Final Project

January 30, 2019

1 Final Project for "Python for Data Science" course

1.1 Getting started

To get started, you need to:

- Download the dataset from: https://www.kaggle.com/uciml/mushroom-classification
- Extract the zip file called "mushroom-classification.zip"

This dataset includes descriptions of different species of gilled mushrooms, each one identified as definitely edible or as definitely poisonous.

1.2 Data analysis

In [1]:
```
# Load dependencies.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

# Load data set.
mushrooms = pd.read_csv('./mushrooms.csv', sep=',')
mushrooms.head(10)
```

Out[1]:

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	\
0	p	x	s	n	t	p	f	
1	e	x	s	y	t	a	f	
2	e	b	s	w	t	l	f	
3	p	x	y	w	t	p	f	
4	e	x	s	g	f	n	f	
5	e	x	y	y	t	a	f	
6	e	b	s	w	t	a	f	
7	e	b	y	w	t	l	f	
8	p	x	y	w	t	p	f	

```
9     e         b           s         y       t    a                 f

  gill-spacing gill-size gill-color   ...   stalk-surface-below-ring  \
0            c         n          k   ...                          s
1            c         b          k   ...                          s
2            c         b          n   ...                          s
3            c         n          n   ...                          s
4            w         b          k   ...                          s
5            c         b          n   ...                          s
6            c         b          g   ...                          s
7            c         b          n   ...                          s
8            c         n          p   ...                          s
9            c         b          g   ...                          s

  stalk-color-above-ring stalk-color-below-ring veil-type veil-color  \
0                      w                      w         p          w
1                      w                      w         p          w
2                      w                      w         p          w
3                      w                      w         p          w
4                      w                      w         p          w
5                      w                      w         p          w
6                      w                      w         p          w
7                      w                      w         p          w
8                      w                      w         p          w
9                      w                      w         p          w

  ring-number ring-type spore-print-color population habitat
0           o         p                 k          s       u
1           o         p                 n          n       g
2           o         p                 n          n       m
3           o         p                 k          s       u
4           o         e                 n          a       g
5           o         p                 k          n       g
6           o         p                 k          n       m
7           o         p                 n          s       m
8           o         p                 k          v       g
9           o         p                 k          s       m

[10 rows x 23 columns]
```

1.2.1 Create a classification model

```
In [2]: # Separate output from input columns.
        X = mushrooms.drop('class', axis=1)
        y = mushrooms['class'].copy()

        # Convert categorical values into indicator variables.
        X = pd.get_dummies(X)
```

```
        # Separate data into training and testing subsets.
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=
        # Create and train a classifier using a decision tree.
        mushroom_tree_classifier = DecisionTreeClassifier(max_leaf_nodes=7, random_state=0)
        mushroom_tree_classifier.fit(X_train, y_train)
        # Make predictions.
        y_pred_dt = mushroom_tree_classifier.predict(X_test)
        y_pred_dt[:10]
```

```
Out[2]: array(['p', 'e', 'e', 'p', 'e', 'e', 'e', 'e', 'e', 'e'], dtype=object)
```

```
In [3]: # Check the accuracy of the decision tree predictions.
        accuracy_score(y_true = y_test, y_pred = y_pred_dt)
```

```
Out[3]: 0.9914211115255501
```

```
In [4]: # Create and train a classifier using a random forest.
        mushroom_forest_classifier = RandomForestClassifier(n_estimators=4, max_leaf_nodes=7, r
        mushroom_forest_classifier.fit(X_train, y_train)
        # Make predictions (again).
        y_pred_rf = mushroom_forest_classifier.predict(X_test)
        y_pred_rf[:10]
```

```
Out[4]: array(['p', 'e', 'e', 'p', 'e', 'e', 'e', 'e', 'e', 'e'], dtype=object)
```

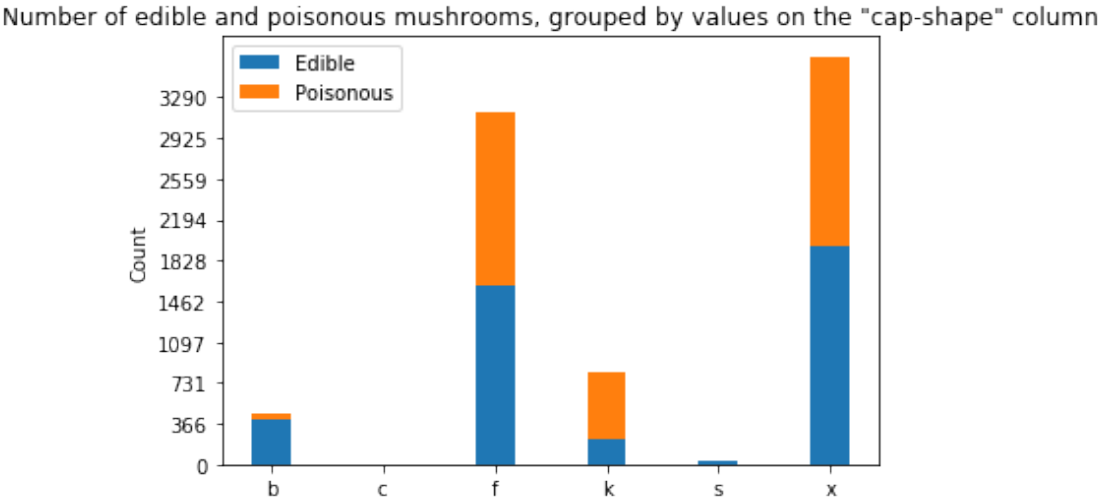
```
In [5]: # Check the accuracy of the random forest predictions.
        accuracy_score(y_true = y_test, y_pred = y_pred_rf)
```

```
Out[5]: 0.9593435285341291
```

