Report:

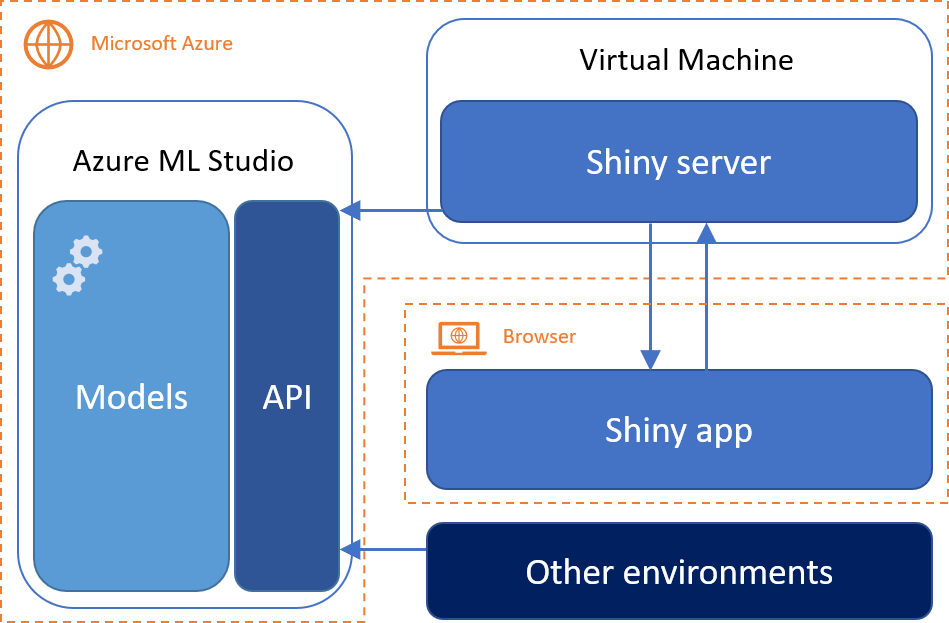
A report that contains at least the following information:

* Data exploration, description, preparation and analysis of the dataset variables (distribution of the variables, outlier detection, correlation analyses, …).
* Model creation: explanation of the models created, the reason behind choosing one algorithm or another, as well as the parameters used.
* Evaluation of the models, and explanation of the results.
* Deployment: explanation of how the model has been deployed and how can be used in a real environment.

# Deployment

## Overview

The following figure shows a schema of the whole platform with its main components that have been developed. In the next sections, each of these components will be describe, as well as the usability of the platform and consumption of the models.



## Infrastructure

For the development of this assignament, it has been used several technologies, frameworks and services hosted in Microsoft Azure. There are two main components deployed in this cloud platform.

On the one hand, we have used **Azure Machine Learning studio** in order to deploy different models and make them accessible on the part of third parties from other environments by consuming a public API. The use of this API is covered in the next section.

On the other hand, we have deployed a **virtual machine** with Ubuntu 14.04, installed the necessary packages and dependencies (such as R and shiny) and hosted a shiny server listening to the port 3838 containing our shiny app. As we have developed the shiny app using Github as control version platform, the deployment of the application into the virtual machine, better said the shiny server, is quite straightforward using git commands once connected to the virtual machine through the ssh protocol. By using this approach, the deployment of the shiny app for its use is not necessary on the part of the users and can be reached publicly through the following link:

<http://mushrooms4all.northeurope.cloudapp.azure.com:3838/>

## Usability and interface

Regarding the use of the platform and consumption of the models, we have provided two different ways.

The first one consists of the use of a shiny app in which the user can perform different actions. A brief explanation of the four parts of the application is given below.

* The first tab of the application is related to the prediction of the class of unknown mushrooms (e.g. edible or poisonous). In this tab, the user can either insert a custom mushroom to be predicted or upload a csv file containing the attributes of one mushroom per row. Immediately, the results of the predictions appear at the end of the page. It is important to note that this csv file cannot contain the attribute class as this is the attribute supposed to be predicted.
* The second tab of the application is focused on the exploration of a set of mushrooms. As for the predictions, the user have to upload a csv file containing one row per mushroom which contains the attributes of the mushrooms in the columns. In this case, the attribute class have to be included in order to visualize all the charts correctly and make the most of the idioms.
* In the third tab, a brief description of the API, input and output formats and some code examples are included. The main point here is the key value pairs on the top of the page, where the user can find the api key token and the endpoint of the resource where the model is located.
* Lastly, the fourth tab is reserved to present the members of the team behind the platform.

The second way of interacting with the platform and consuming the models to make predictions of unknown mushrooms is through a publicly available API built on top of Azure Machine Learning studio. By this way, the access of the models from any environment able to make http requests is guaranteed. There are two main concepts to explain here.

* Security and access. The models are located in the following endpoint through a POST request.

https://europewest.services.azureml.net/workspaces/9f6cbb787ae24af8bd2cb1fe82f331a7/services/2916893a381f4fdcaeb283d15618340b/execute?api-version=2.0&details=true

To be allowed to consume the model, the client has to include the following bearer token to the header of the request.

stBIvZeMi7nCKQyDTKx2rC3CaCWzmyKgE7o0qh6jhcE5rseLXxFhuqZd7hSPoI40KKV+80pIc/GtsF+Wqkzs7A==

* Payload. The data set of unknown mushrooms must be serialized into a JSON object with the following format, where the Values property contains one array of values per mushroom.

{

"Inputs": {

"input1": {

"ColumnNames": [

"cap-shape",

"cap-surface",

"cap-color",

"bruises",

"odor",

"gill-attachment",

"gill-spacing",

"gill-size",

"gill-color",

"stalk-shape",

"stalk-root",

"stalk-surface-above-ring",

"stalk-surface-below-ring",

"stalk-color-above-ring",

"stalk-color-below-ring",

"veil-type",

"veil-color",

"ring-number",

"ring-type",

"spore-print-color",

"population",

"habitat"

],

"Values": [

[

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value"

]

]

}

},

"GlobalParameters": {}

}

In order the interpret the predictions, a response of the request with this format must be expected. The last two values in each array corresponds with the predicted class and the probability, respectively.

{

"Results": {

"output1": {

"type": "DataTable",

"value": {

"ColumnNames": [

"cap-shape",

"cap-surface",

"cap-color",

"bruises",

"odor",

"gill-attachment",

"gill-spacing",

"gill-size",

"gill-color",

"stalk-shape",

"stalk-root",

"stalk-surface-above-ring",

"stalk-surface-below-ring",

"stalk-color-above-ring",

"stalk-color-below-ring",

"veil-type",

"veil-color",

"ring-number",

"ring-type",

"spore-print-color",

"population",

"habitat",

"Scored Labels",

"Scored Probabilities"

],

"ColumnTypes": [

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"String",

"Numeric"

],

"Values": [

[

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"value",

"0"

]

]

}

}

}

}