to be selected as base for ensemble.

Selection:

In order to best evaluate the results and have control over machine learning models it has been decide to use the ones that are the most known in area of machine learning and are simple to both setup and adjust randomized runs. Therefore following models are going to be selected:

- Logistic regression
- K – nearest neighbours
- Decision Tree

Models description

Logistic regression:

In order to better understand logistic regression let us describe linear regression first. Linear regression is a method for discovering relations between the value of a continuous target variable(for example price of a particular car) and independent variables related with the same instance(that would be the features; for example size of the car or power of the engine). Linear regression attempts to discover function: $y = \beta + \beta_1 X_1 + \ldots + \beta_n X_n$, where y is a value of the target variable, β is the intercept term, β_1 … β_n are the coefficients(parameters) of the independent variables(features) X_1 … X_n

Logistic regression is a machine learning model that can be used for classification. It is built upon the same principles as linear regression but its purpose is to classify discrete values. This is why linear regression uses different cost function and inference function.

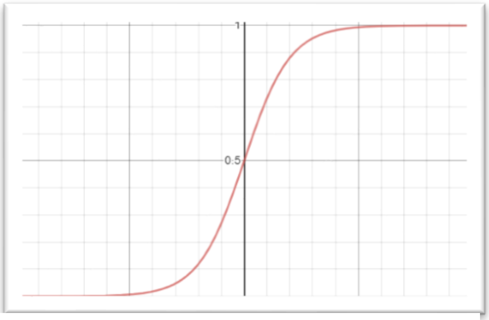


Figure 1: Sigmoid fucntion

As logistics’ regression output is supposed to be binary within the class the prediction is build using a function that outputs values in a range from 0 to 1. In logistic regression inference function ***h*** is given by formula:
$h(x) = sigmoid\ (x * theta)$, where **x** is input space, ***theta*** are coefficients of the logistic regression function and sigmoid activation function given by formula $sig(z) = \frac{1}{1+e^{-z}}$ where ***e*** is the mathematical constant.

At the end the output of function h is treated as probability of the outcome. In order to achieve binary response(0 or 1) a fixed threshold(usually 0.5) is set:

If $h < 0.5$, then logistic regression assigns sample to class 0

If $h \geq 0.5$, then logistic regression assigns sample to class 1

The cost function of logistic regression also have been modified as it is necessary to change the punishment from continuous values to discrete.

Example linear regression - a car that cost 50 000$.

- *y_true = 50 000$*
- *prediction_1 = 45 000$*
- *prediction_2 = 60 000$*

In linear regression it is correct to say that *prediction_2* has twice as big error *as prediction_1*. Linear regression model related with *prediction_1* would be twice as accurate as linear regression model related with prediction_2 .

Example logistic regression – a patient that has a cancer. Let *"1"* be denoting a patient with cancer and *"0"* a healthy patient then:

- *y_true = 1*
- *prediction_1 = 0.7*
- *prediction_2 = 0 .40*

In case of logistic regression this case *prediction_1* correctly identified the patient to have a 70% (class 1) chance to of having a cancer and 30%(class 0) against it. On the other hand the *prediction_2* returned 40%(class 1) certainty for the patient to have cancer and 60%(class 0) that he has not. In this case simple comparison of 0.30 error vs 0.6 error would tell us that logistic regression model related with *prediction_1* is only two times better than logistic regression model related with *predicrtion_2*. That is wrong to assume therefore adequate function that punishes the model non-linearly has to come in replacement. In figure 3 and 4 we can now see that errors are no longer linearly represented, but the model punishes incorrect classification using logarithmic function, which allows the logistic regression to achieve its goal and increase the value of an error significantly when classes are incorrectly classified with high confidence.

Logistic regression cost function :

$-\log(h(x))$ if y = 1

$-\log(1 - \ h(x))$ if y = 0

Where y is the actual class of the sample, h(x) is the prediction.

