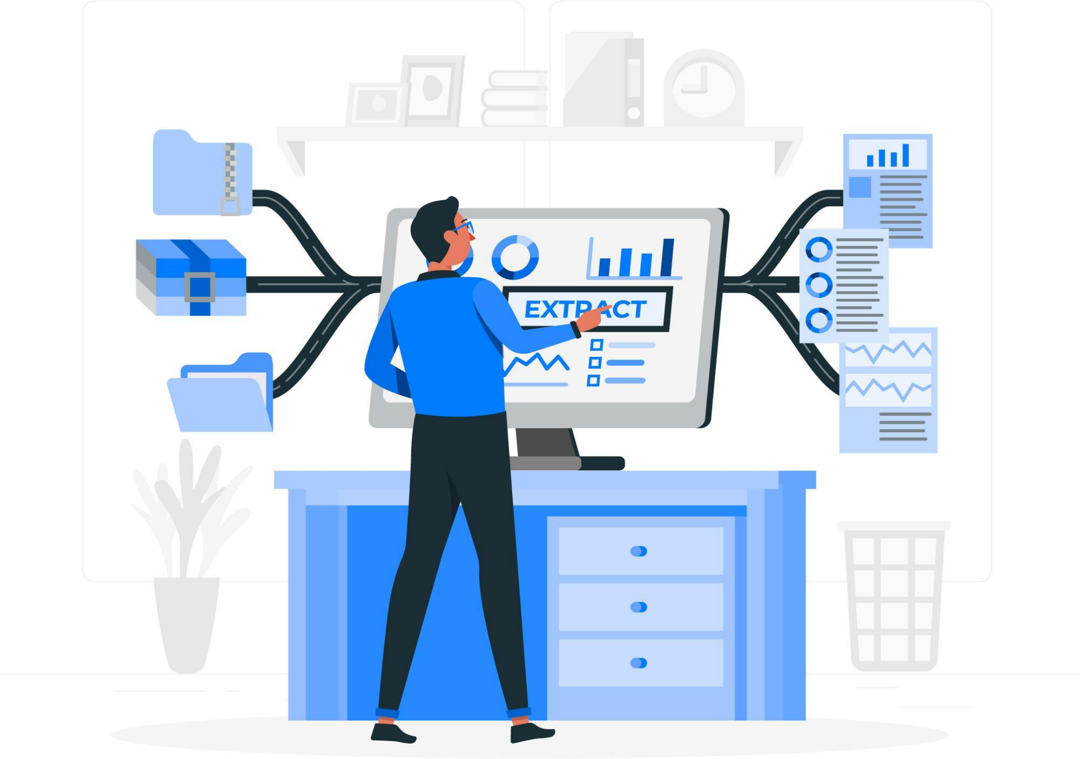
Veasualizer

Batch 23:
P Yuvaraj (18N31A05H2)
M Lakshman (18N31A05D1)
SK Anwar (17N31A05L4)

Under the guidance of
Mrs. P Honey Diana



Abstract

Veasualizer is a modern data visualization and a “ML” tool.

Using Veasualizer a user with zero knowledge can make beautiful visualizations: Graphs and plots. Not just that, Veasualizer has features using which users can Train, Test and Draw predictions from data by just uploading their dataset to Veasualizer.

After uploading dataset, User will have freedom to choose different algorithms suitable for their data and can even choose different parameters for training the dataset on which accuracy of the model depends on.

At the end user will get an Exploratory Data Analysis Report (EDA) for their dataset.

Objective next

Objective

The whole idea of this project is based on its user base, that is people who are going to use it.

Our main objective of Veasualize is to make stuff, which is complex, look easy and useful to everybody.

Just by some clicks and uploading datasets, Users will be able to plot different plots like: Scatter plot, Bar graphs, Bubble plots, Box plots, Dot plots, Pie charts and more and ML algorithms include Regression and Classification algorithms like: Linear, Polynomial, Logistic etc

Literature survey next

Literature Survey

We tried to find a project or tool similar to our project but, There are popular tools like Power BI, Tableau, Which can only visualize user's dataset that too, These tools compresses a lot of complex features and make simple task of data visualization tough.

And there are no tools which can take pre-processed data as input from user and use ML algorithms to forecast or draw predictions from it (Train - Test - Predict).

Veasualizer, Our proposed system is a simple tool which has all the problems explained above, full filled.

Proposed system next

Proposed System

Veasualizer, Our proposed system is a simple, user friendly and powerful tool.

Veasualizer is a software that can generate plots and graphs from the dataset uploaded by user, and these can be downloaded in jpg or png format. Apart from visualizing user's dataset, Here using this tool users can also draw predictions.

In our project (proposed system), we are implementing:

1. Bar Graph, Histogram, Scatterplot, Area plot, Pie plot and more
2. ML algorithms include,
   2.1. Regression: Linear, Logistic and Polynomial
   2.2. Classification: Decision Tree, Random Forest and SVM

Requirements next

Requirements

Software Requirements

The development and deployment of the application requires the following general and specific minimum requirements for software:

1. Programming Language Translators - Python 3.7 and above.

2. IDE – Spyder or Visual studio code.

3. Operating System used Windows 7 or above.

Hardware Requirements

The development and deployment of the application requires the following general and specific minimum requirements for hardware:

1. Processor- (32-bit or 64-bit)

2. RAM (4 GB or above)

3. Hard disk: 450MB

UML

●UML stands for **Unified Modeling Language**. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

●The goal is for UML to become a common language for creating models of object oriented computer software.

●In its current form UML comprised of two major components : a Meta-model and a notation.

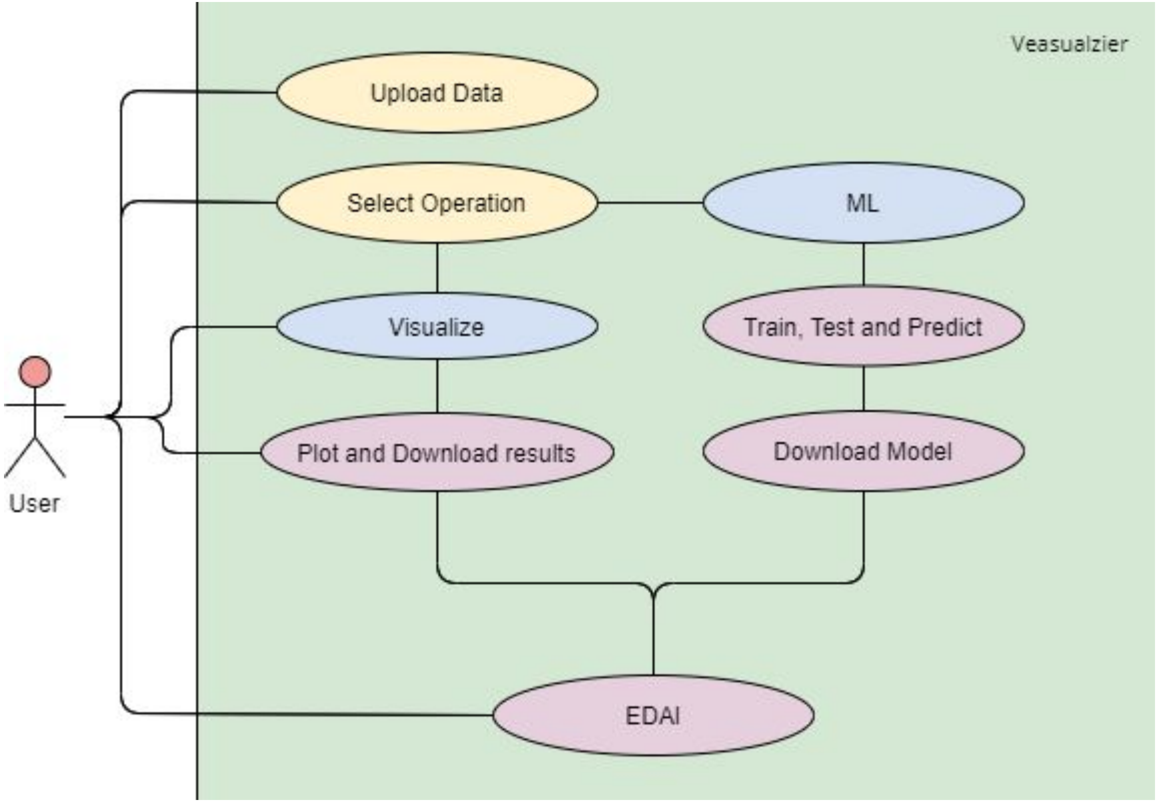
●The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Types of UML:

1. Use case diagram
2. Class diagram
3. State diagram
4. Collaboration Diagram
5. Activity Diagram

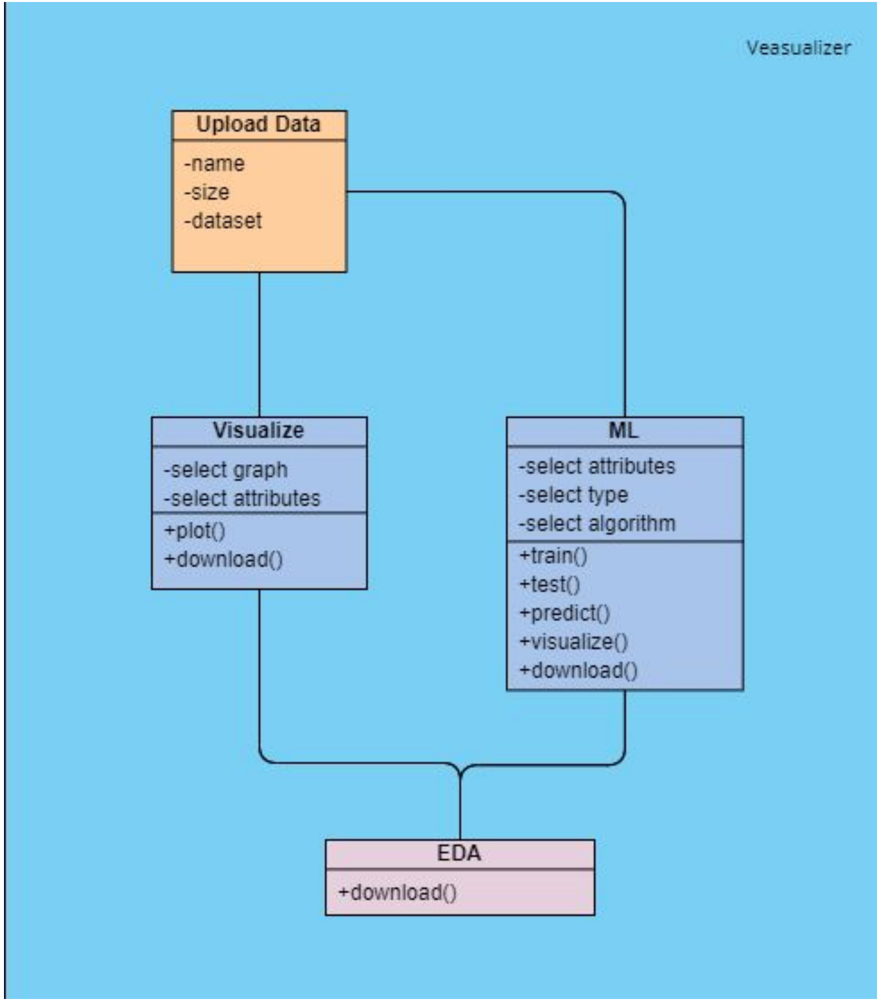
Use Case Diagram

- A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.
- Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.



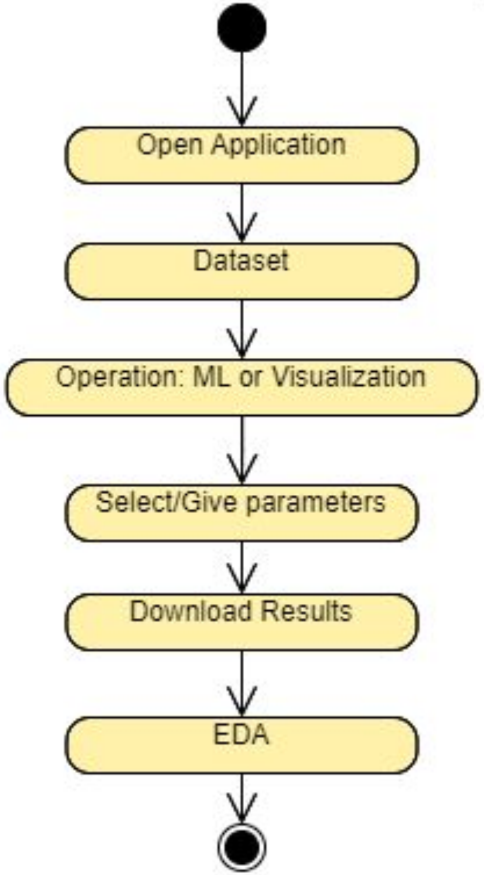
Class Diagram

- The class diagram is used to refine the use case diagram and define a detailed design of the system.
- The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes.
- The relationship or association between the classes can be either an "is-a" or "has-a" relationship.
- Each class in the class diagram may be capable of providing certain functionalities.



State Diagram

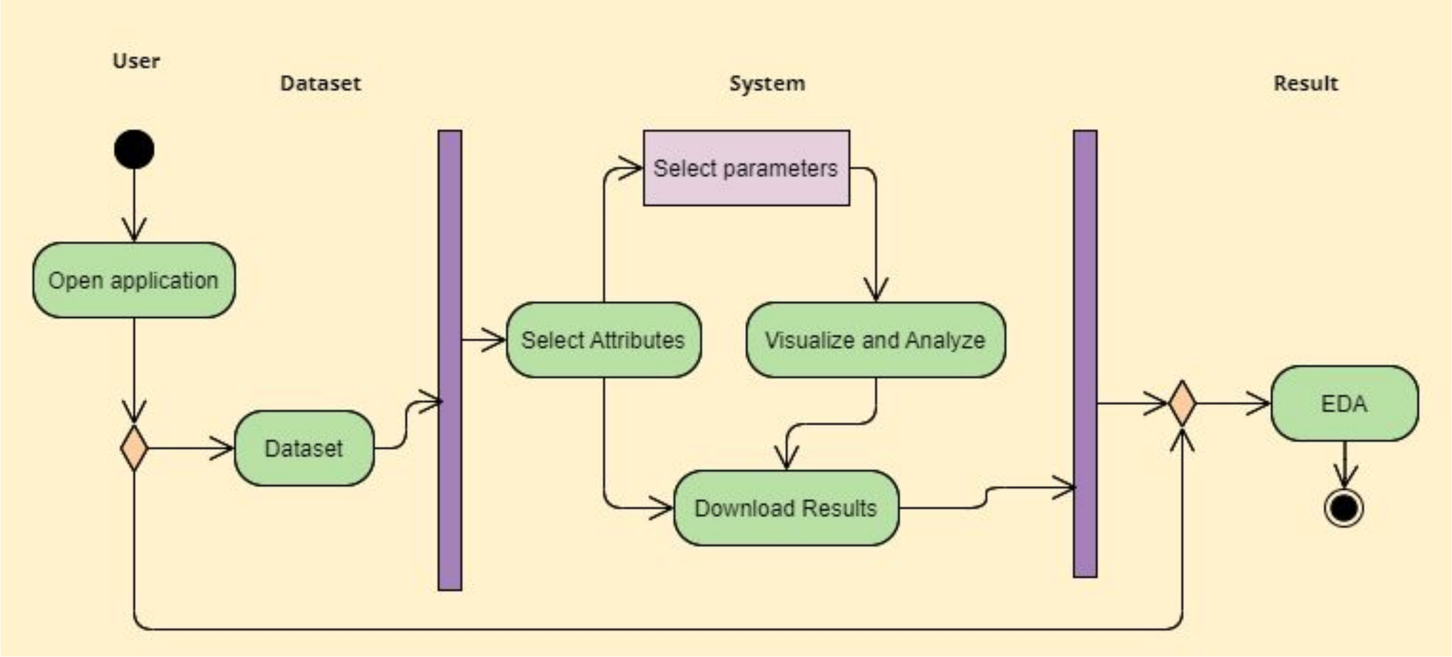
- A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle.
- Objects in the system change states in response to events.
- In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.



Veasualizer

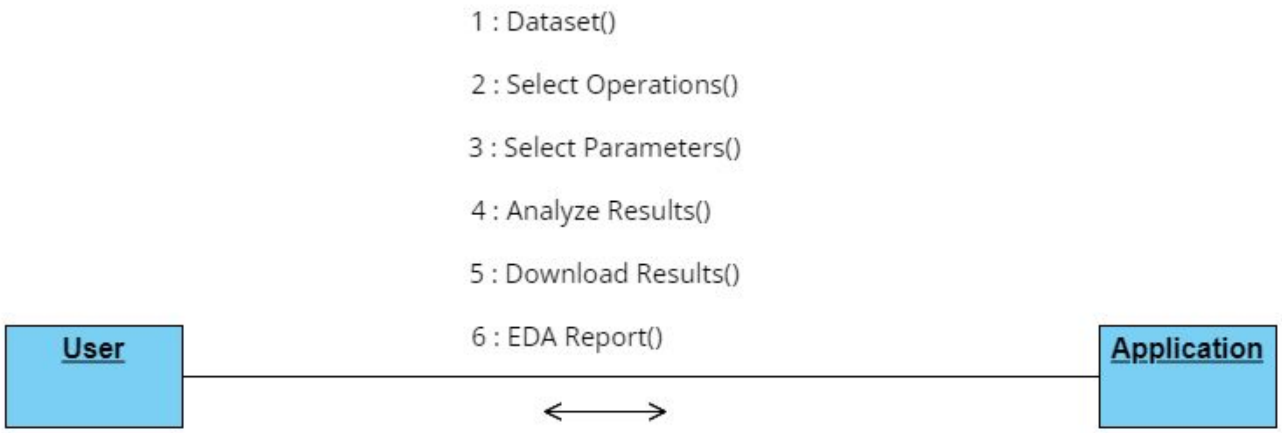
Activity Diagram

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions



Collaboration Diagram

- A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.



Implementation

- We chose the base technology as “Python”, but in Python we have many frameworks/technologies which can be used for this project’s implementation. We had 3 options to choose from:
  1. Flask 2. Django 3. Streamlit

- All these 3 frameworks are used to build web apps but their use-case is completely different though. There are some advantages and disadvantages for each of this framework.