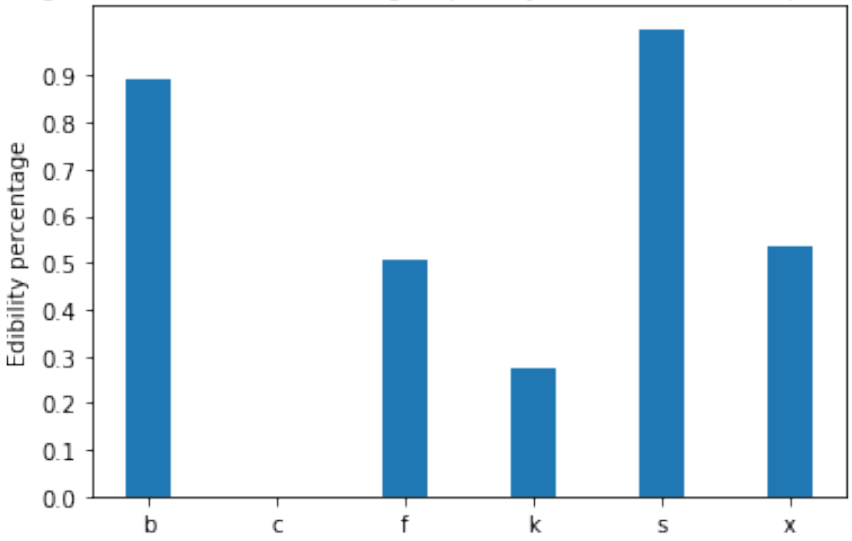
1.2.2 Analyse features individually

```
In [6]: # Iterate each column.
        columns = mushrooms.columns.values
        edibles = mushrooms[mushrooms['class'] == 'e']
        poisonous = mushrooms[mushrooms['class'] == 'p']
        for column in columns:
            if column != 'class':
                # Get counts for each value in the column.
                edibleCount = edibles[column].value_counts()
                poisonousCount = poisonous[column].value_counts()
                # Create dataframe and set missing values.
                data = pd.concat([edibleCount, poisonousCount], axis=1, sort=True)
                data.columns = ['edible', 'poisonous']
```

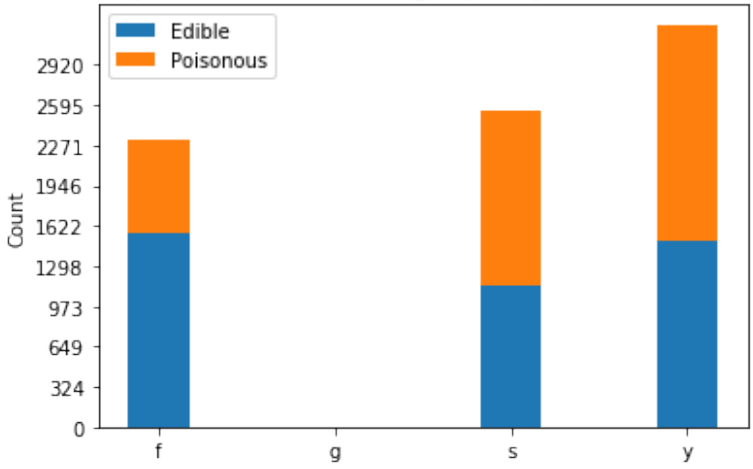
```
data = data.fillna(0).astype(int)

# Draw plot bar with counts.
ind = np.arange(len(data.index))
values1 = data['edible'].tolist()
values2 = data['poisonous'].tolist()
totals = (data['edible'] + data['poisonous']).tolist()
maxValue = max(totals)

p1 = plt.bar(ind, values1, 0.35)
p2 = plt.bar(ind, values2, 0.35, bottom=data['edible'].tolist())

plt.ylabel('Count')
plt.title('Number of edible and poisonous mushrooms, grouped by values on the '
plt.xticks(ind, data.index)
plt.yticks(np.arange(0, maxValue, maxValue/10))
plt.legend((p1[0], p2[0]), ('Edible', 'Poisonous'))
plt.show()

# Draw plot bar with percentages.
percentages = (data['edible']/(data['edible'] + data['poisonous'])).tolist()

p3 = plt.bar(ind, percentages, 0.35)

plt.ylabel('Edibility percentage')
plt.title('Percentage of edible mushrooms, grouped by values on the "' + colum
plt.xticks(ind, data.index)
plt.yticks(np.arange(0, 1.0, 0.1))
plt.show()
```