In order to get rid of those two cases, the cost function for the logistic regression gets compressed into:

$$-\ (ylog(x)\ +\ (1-y)log(1-h(x))$$

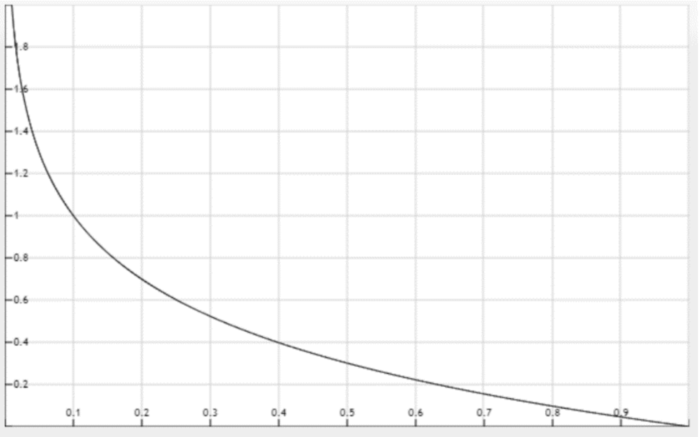


Figure 2: Cost function for y=1

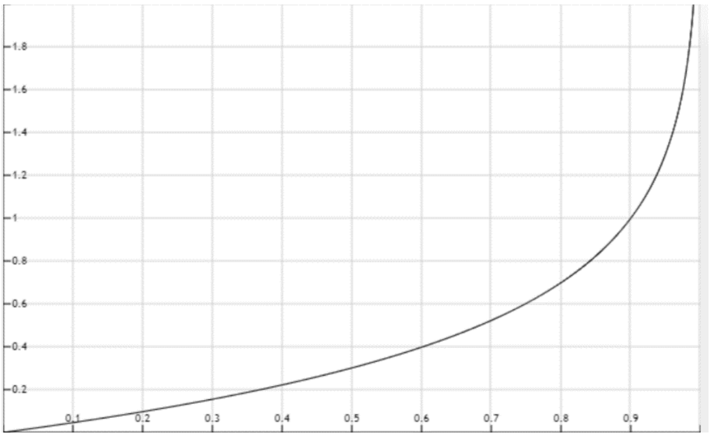


Figure 3: Cost function for y=0

K – nearest neighbours:

K – nearest neighbours is a machine learning model that can be used both for classification and regression. This paper will focus only on the classification functionality of the KNN. [4]For a data record **t** to be classified, its **K** nearest neighbours are retrieved, and this forms a neighbourhood of **t**. Majority voting among the data records in the neighbourhood is usually used to decide the classification for **t** with or without consideration of distance-based weighting. The class is determined based on the most frequent class within some range. The value **K** is user defined meaning that the researcher/programmer have to find the most suitable K for the assessed data set.

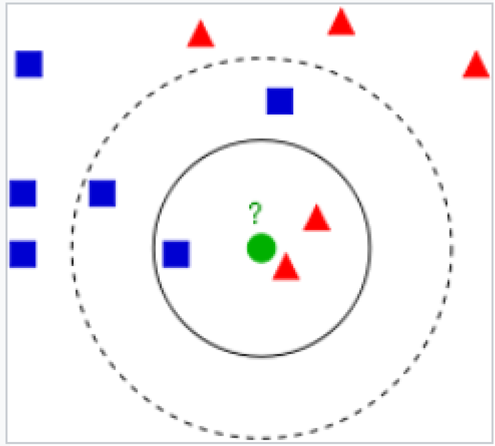


Figure 4: [5]KNN sample classification

The **K** indicates how many neighbours should KNN consider when looking for the answer about samples class. For instance in figure 3 we can see the outcome of classification K = 3 and K = 5. For K=3 the test sample would have been categorized as red triangle. For K=5 the test sample would have been categorized as blue square. The distance between each of the points is usually calculated using the Euclidean distance: $d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$

Let`s look at the following example describing the price of the vehicle:

ID	stroke	city_mpg
car_1	2.72	24
car_2	3.45	17
car_3	2.68	27
car_4	3.16	30

$$d(car\_1, car\_2) = \sqrt{(2.72 - 3.45)^2 + (24 - 17)^2}$$

$$d(car\_1, car\_2) = \sqrt{0{,}5329 + 49} \approx 7.03796$$

Looking closer on the result of the Euclidean distance calculation, one might observe that the distance of both features is very close to the distance of *city_mpg* itself(when considering only *city_mpg* the result of the Euclidean distance would be **7**). This is due to the fact that in example above the feature *city_mpg* has approx. 90 times bigger impact on the result of the distance between those two samples. This means that if the features of the dataset under investigation are of different scale the model built using KNN is going to produce incorrect results. The bigger the scale of a feature is the more weight it carries in the prediction.