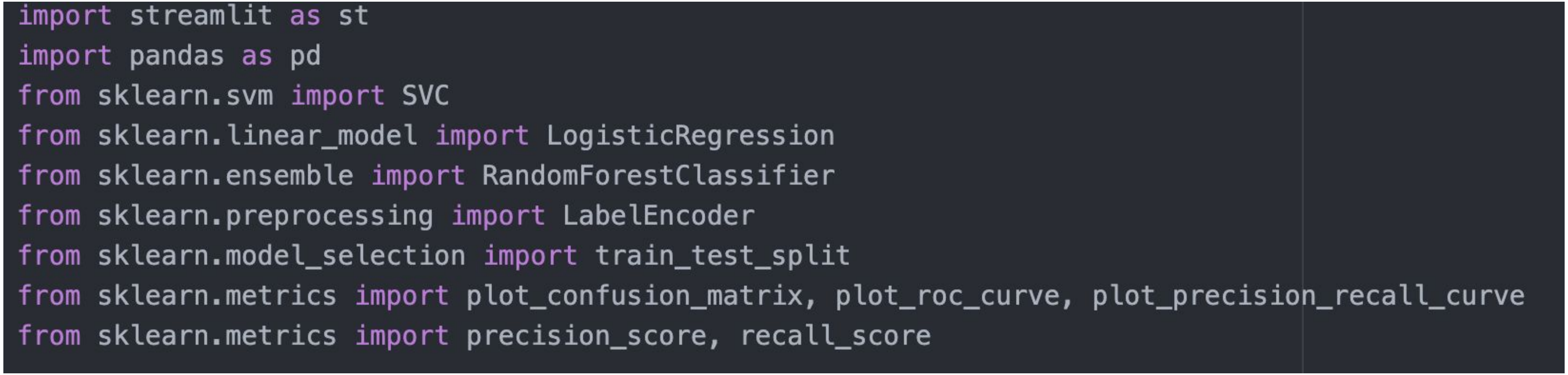
django Streamlit Flask

Flask Vs Django Vs Streamlit

- Flask provides support for API while Django doesn’t have any support for API.
- Streamlit turns data scripts into shareable web apps in minutes. All in pure Python. No front-end experience required.
- **Use Streamlit** if you want a structured data dashboard with many of the components you’ll need already included. Use Streamlit if you want to build a data dashboard with common components.
- **Use Flask** if you want to build a highly customized solution from the ground up and you have the engineering capacity.
- **Use Django** if your project requirements are static and most important thing for you is time complexity and when you need a production ready-framework.

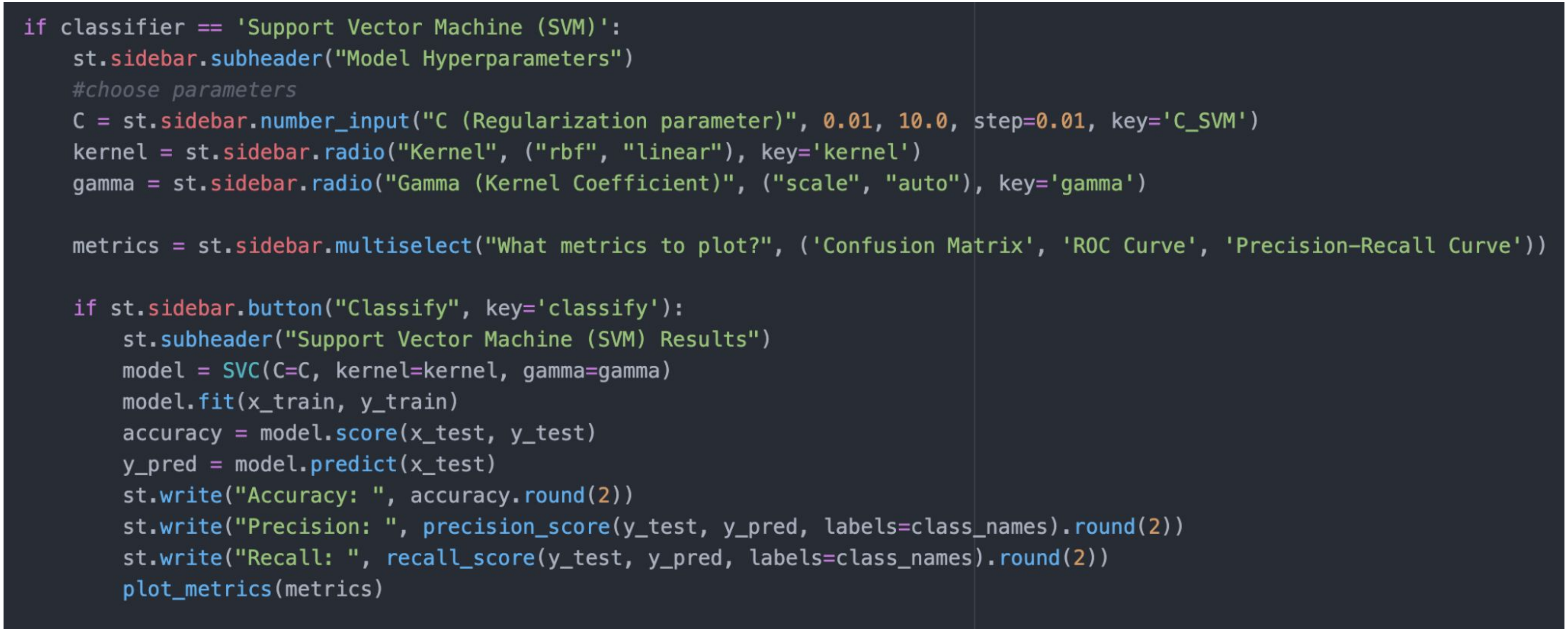
Sample Coding:

Modules/ Packages imported:



```python
import streamlit as st
import pandas as pd
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import plot_confusion_matrix, plot_roc_curve, plot_precision_recall_curve
from sklearn.metrics import precision_score, recall_score
```

Choose Classifier:

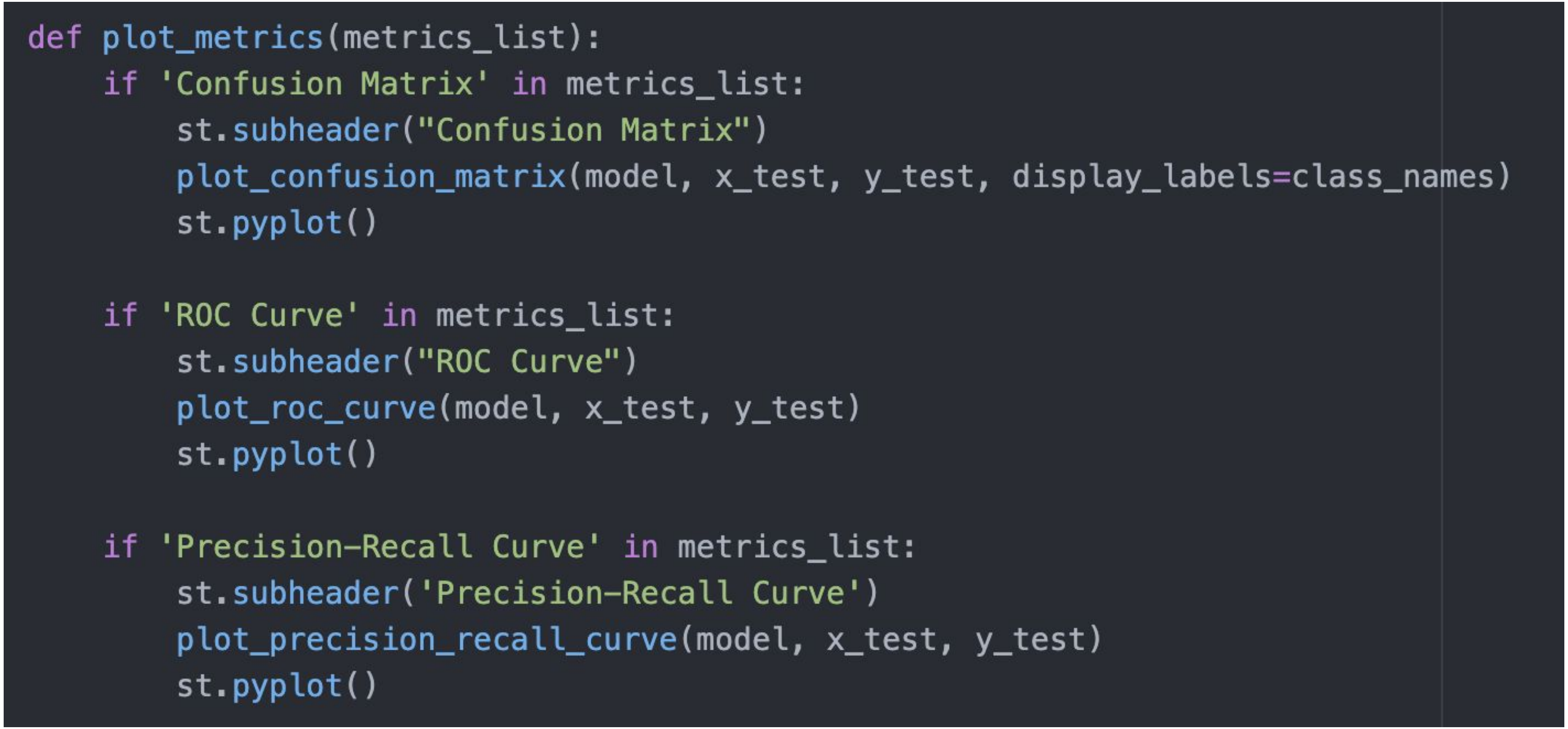


```
if classifier == 'Support Vector Machine (SVM)':
    st.sidebar.subheader("Model Hyperparameters")
    #choose parameters
    C = st.sidebar.number_input("C (Regularization parameter)", 0.01, 10.0, step=0.01, key='C_SVM')
    kernel = st.sidebar.radio("Kernel", ("rbf", "linear"), key='kernel')
    gamma = st.sidebar.radio("Gamma (Kernel Coefficient)", ("scale", "auto"), key='gamma')

    metrics = st.sidebar.multiselect("What metrics to plot?", ('Confusion Matrix', 'ROC Curve', 'Precision-Recall Curve'))

    if st.sidebar.button("Classify", key='classify'):
        st.subheader("Support Vector Machine (SVM) Results")
        model = SVC(C=C, kernel=kernel, gamma=gamma)
        model.fit(x_train, y_train)
        accuracy = model.score(x_test, y_test)
        y_pred = model.predict(x_test)
        st.write("Accuracy: ", accuracy.round(2))
        st.write("Precision: ", precision_score(y_test, y_pred, labels=class_names).round(2))
        st.write("Recall: ", recall_score(y_test, y_pred, labels=class_names).round(2))
        plot_metrics(metrics)
```