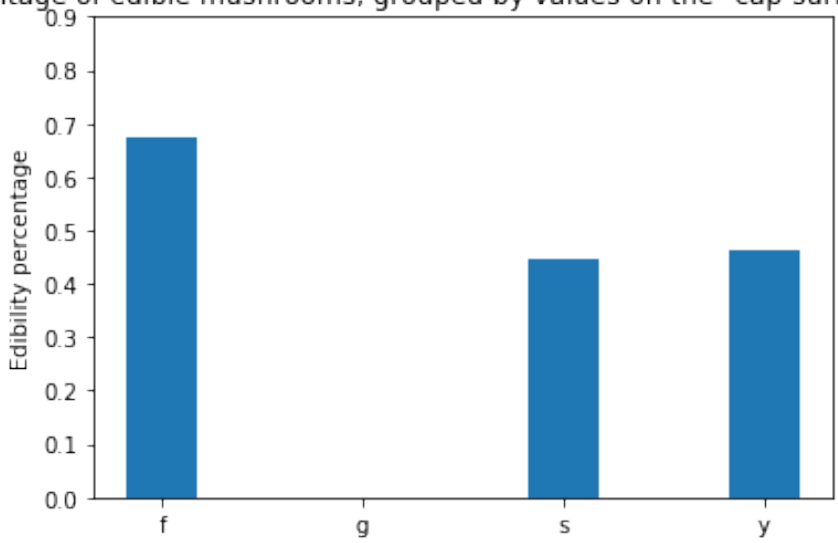
Percentage of edible mushrooms, grouped by values on the "cap-shape" column



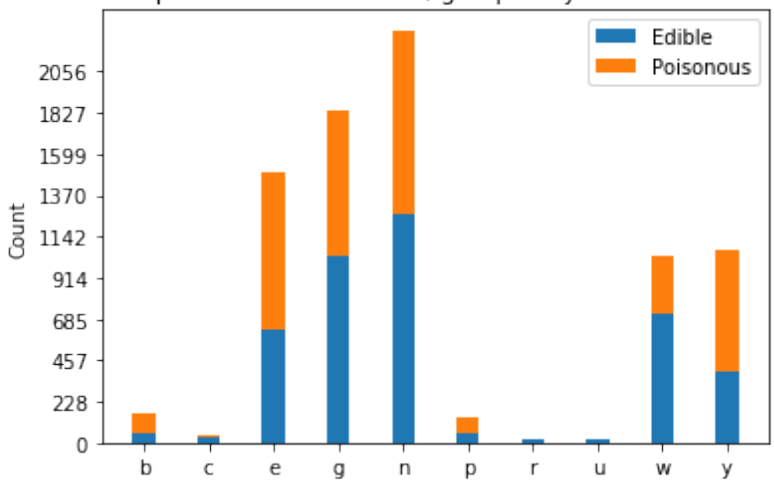
Number of edible and poisonous mushrooms, grouped by values on the "cap-surface" column



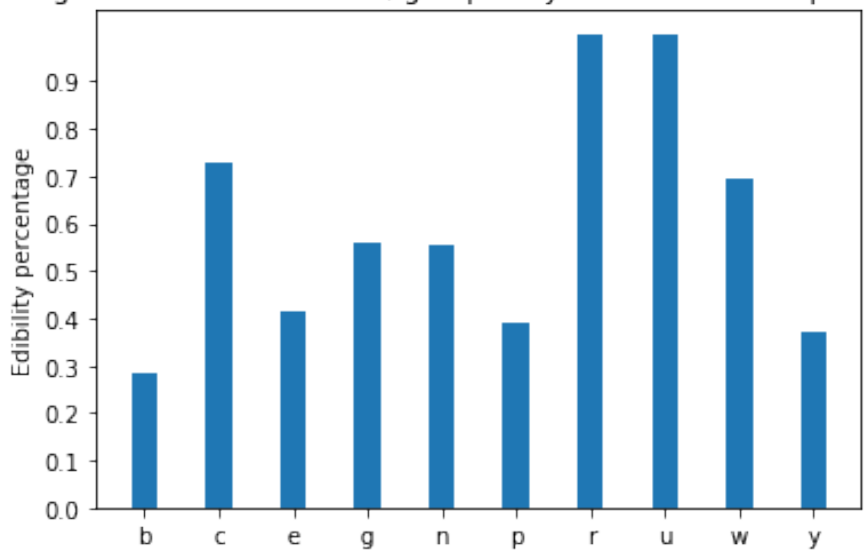
Percentage of edible mushrooms, grouped by values on the "cap-surface" column