In order to solve this problem before feeding the data into a KNN model, it should have been pre-processed. The aim of the pre-processing is to modify values of the features so that all of them are in the same scale. That process is called feature scaling. One method of rescaling is to use min-max scaling:

$$x' = \frac{x - \min(x)}{max(x) - \min(x)}$$

When considered the same example after feature scaling the car dataset ends up with following values:

ID	stroke	city_mpg
car_1	0.05194	0.5384
car_2	1	0
car_3	0	0.7692
car_4	0.6233	1

Use of min-max scaling modifies the data in such way that the values are in range of (0,1). That means that if the KNN does not have predefined weights all of the features` distances are equally relevant.

Moreover one might use exploratory analysis and visual representation to present the difference between scaled and not scaled data. The visual distances between samples represents the actual difference between those samples within certain feature

In order to visualise this problem figure 5 and 6 was created:

- Figure 5 represents relative distances between *city_mpg* and *stroke* before scaling. The distances between datapoints on *stroke* axis are negligible compared to the distances between city_mpg.
- Figure 6 represents relative distances between *city_mpg* and *stroke* after scaling. Here the distances between datapoints on *stroke* axis are of much more importance compared to the previous state of the data.

Those two figures visually represent how KNN would learn from the datapoints. When the values of *city_mpg* are much greater than values of *stroke* the KNN almost disregards *stroke* distances.

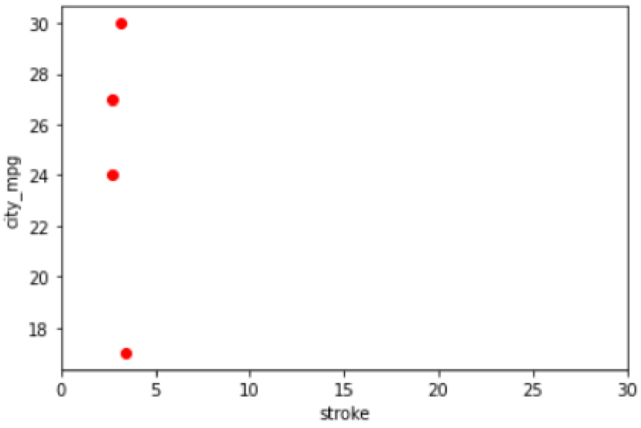


Figure 5: Raw dataset

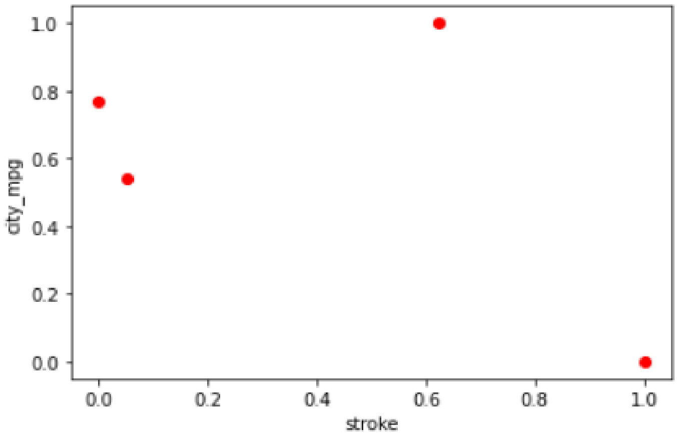


Figure 6:Scaled dataset

But when the features are of an equal importance KNN can correctly justify the dissimilarities between the samples.