Choosing metrics:

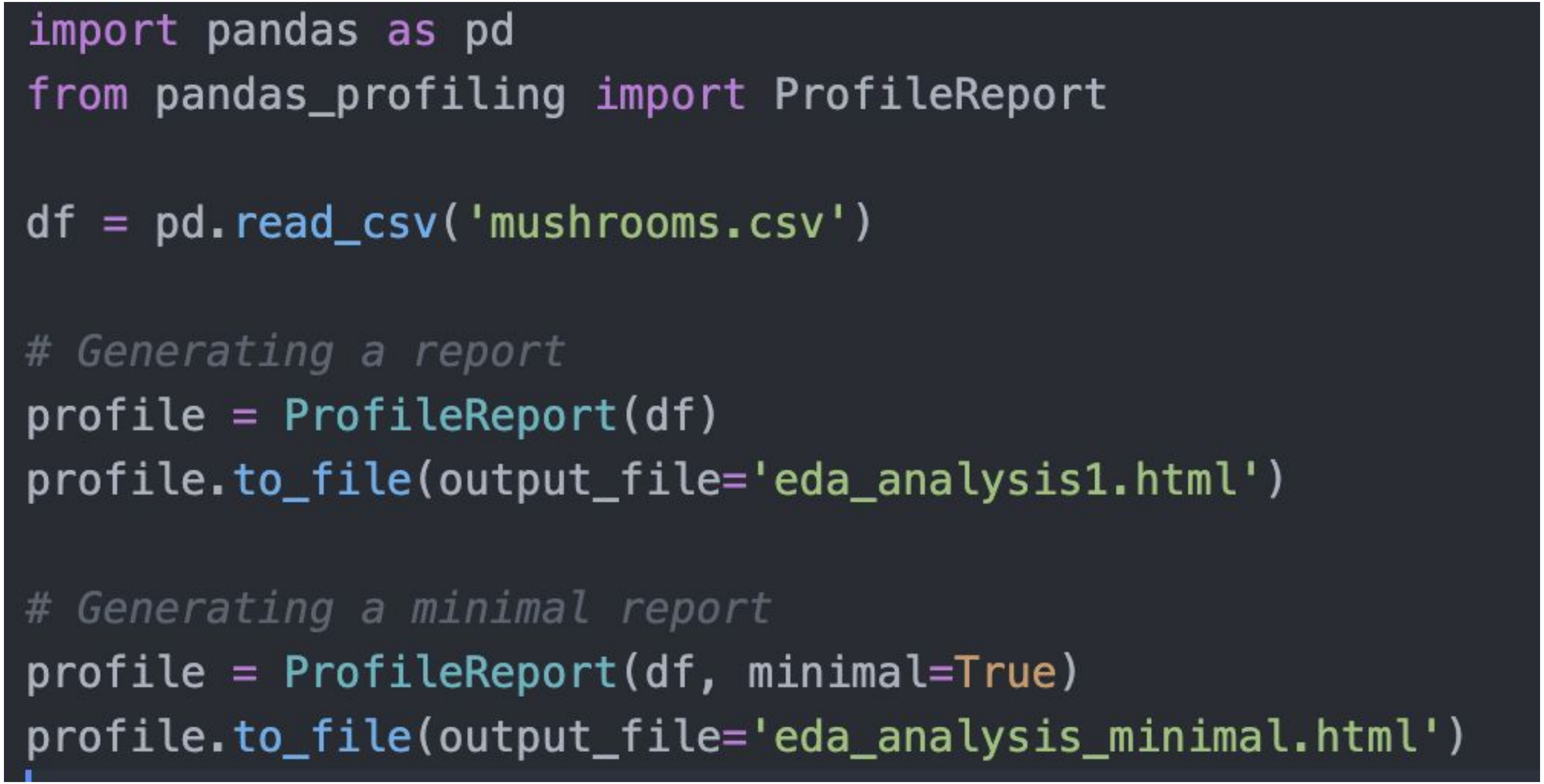


```python
def plot_metrics(metrics_list):
    if 'Confusion Matrix' in metrics_list:
        st.subheader("Confusion Matrix")
        plot_confusion_matrix(model, x_test, y_test, display_labels=class_names)
        st.pyplot()

    if 'ROC Curve' in metrics_list:
        st.subheader("ROC Curve")
        plot_roc_curve(model, x_test, y_test)
        st.pyplot()

    if 'Precision-Recall Curve' in metrics_list:
        st.subheader('Precision-Recall Curve')
        plot_precision_recall_curve(model, x_test, y_test)
        st.pyplot()
```

EDA Report Generation:



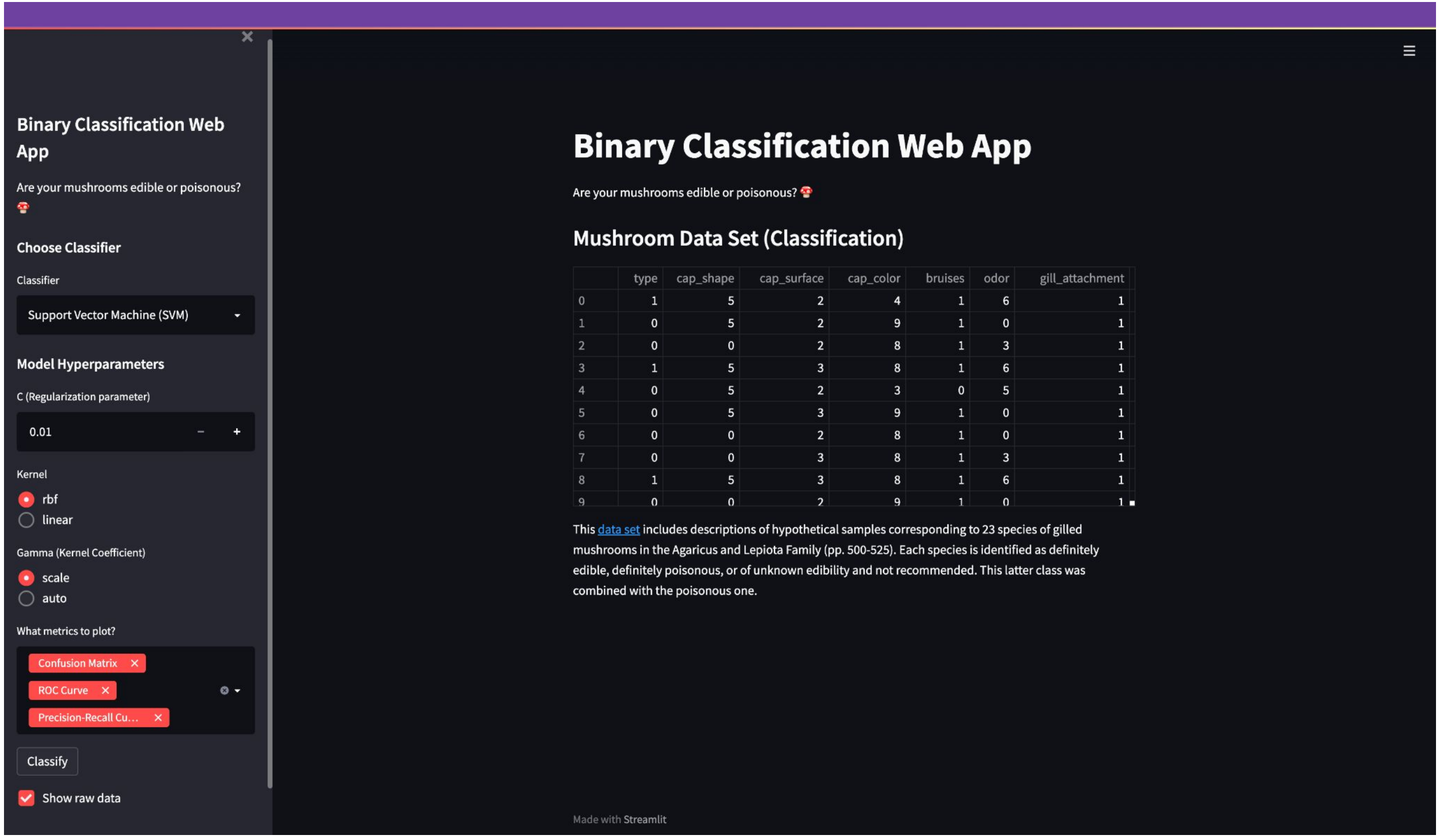
```python
import pandas as pd
from pandas_profiling import ProfileReport

df = pd.read_csv('mushrooms.csv')

# Generating a report
profile = ProfileReport(df)
profile.to_file(output_file='eda_analysis1.html')

# Generating a minimal report
profile = ProfileReport(df, minimal=True)
profile.to_file(output_file='eda_analysis_minimal.html')
```

Output Screens:



Binary Classification Web App

Are your mushrooms edible or poisonous? 🍄

Choose Classifier

Classifier: Support Vector Machine (SVM)

Model Hyperparameters

C (Regularization parameter): 0.01

Kernel

● rbf

○ linear

Gamma (Kernel Coefficient)