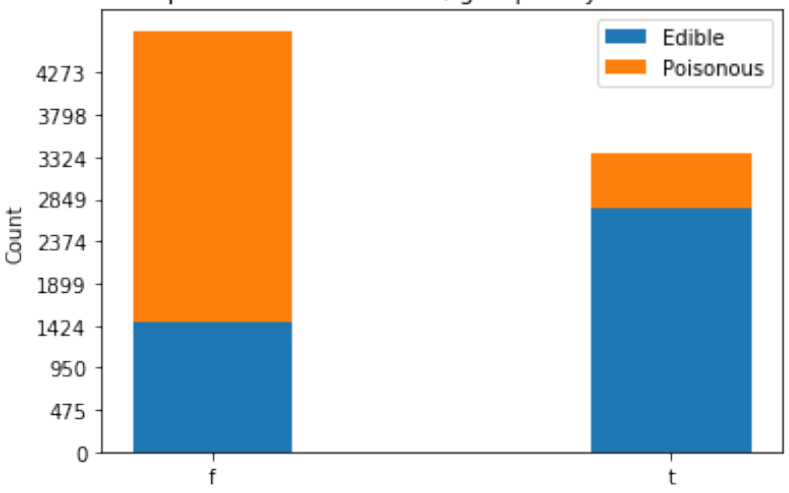
Number of edible and poisonous mushrooms, grouped by values on the "cap-color" column



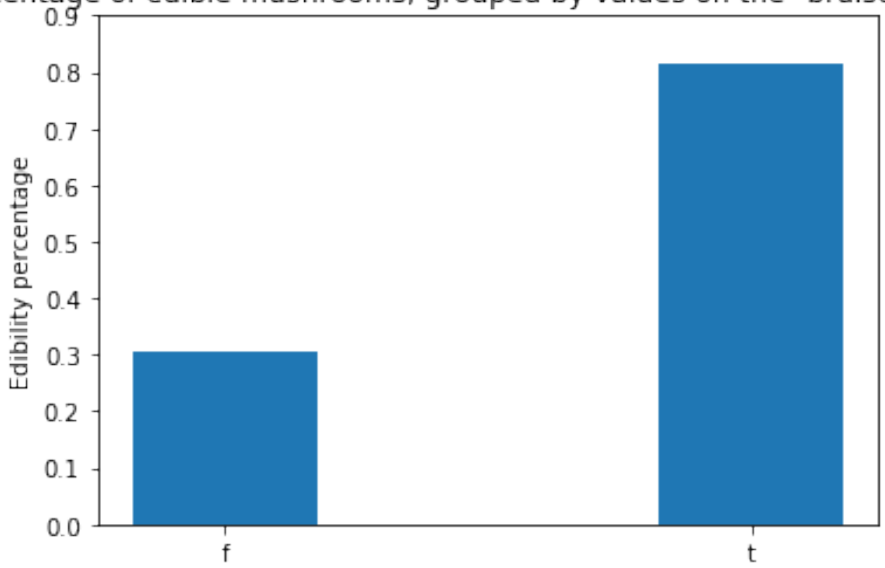
Percentage of edible mushrooms, grouped by values on the "cap-color" column



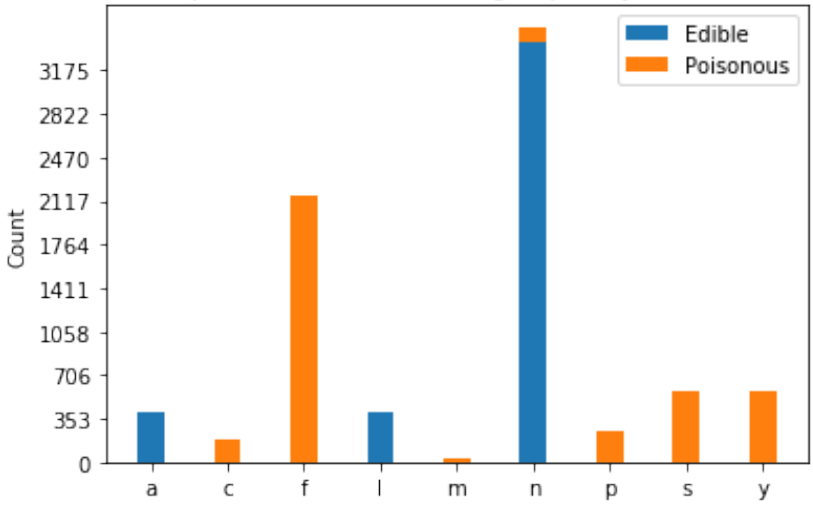
Number of edible and poisonous mushrooms, grouped by values on the "bruises" column