Decision tree:

The last algorithm this research is going to operate with is the decision tree. It is another algorithm that is used both in classification and regression. Depending on the target variable, whether it is a continuous variable or a discrete value. Throughout this paper when talking about decision trees only the decision tree classification will be concerned.

The decision tree algorithm is used in predictive analysis and machine learning. It attempts to predict a target value based on split decisions made about features of the data set. The decision tree is flow-chart like structure. Example of a decision tree is shown in figure 7. Decision`s tree elements are:

- The root: is the tree topmost node and it is the only node that has no input branches.
- Internal nodes: are the nodes consisting both input and output branches. Each of the internal nodes is labelled with an input feature.
- Branches: represent the split based on internal nodes.
- Leaves - are bottom most elements and have contrary to the root, have no output branches. A leaf represents an answer – classification of a class.

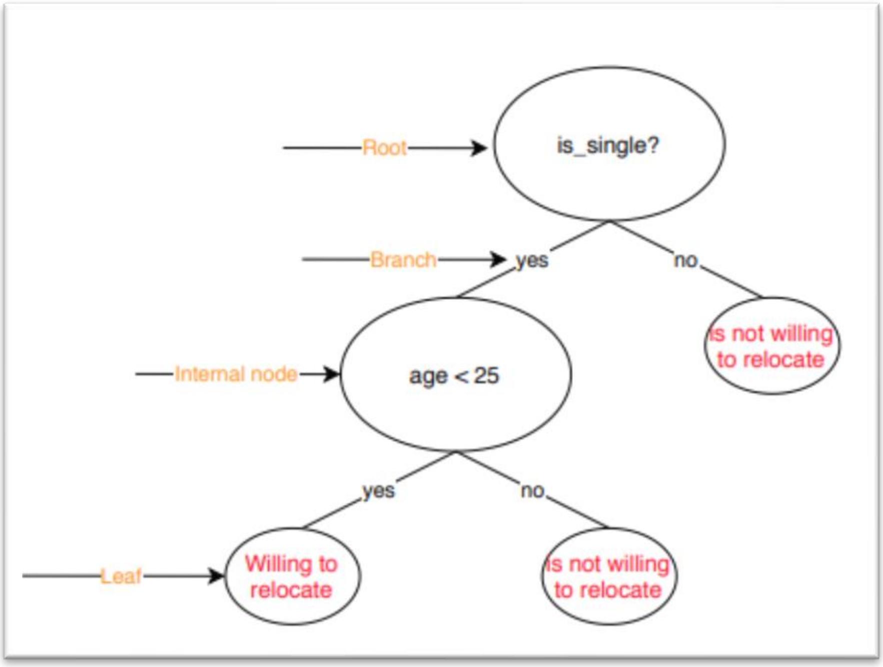


Figure 7: Decision Tree example

Decision tree is a greedy algorithm that undertakes decision based on best possible outcome in given moment. When selecting the feature for internal node from all of the candidates the one with most value for the tree is selected.

Split metrics:

Gini impurity: [8]Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset. In case of Gini impurity the closer the value after the split is to 0 the better.

GiniScore = $sum(pk * (1 - pk)$, where pk is the proportion of same class inputs present in a particular group.

Information gain: Information gain measure the purity of the outcome nodes after making a split. It measures how much has the entropy of the outcome node decreased compared to its parental state. The information gain is the difference between the entropy of previous node compared with the one after the split therefore the closer the value 1 the better.

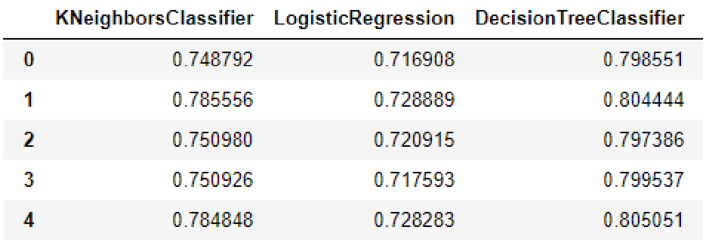
Information gain = $E_{n-}\ E_{n-1}$ *, where* E_n *is the entropy before splitting and* E_{n-1} *is the entropy after the split.*