● scale

○ auto

What metrics to plot?: Confusion Matrix, ROC Curve, Precision-Recall Cu…

☑ Show raw data

Binary Classification Web App

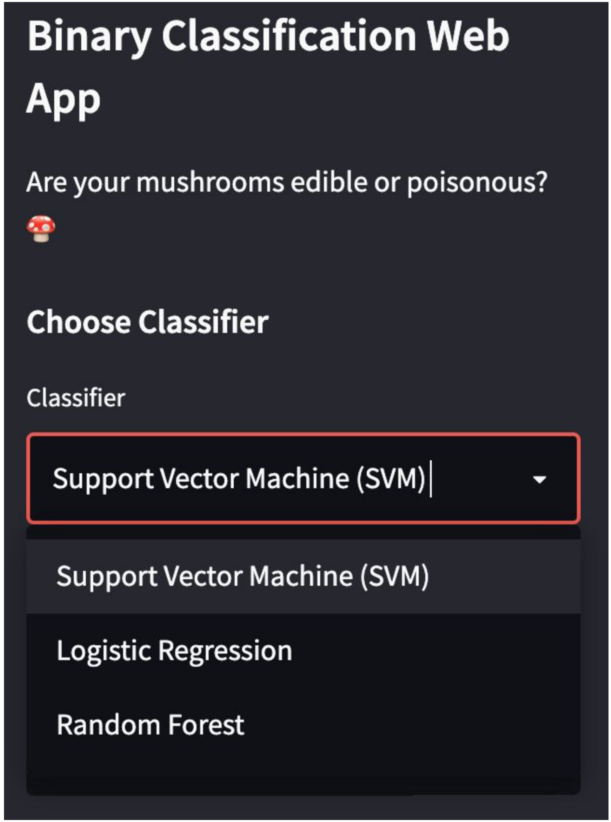
Are your mushrooms edible or poisonous? 🍄

Mushroom Data Set (Classification)

	type	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment
0	1	5	2	4	1	6	1
1	0	5	2	9	1	0	1
2	0	0	2	8	1	3	1
3	1	5	3	8	1	6	1
4	0	5	2	3	0	5	1
5	0	5	3	9	1	0	1
6	0	0	2	8	1	0	1
7	0	0	3	8	1	3	1
8	1	5	3	8	1	6	1
9	0	0	2	9	1	0	1

This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one.

Choose Algorithm:



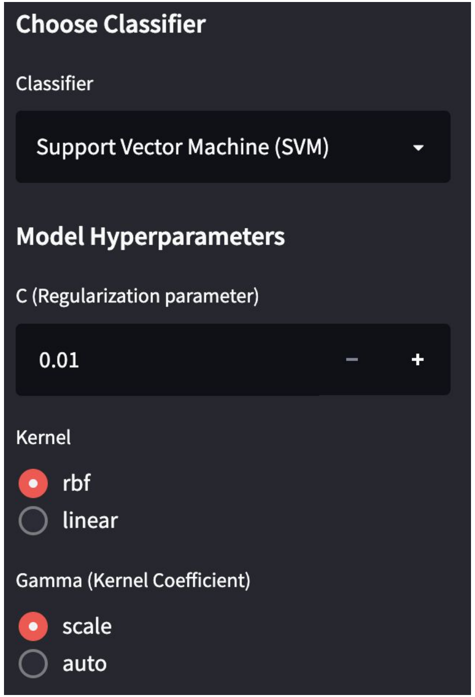
Binary Classification Web App

Are your mushrooms edible or poisonous? 🍄

Choose Classifier

Classifier: Support Vector Machine (SVM)

If SVM:



Choose Classifier

Classifier: Support Vector Machine (SVM)

Model Hyperparameters

C (Regularization parameter): 0.01

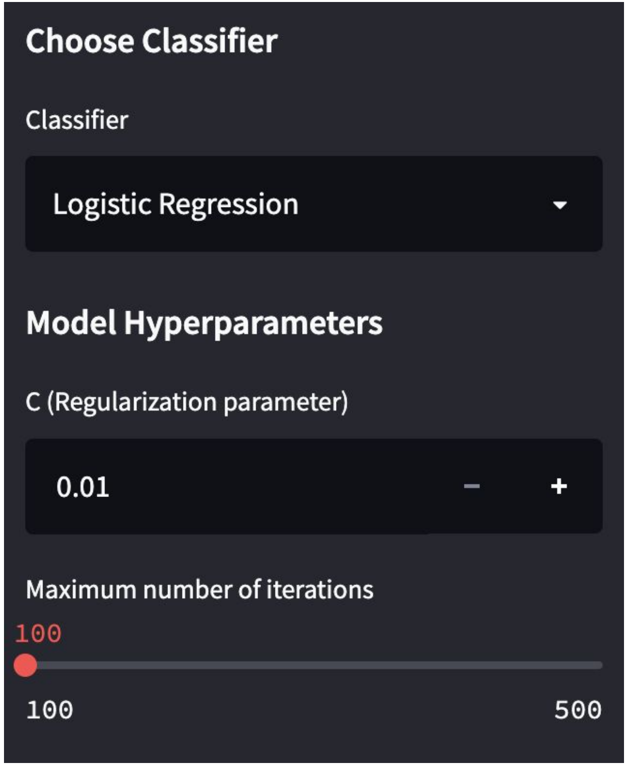
Kernel

● rbf
○ linear

Gamma (Kernel Coefficient)

● scale
○ auto

If Logistic Regression:



Choose Classifier

Classifier

Logistic Regression

Model Hyperparameters

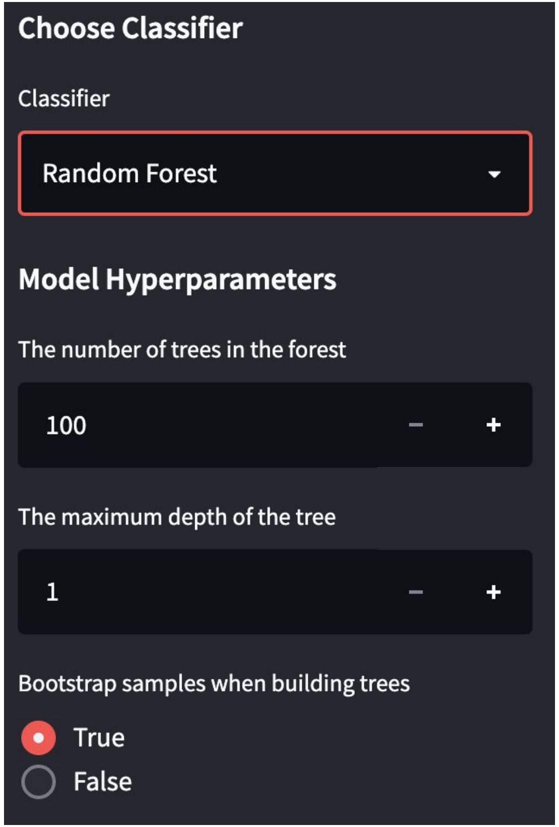
C (Regularization parameter)

0.01

Maximum number of iterations

100

If Random Forest:



Choose Classifier

Classifier

Random Forest

Model Hyperparameters

The number of trees in the forest

100

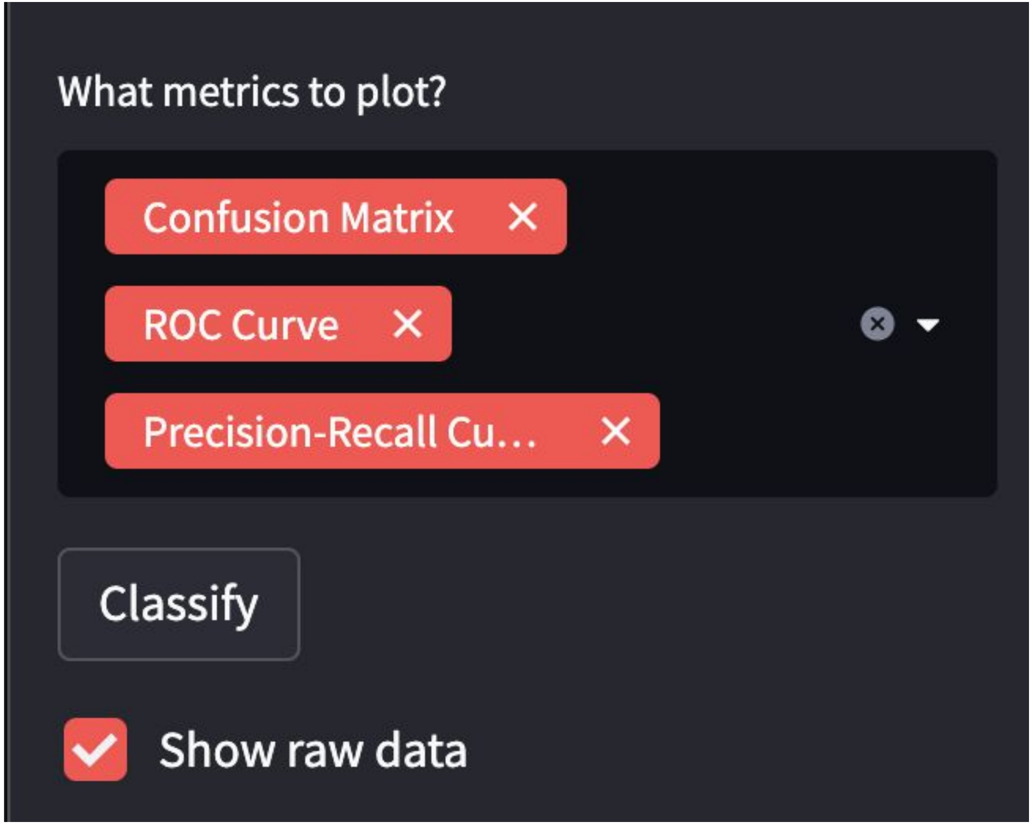
The maximum depth of the tree

1

Bootstrap samples when building trees

● True
○ False

Choose metrics to plot:



What metrics to plot?