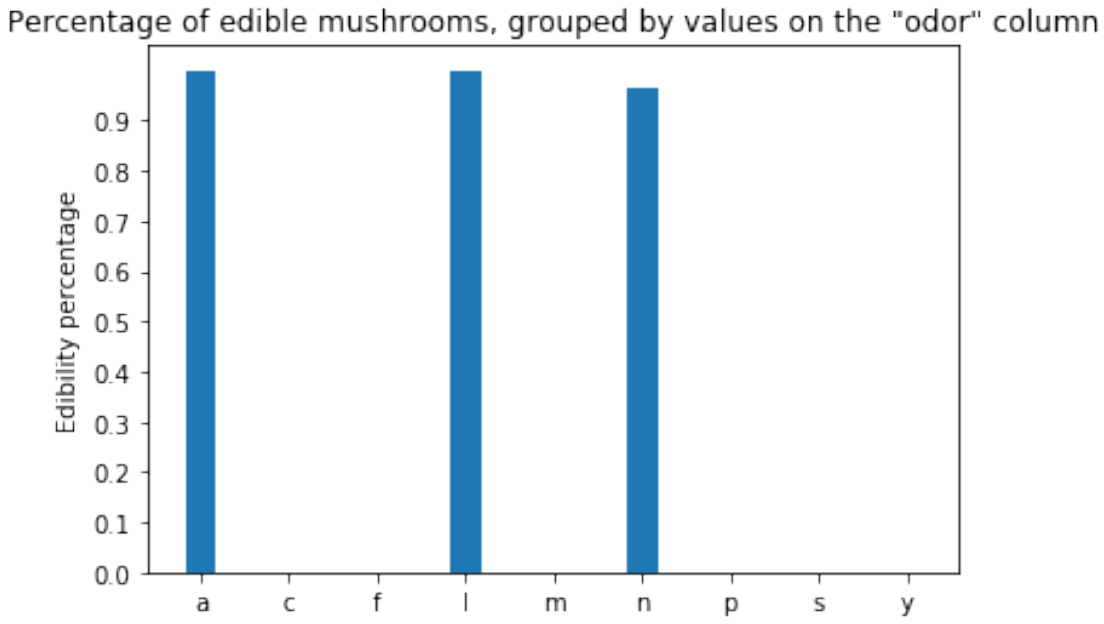


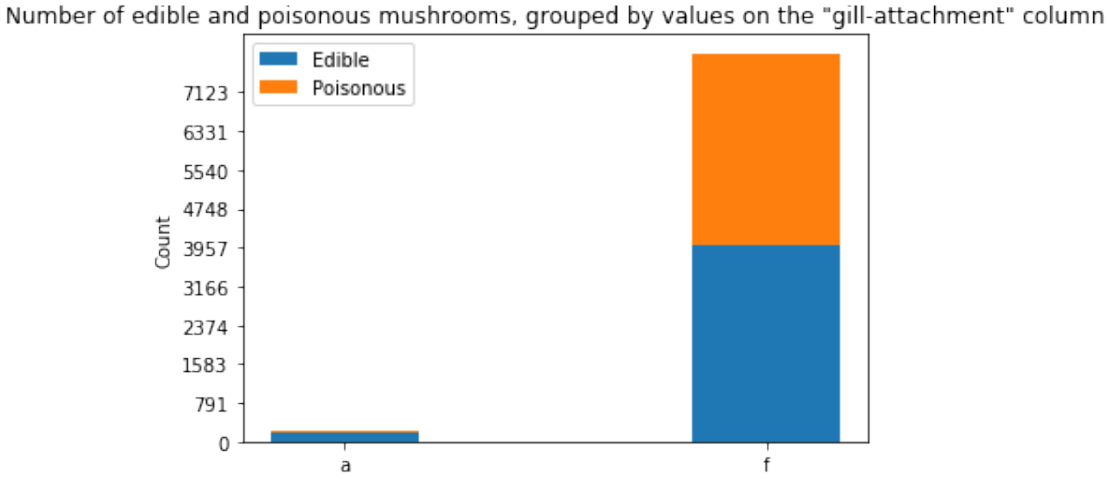
Percentage of edible mushrooms, grouped by values on the "bruises" column



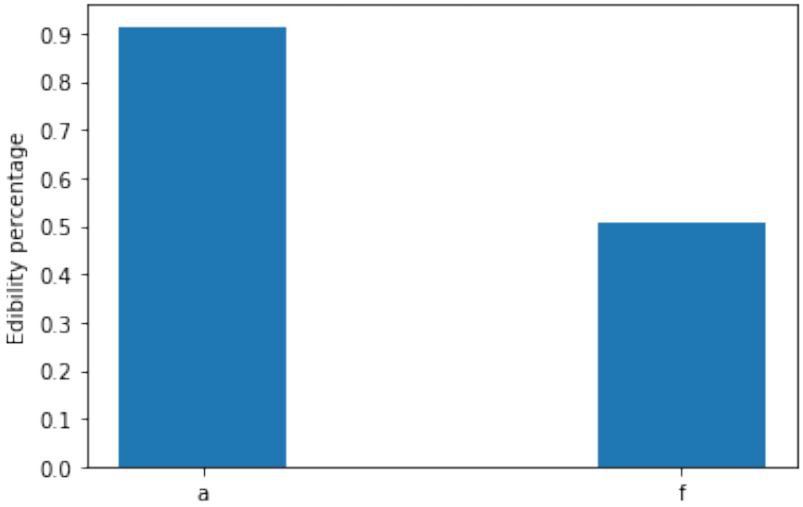
Number of edible and poisonous mushrooms, grouped by values on the "odor" column