Results

Titanic dataset:

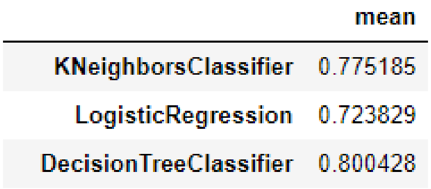
Classic methods

In order to achieve relevant results, there has been 1000 randomized runs for the titanic dataset. During each of the runs models got initialized and its parameters were randomly set. After having finished 1000 iterations the Decision Tree classifier was a leader within the classic models. The accuracy of the Decision Tree classifier was much better than KNN and Logistic regression on its own. In order to be sure that the results are reflecting the real state of the data and not coincidence the analysis of variance has been used. Figure 10 represent the p-values among all of the possible pairs. Very low value of p-val indicates that these means are different from other and the results are not random. For all of the pairs the p-value is near 0 which means that there is closely to 100% that the difference between the model is not random. This is satisfying as p-value below 0.05 or below 5% is a standard in academia and allows us to conclude that Decision Tree classifier works best.



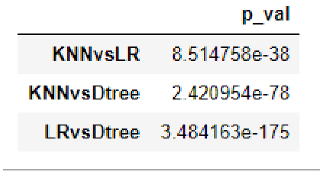
	KNeighborsClassifier	LogisticRegression	DecisionTreeClassifier
0	0.748792	0.716908	0.798551
1	0.785556	0.728889	0.804444
2	0.750980	0.720915	0.797386
3	0.750926	0.717593	0.799537
4	0.784848	0.728283	0.805051

Figure 8: Results of classic models



	mean
KNeighborsClassifier	0.775185
LogisticRegression	0.723829
DecisionTreeClassifier	0.800428

Figure 9: Means of classic models



	p_val
KNNvsLR	8.514758e-38
KNNvsDtree	2.420954e-78
LRvsDtree	3.484163e-175

Figure 10: P-values of classic machine learning models

Ensembled methods

Now it is time to try ensembled learning and try combining the models together. During this phase 3 new models has been created.

- KNN combined with Logistic Regression
- KNN combined with Decision Tree
- Logistic regression combined with Decision Tree

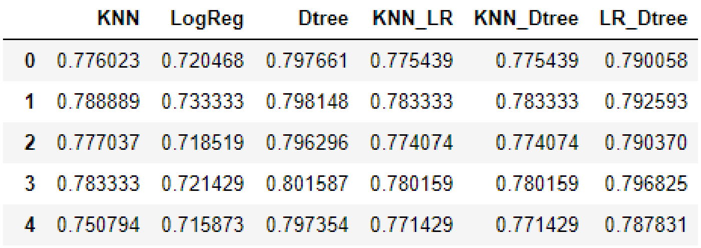
Each of the new models consist of two old models. The mean of probability within those two models gets calculated and then is treated like regular classification problem. The decision boundary still remains 50%/0.5 though it could be modified depending on problem domain. The output of the ensemble varies from 0 to 1 or from 0% to 100%. That allows to adjust the errors between models and uniqueness of each of them. When one of the models is certain about the class of specific sample then even the low value of probability will not change the result.

Example: Passenger with id=1 survived the crash. KNN is certain about the class and LR is not uncertain:

- KNN vote: 70% for passenger with id 1 to survive
- Logistic Regression vote: 50% for passenger with id 1 to survive (that means that logistic regression for that person with id 1 failed to determine as 50% is close to random)
- Ensemble vote: (70%+50%)/2 = 60% , that means that KNN and LR can trade information and help each other to determine the outcome of the class.

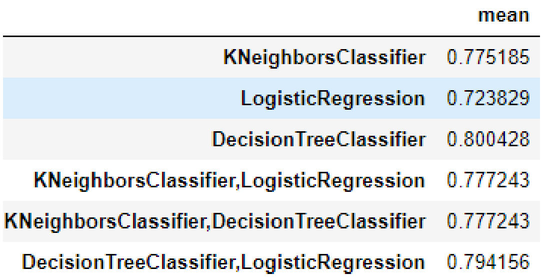
In the actual titanic dataset performance gain was present only when combining the Logistic regression and KNN algorithms. KNN have supplemented the Logistic regression in areas that it was failing before and increased its performance by around 0.2%. For ensembled involving Decision tree, this research has shown lack of improvement. Fig 12 and 13 represents analysis of variance and the means of all of the models.

