Business understanding

Mushroom hunting is a hobby popular in Estonia, Russia, Poland and many other countries. It implies going a far distance from cities to some forest and gathering edible mushrooms (to eat them afterwards). It is common to spend 3-5 or more hours in the forest and to gather about 10 liters of mushrooms, although the volume depends greatly on how lucky and trained the hunter is. Of course, not all mushrooms that can be found in the forest are edible, and this is a major inconvenience, since an inexperienced gatherer could pick all the mushrooms and later (after consulting with experienced gatherers) find out that some or even most of them are not edible.

Given the small dataset we would not try to build a system that will be sufficient to determine if a mushroom is safe to eat, since wrongly assuming that a mushroom is safe could end up badly (need for hospitalization). This may only change if we find a larger dataset that will be homogeneous-enough with ours. Instead, we would like to build a system (probably an application) that will allow an inexperienced mushroom hunter to gain some experience and to learn how inedible mushrooms look, so that he/she will not pick them up. This means that afterward consulting is still necessary. Given that, in the long run the system could ask the hunters to provide their data and therefore become better; possibly one day eliminating the need of consulting. The system should determine if a mushroom is edible in under a second on a personal computer, learning time is not bounded strictly (a few hours on a personal computer). Usability questions (for example: "Will there be an Android app for this and will it be easy to use to beginner hunters?") are currently outside of the scope of the project.

Our project success criteria: we hope to create a system that potentially could be used by inexperienced mushroom hunters to gain some advantage. The instructors could help us evaluate if the system would be sufficient for this.

In this project we will use Kaggle dataset (https://www.kaggle.com/uciml/mushroom-classification) and our personal computers. In a long run, we could use help of experienced mushroom hunters to better predict edibility. We will spend some time seeking bigger suitable datasets in the Internet. If we success finding one, we will use it.

Project should hopefully be done until deadline.

There is a risk that a dataset this small will not be sufficient to meet the goals. As a "workaround", we will publish as it is but with a feature to add more data (discussed earlier). Also, there is a risk that our clusterization goal will fail (since we do clusterization and not classification, which is much simpler). The "workaround" will be as with the first risk.

Things that grow in forest and have mycelia are mushrooms, the project is centered about such mushrooms.

Mushroom hunter/shroomer/gatherer/hunter - a person who is involved with activity described in the project overview. This activity is called mushroom hunting/shrooming/gathering/hunting. Edible mushroom is safe to eat if processed properly, no important health concerns present. Inedible means not edible.

Cap, gill, stalk, veil, ring, spore – all parts of mushroom. Properties of cap are: shape, surface, color. Properties of gill are: attachment, spacing, size, color. The mushroom can have or not have bruises and may have a type of odor.
Habitat – where a species of a mushroom usually grows. Population refers to how many mushrooms there are in a group (it can be either an individual mushroom, or many, but sparsely, or abundant, etc.).

Costs and benefits - our project is not for money, but if we need profit, we may consider making our system paid. Costs – since our project is run 2 enthusiasts, there will not be any significant costs to the project.

Our model should achieve the following: system correctly identifies at-least 80% of edible mushrooms (true positive rate), while correctly identifying at-least 90% of not edible mushrooms (false positive rate not more than 10%). The system can at-least vaguely classify the mushrooms so that a manual labeler could label all the clusters; so that our system could actually know species of mushrooms (and display them to the users). The system knows the most significant properties to look for to tell edible and inedible mushrooms apart (in future, could be used to create entry-level shroomer guide).

Hard to tell how to estimate other variables of success of our models.

Data understanding

Gathering data:

We wanted to come up with our own project, not using idea provided by course instructors, so we were seeking for inspiration, challenge and interesting biological data for classification that we wanted to explore, and make a practical use of our knowledge from IDS course. Of course, it would be great if we could deliver some benefits for people. We found interesting data from online community of data scientists and machine learning practitioners Kaggle. Dataset named “Mushroom Classification” (https://www.kaggle.com/uciml/mushroom-classification).

It is good, that data was in .csv file, because we are already familiar with importing it via jupyter notebook. Data availability is on very high level, it is not only open for everybody, but also pretty popular if we watch on it’s rating on Kaggle. It was easy to find this data using keyword search on Kaggle webpage. Dataset was posted by UCI Machine Learning in year 2016.

Describing data:

At the end of our search we have now a csv file that contains 8124 rows and 23 columns. Each row is a mushroom and each column consist of mushroom’s properties. Columns describe shape and color of

different parts of mushroom, where mushroom may be found and how big is the population. Also there are columns that provide information whether mushroom has specific part or not, other describe mushroom as a whole: odor and bruices. And main column, that we need for our project is “class” which define mushroom as edible or poisonous.

Exploring data:

All data in dataset is categorical. Each sell describes some property of mushroom using single letter (string in python). In order to use machine learning methods we are planning to change all categorical columns into binary features using one-hot-encoding.

Out of all entries/rows, 52% are flagged as edible and 48% as poisonous. In the data, all veil types were partial (whole column has only one value for all row), which means that this attribute is not useful. For some properties, there was one dominant value: gill-attachment (97%), gill-spacing (84%), gill-size (69%), stalk-surface-above-ring (64%), stalk-surface-below-ring (61%), veil-color (98%), ring-number (92%). We consider dropping some of them (especially those where dominant parameter is >90% common), since we expect that they may not carry much information, so that we could make the model and the system depend on less characteristics of a mushroom. We may decide in the future to drop this column.

We have verified that data contains no missing or wrong cells.

Verifying data quality:

We are confident, that this data suits very well for our project. The only disadvantage of this dataset is it’s size. We consider the probability of finding larger dataset and using them together or just expanding existing dataset with new rows. However, it is not our main goal and we are able to complete this project using only this “Mushroom Classification” dataset. Previous report on “exploring data” task proved that data quality is on high enough level.

Planning your project

Tasks and methods:

* create an initial report (done)

* use Ridge and/or Lasso regression to find strongest correlations between edibility and other features

* Find strongest corellations between features other than edibility (may help to find unimportant features)

* Try creating a good machine learning model for predicting edibility

* Try creating a good random forest model for predicting edibility

* Attempt to remove some unnecessary features without sacrificing accuracy.

* Make a command line interface for predicting edibility from features/properties

It is hard to estimate how much time each task will take, but we hope that all the tasks are comparable in difficulty. We would like to split the work about evenly.

Tools:

Python version 3

Extra libraries for python (for sure pandas)

Jupiter notebook.

Git hub (for communication and working on distance)