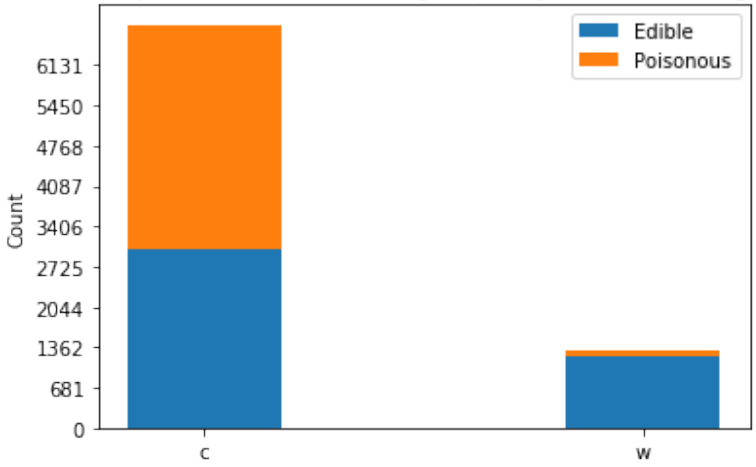

Percentage of edible mushrooms, grouped by values on the "odor" column
Edibility percentage
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
a
c
f
l
m
n
p
s
y


Number of edible and poisonous mushrooms, grouped by values on the "gill-attachment" column
Edible
Poisonous
Count
0
791
1583
2374
3166
3957
4748
5540
6331
7123
a
f

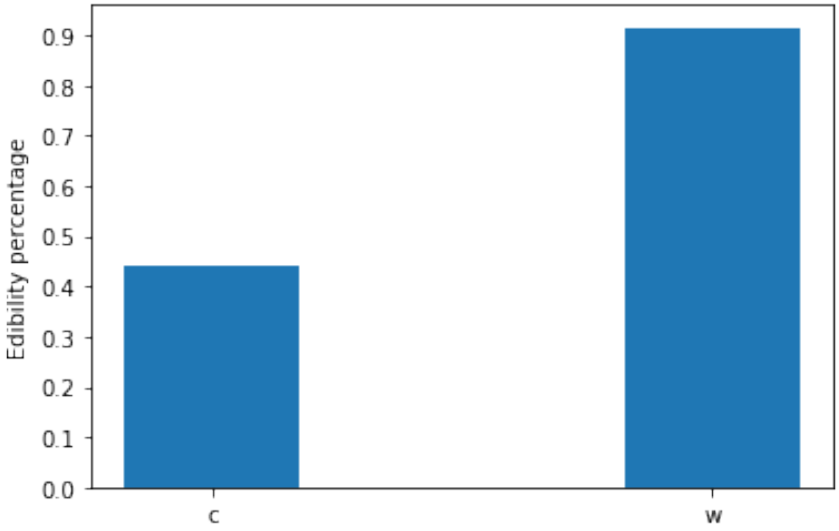
Percentage of edible mushrooms, grouped by values on the "gill-attachment" column



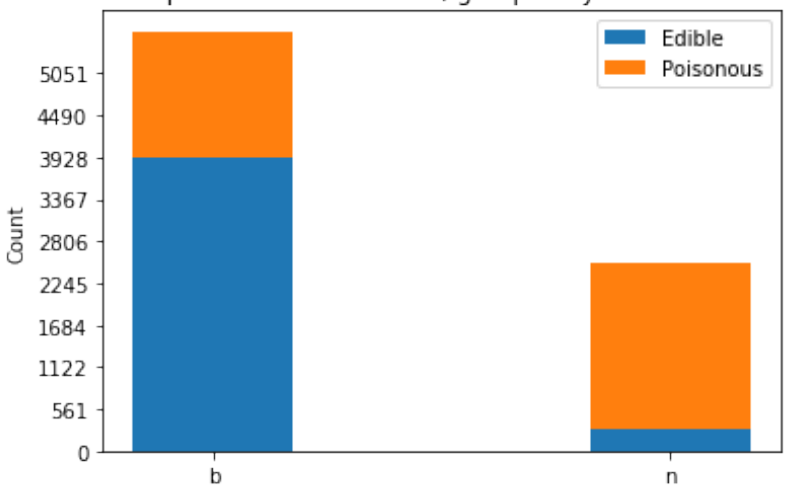
Number of edible and poisonous mushrooms, grouped by values on the "gill-spacing" column



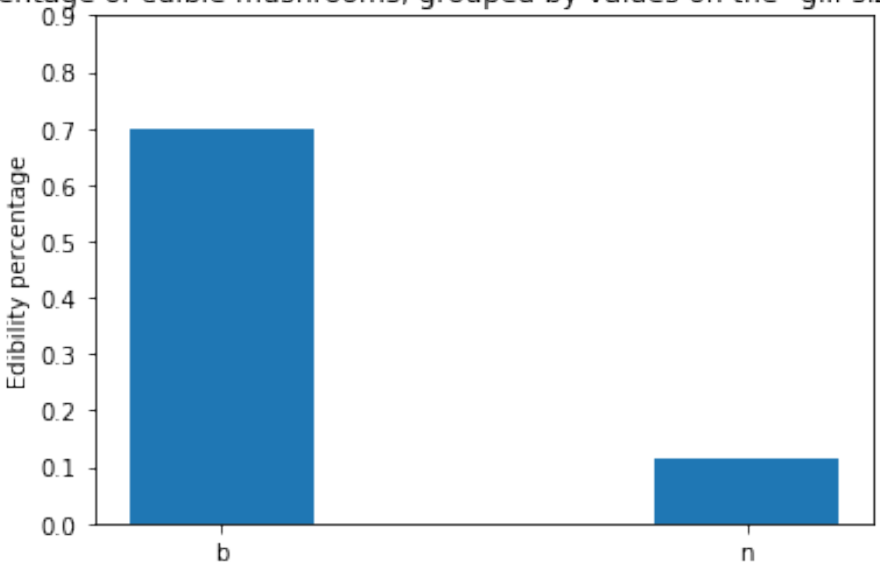
Percentage of edible mushrooms, grouped by values on the "gill-spacing" column