Moreover the both KNN and Logistic regression ensembled with Decision tree presented the same results. That is also confirmed by figure 13 where p-value between those two ensembles is 1.



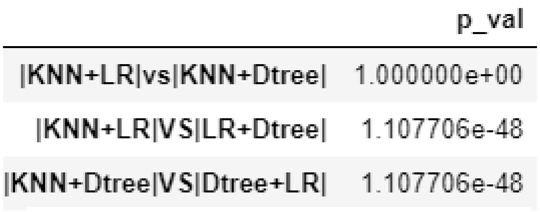
	KNN	LogReg	Dtree	KNN_LR	KNN_Dtree	LR_Dtree
0	0.776023	0.720468	0.797661	0.775439	0.775439	0.790058
1	0.788889	0.733333	0.798148	0.783333	0.783333	0.792593
2	0.777037	0.718519	0.796296	0.774074	0.774074	0.790370
3	0.783333	0.721429	0.801587	0.780159	0.780159	0.796825
4	0.750794	0.715873	0.797354	0.771429	0.771429	0.787831

Figure 11: Results of all the models



	mean
KNeighborsClassifier	0.775185
LogisticRegression	0.723829
DecisionTreeClassifier	0.800428
KNeighborsClassifier,LogisticRegression	0.777243
KNeighborsClassifier,DecisionTreeClassifier	0.777243
DecisionTreeClassifier,LogisticRegression	0.794156

Figure 12: Means of all machine learning models



	p_val
|KNN+LR|vs|KNN+Dtree|	1.000000e+00
|KNN+LR|VS|LR+Dtree|	1.107706e-48
|KNN+Dtree|VS|Dtree+LR|	1.107706e-48

Figure 13: p-values for ensembles.

Conclusion

What kind of machine learning works best with classification?

In case of titanic dataset the decision tree seemed to be the dominant from the 3 selected algorithms. Both KNN and Logistic regression also provided positive results. But clearly in case of titanic data set the greedy approach represented by Decision tree seemed to be the most efficient solution. Another interesting issue related with the titanic dataset is the number of samples. Perhaps KNN and LR would outperform the Decision tree having provided more samples. Checking different datasets is definitely future work related with this project.

Is there a performance gain using ensembled machine learning techniques?

The research allowed to gain 0.2% when assembling KNN and LR. It means that there are areas that those two algorithms supplement each other. On the other hand the decision tree turned out to be covering all of the input spaces better than KNN and LR. This is why no performance gain was present when combining the decision tree. Future work related with this research could issue combining more than 2 machine learning models and trying different options in order to find group of algorithms that supplement each other in a positive way How to ensure that the results are not random?

How to ensure that the results are not random and they can be trusted?

In order to achieve this research has randomize the creation of the machine learning models. During randomized runs parameters of the models has been randomly selected. Then after the results has been crafted the analysis of variance has been used. P-value allowed to understand the actual difference within the data. Very low value of p meant that the samples under test are different and “better” one can be determined.

References:

1. Opitz D, Maclin R. Popular Ensemble Methods: An Empirical Study. Vol 11.; 1999.
2. https://www.kaggle.com/c/digit-recognizer/data
3. https://www.kaggle.com/c/titanic/data
4. Guo G, Wang H, Bell D, Bi Y, Greer K. KNN model-based approach in classification. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2003;2888:986-996. doi:10.1007/978-3-540-39964-3_62
5. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
6. https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148#:~:text=We%20can%20call%20a%20Logistic,instead%20of%20a%20linear%20function.
7. https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052
8. https://en.wikipedia.org/wiki/Decision_tree_learning
9. Wasserstein RL, Lazar NA. The ASA's statement on p-values: context, process, and purpose. Am Stat. 2016;70:129–133.