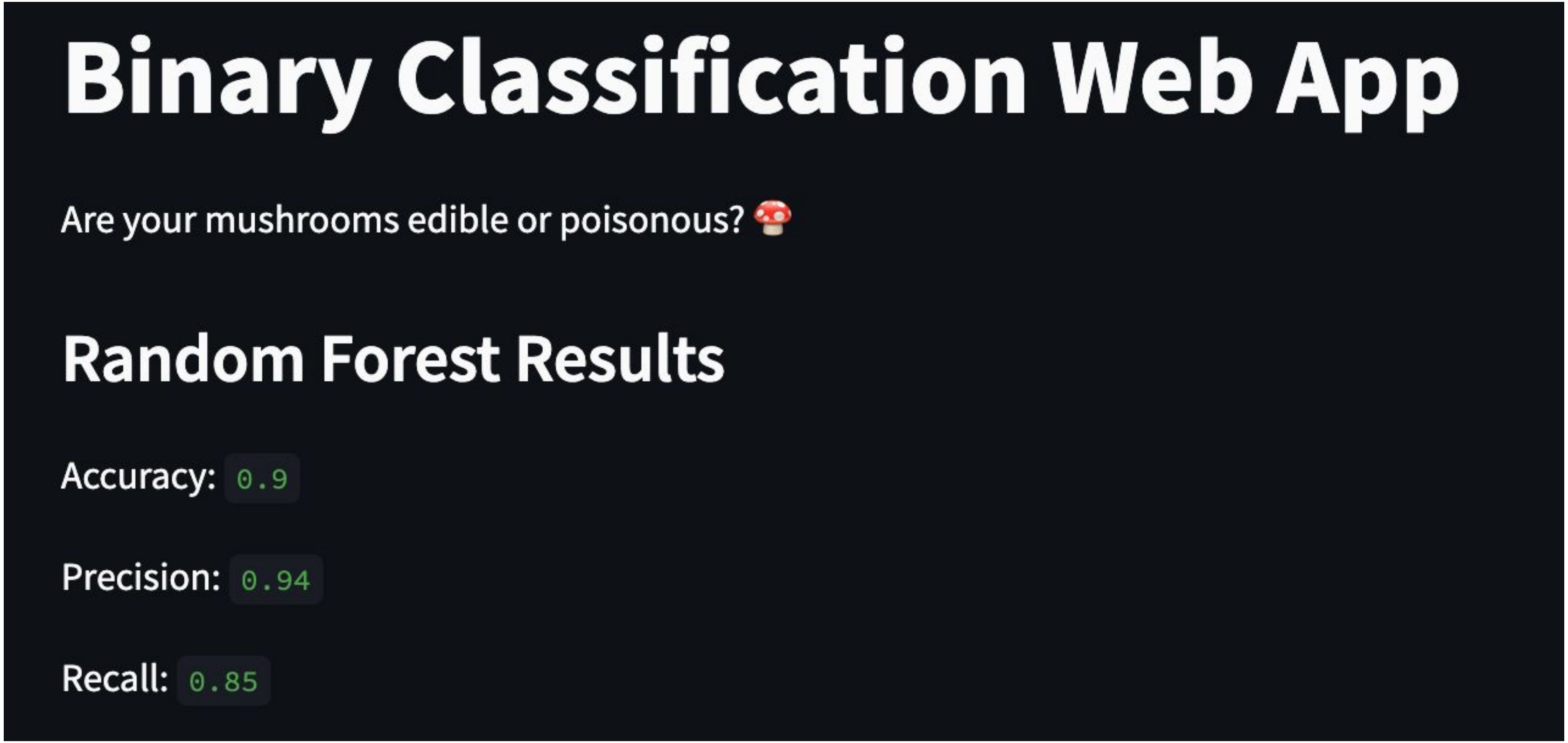
Confusion Matrix ✕

ROC Curve ✕

Precision-Recall Cu… ✕

☑ Show raw data

Result screens:



Binary Classification Web App

Are your mushrooms edible or poisonous? 🍄

Random Forest Results

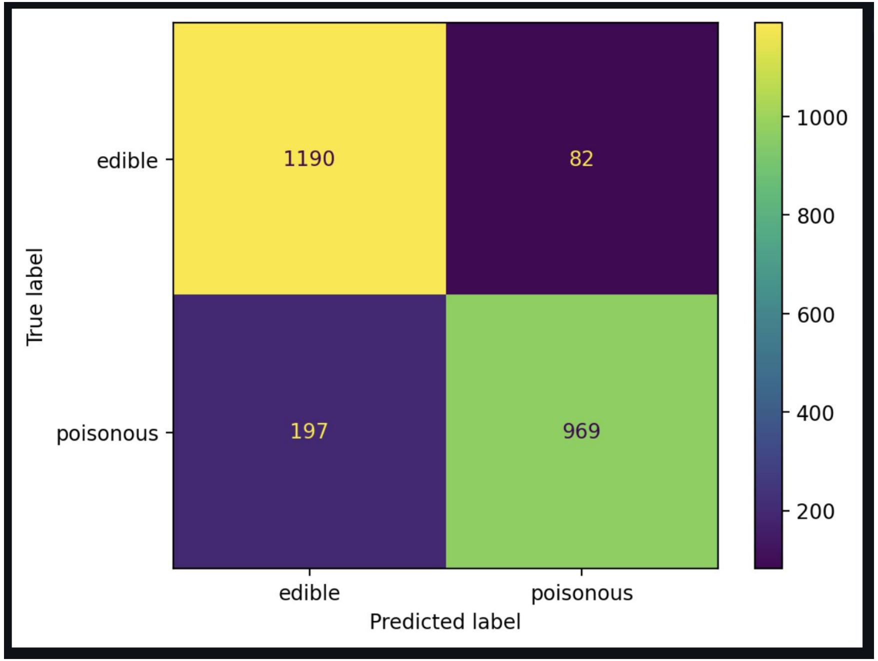
Accuracy: `0.9`

Precision: `0.94`

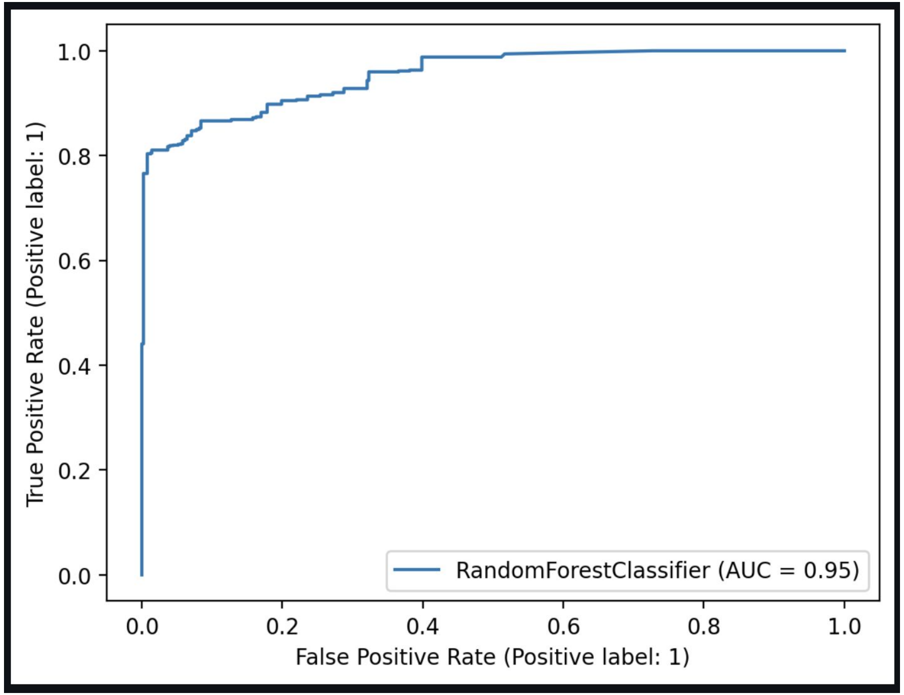
Recall: `0.85`

- Download Model

Confusion Matrix:

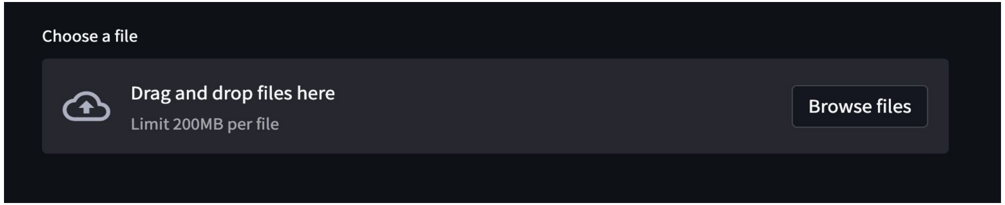


ROC Curve:



Exploratory Data Analysis (EDA) Report:

Upload Dataset:

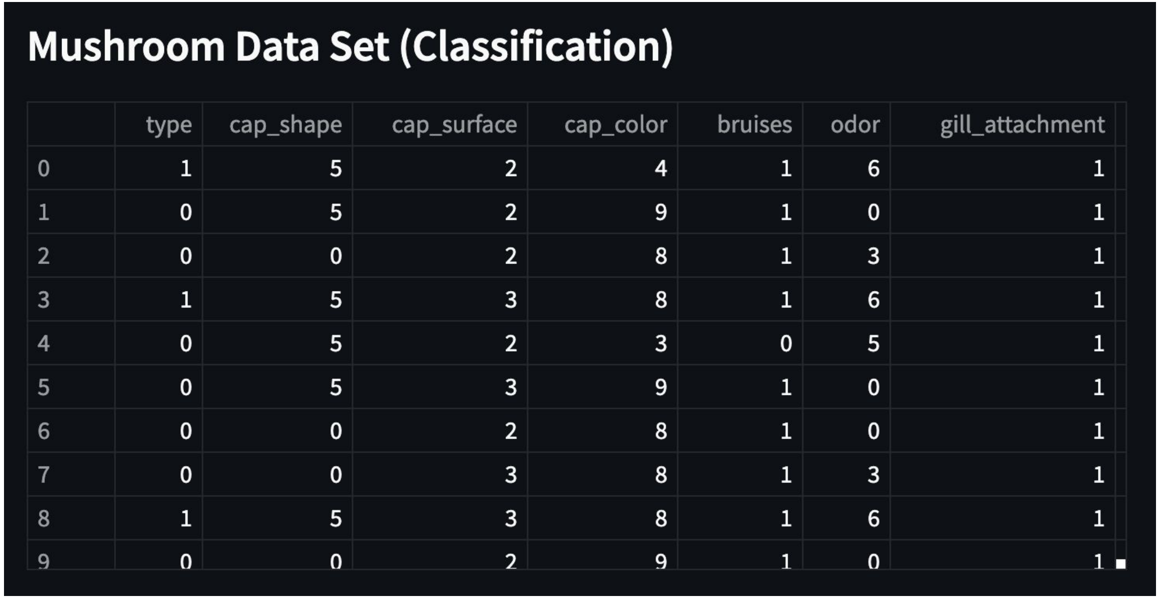


Choose a file

Drag and drop files here

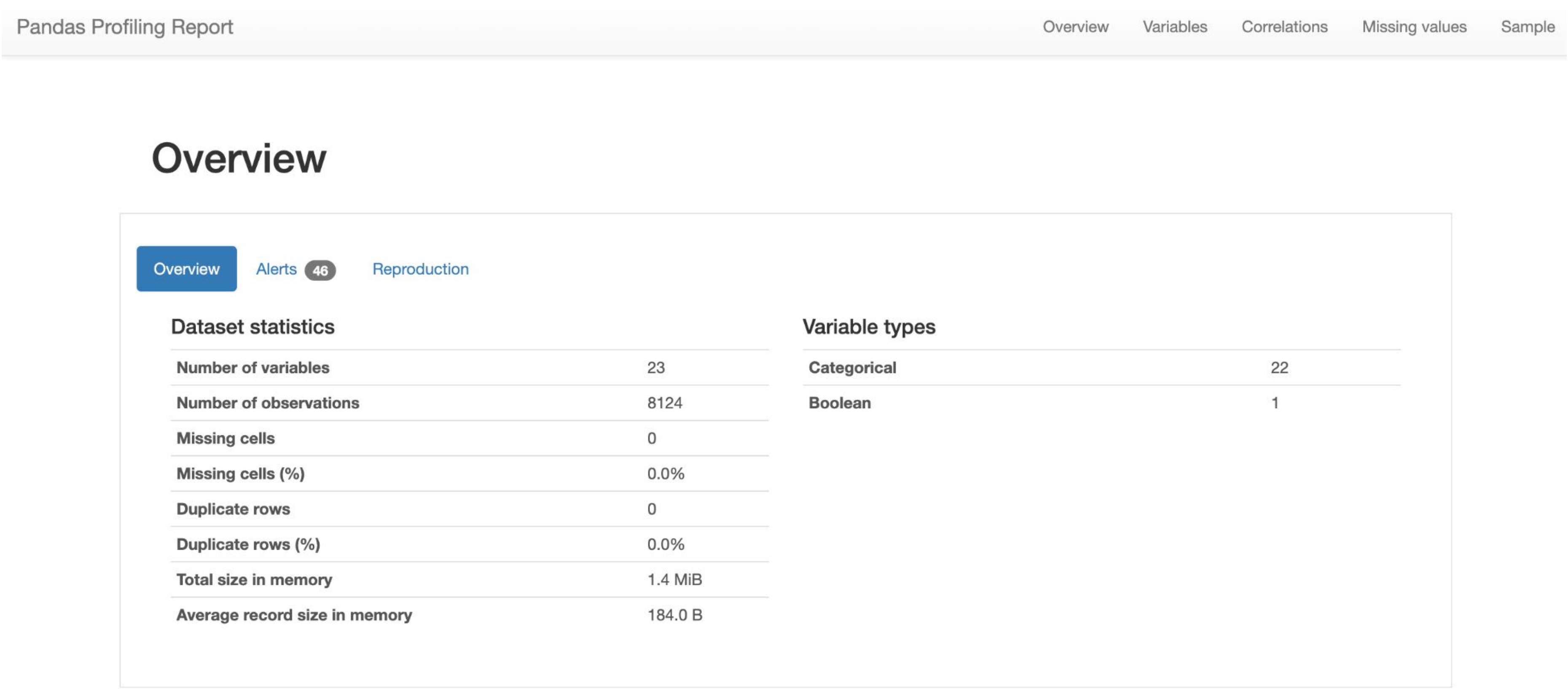
Limit 200MB per file

Show Dataset:



Mushroom Data Set (Classification)

	type	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment
0	1	5	2	4	1	6	1
1	0	5	2	9	1	0	1
2	0	0	2	8	1	3	1
3	1	5	3	8	1	6	1
4	0	5	2	3	0	5	1
5	0	5	3	9	1	0	1
6	0	0	2	8	1	0	1
7	0	0	3	8	1	3	1
8	1	5	3	8	1	6	1
9	0	0	2	9	1	0	1



Overview

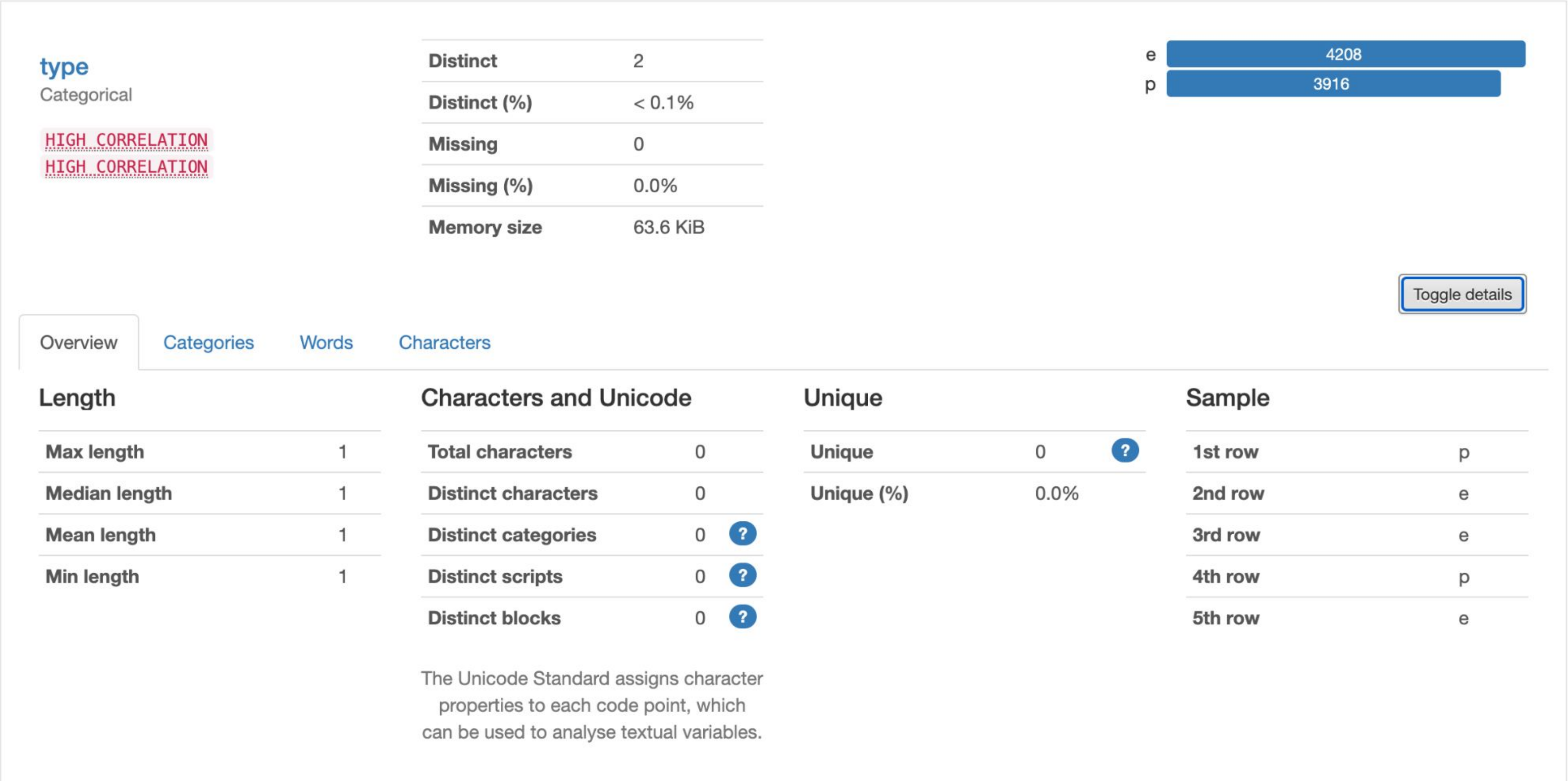
Dataset statistics

Number of variables	23
Number of observations	8124
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	1.4 MiB
Average record size in memory	184.0 B

Variable types

Categorical	22
Boolean	1

Variables



type

Categorical

HIGH CORRELATION
HIGH CORRELATION

Distinct	2
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	63.6 KiB

e	4208
p	3916

Length

Max length	1
Median length	1
Mean length	1
Min length	1

Characters and Unicode

Total characters	0
Distinct characters	0
Distinct categories	0 ?
Distinct scripts	0 ?
Distinct blocks	0 ?