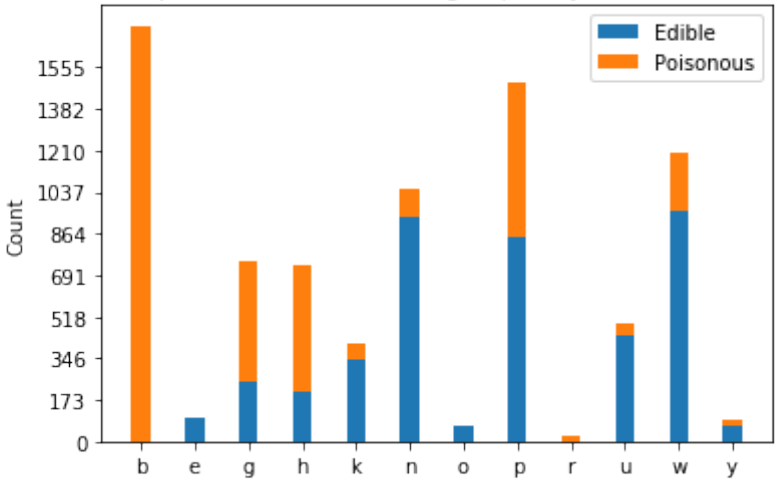
Number of edible and poisonous mushrooms, grouped by values on the "gill-size" column



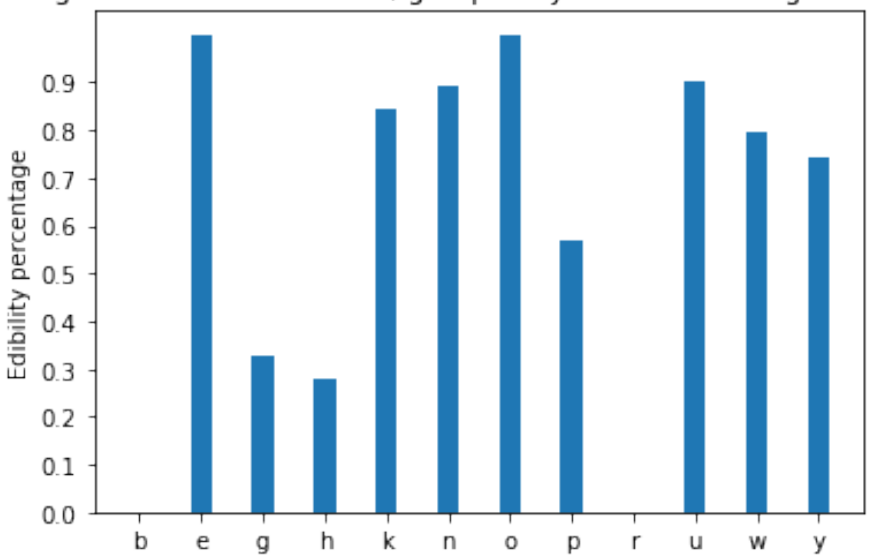
Percentage of edible mushrooms, grouped by values on the "gill-size" column



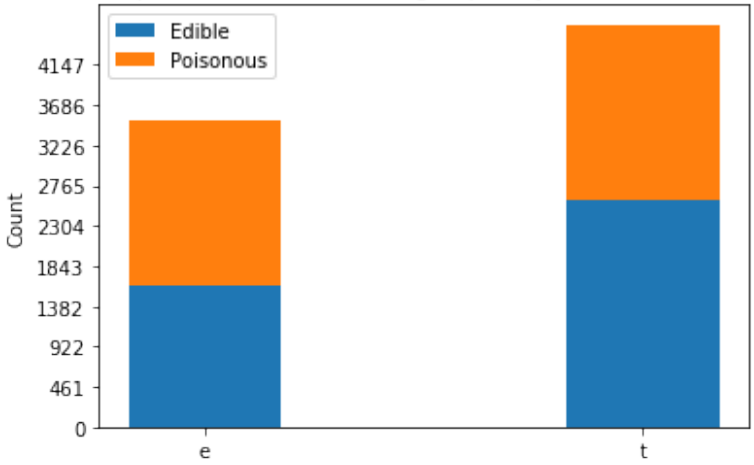
Number of edible and poisonous mushrooms, grouped by values on the "gill-color" column