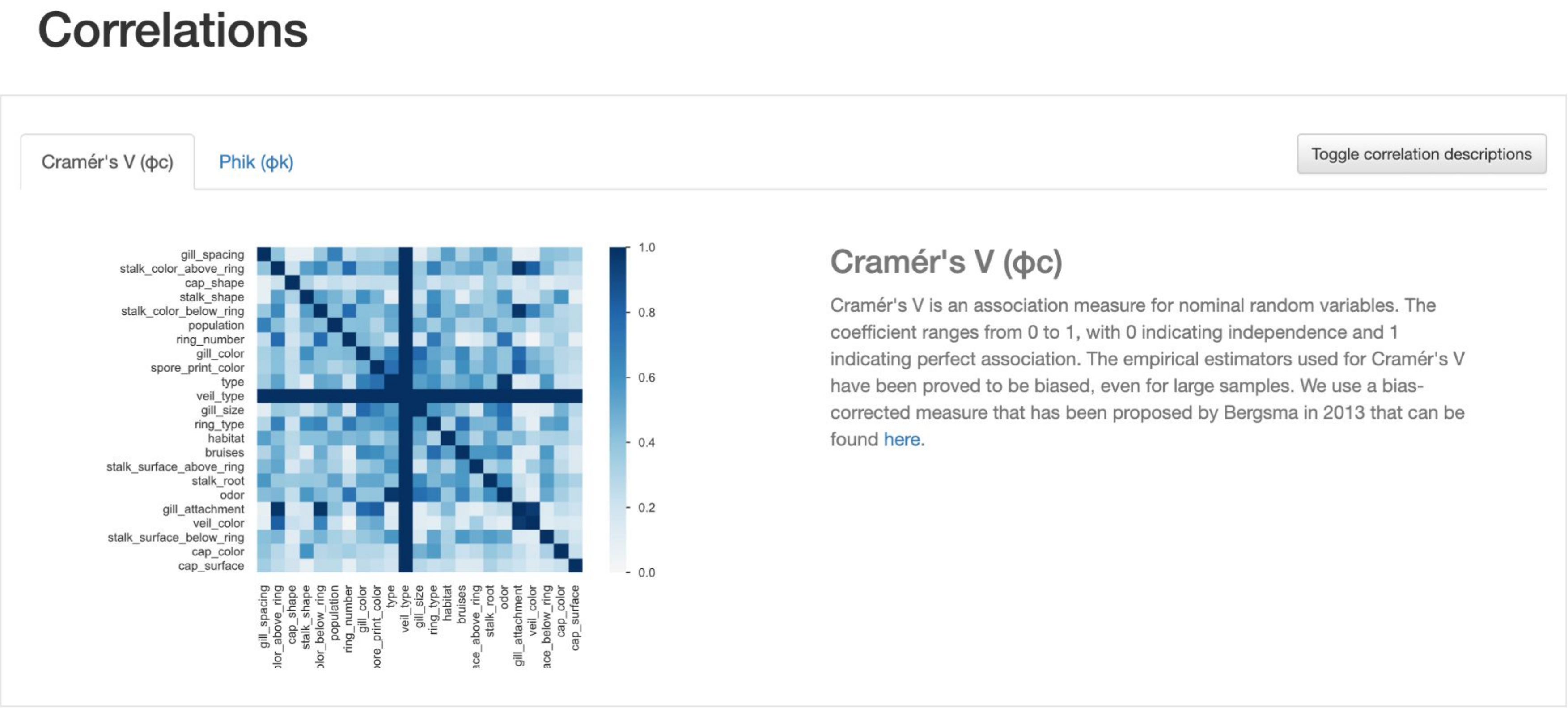
The Unicode Standard assigns character properties to each code point, which can be used to analyse textual variables.

Unique

Unique	0 ?
Unique (%)	0.0%

Sample

1st row	p
2nd row	e
3rd row	e
4th row	p
5th row	e

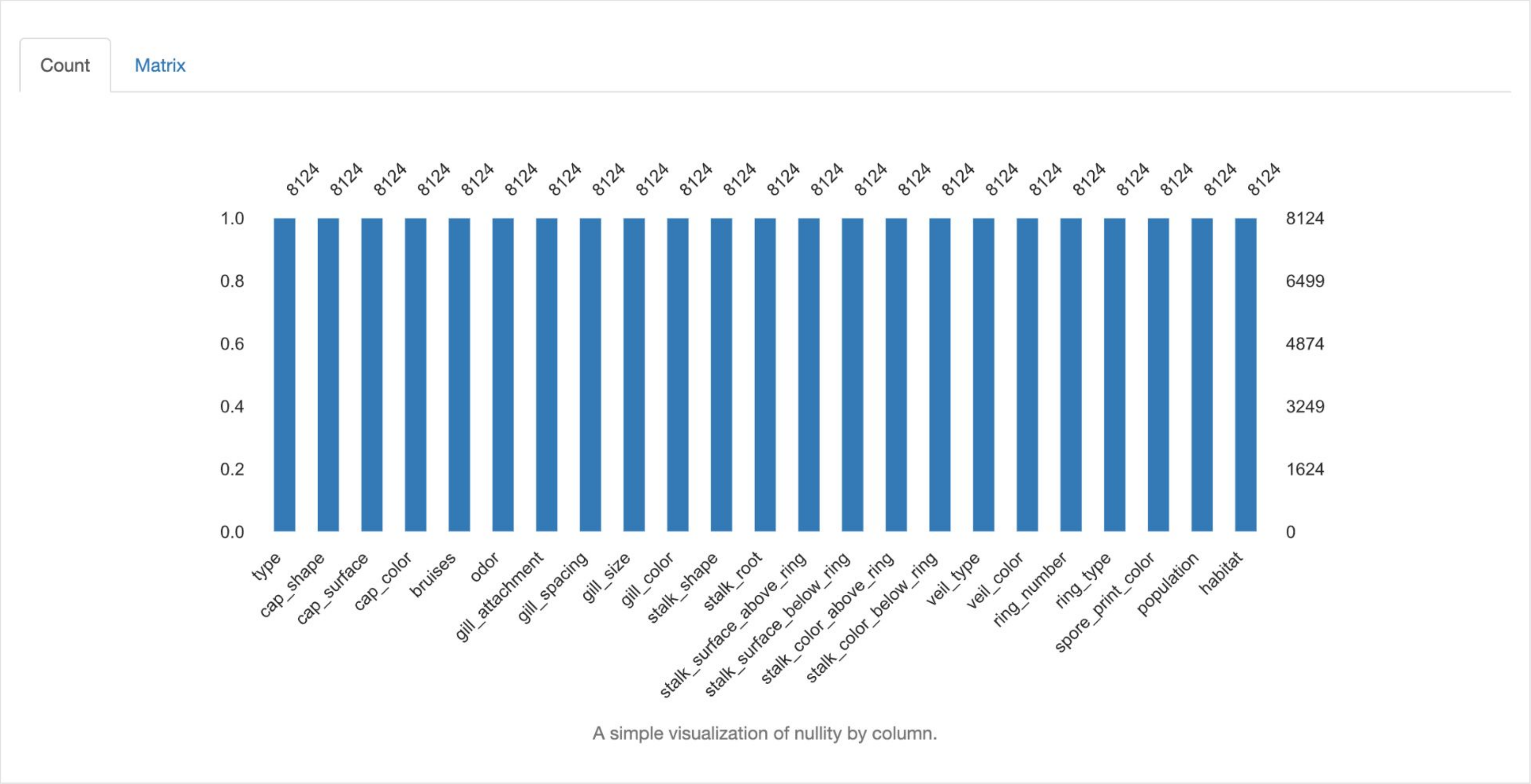


Correlations

Cramér's V (ϕc)

Cramér's V is an association measure for nominal random variables. The coefficient ranges from 0 to 1, with 0 indicating independence and 1 indicating perfect association. The empirical estimators used for Cramér's V have been proved to be biased, even for large samples. We use a bias-corrected measure that has been proposed by Bergsma in 2013 that can be found here.

Missing values



A simple visualization of nullity by column.

Conclusion:

Veasualizer is a data visualization and ML tool and Using Veasualizer a user with zero knowledge can make beautiful visualizations: Graphs and plots. Not just that, Veasualizer has features using which users can Train, Test and Draw predictions from data by just uploading their dataset to Veasualizer.

The whole idea of this project is based on its user base, that is people who are going to use it. Our main objective of Veasualize is to make stuff, which is complex, look easy and useful to everybody. Just by some clicks and uploading datasets, Users will be able to plot different plots like: Scatter plot, Bar graphs, Bubble plots, Box plots, Dot plots, Pie charts and more and ML algorithms include Regression and Classification algorithms like: Linear, Polynomial, Logistic etc

The Exploratory Data Analysis report (EDA) is what makes this project unique, EDA includes Overview of the dataset uploaded, Correlation between variables, Missing values in the dataset and Statistical description of each variable.

Thank you