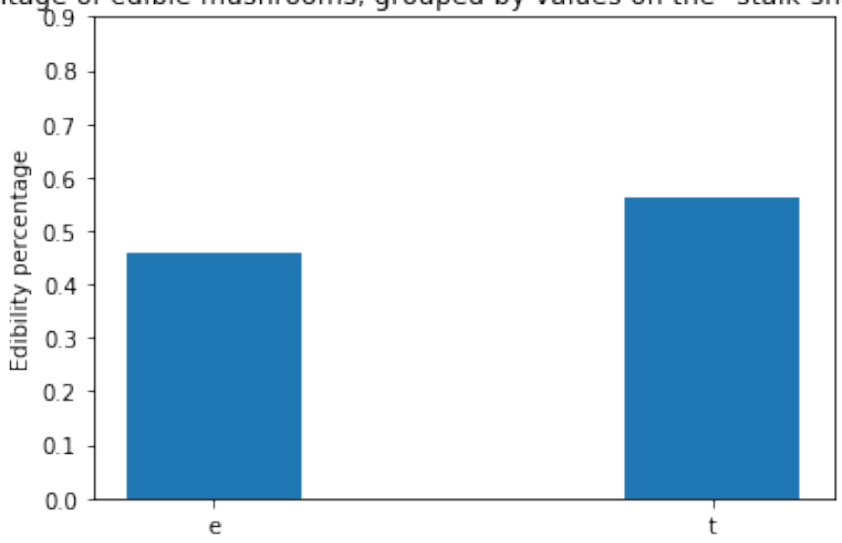
Percentage of edible mushrooms, grouped by values on the "gill-color" column



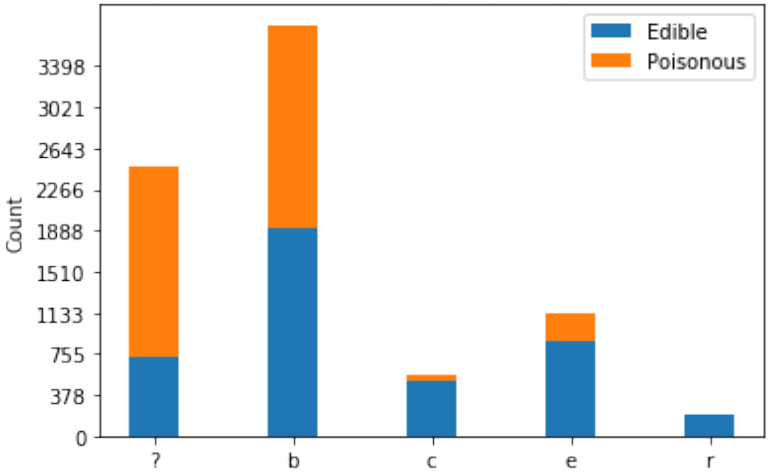
Number of edible and poisonous mushrooms, grouped by values on the "stalk-shape" column



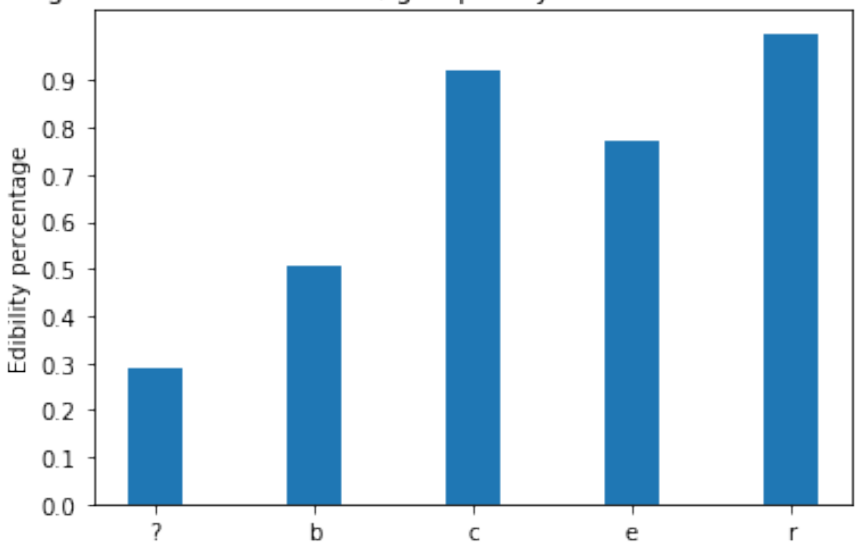
Percentage of edible mushrooms, grouped by values on the "stalk-shape" column



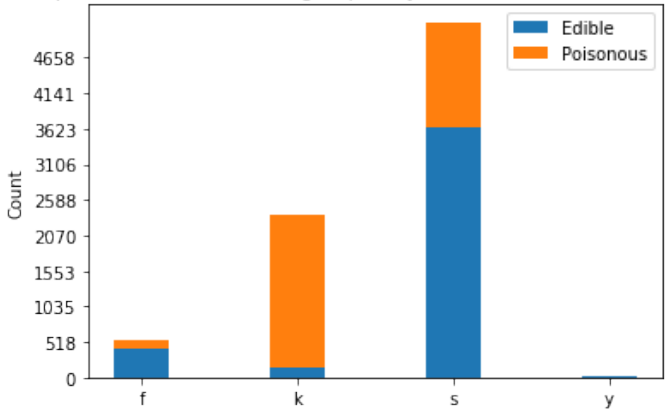
Number of edible and poisonous mushrooms, grouped by values on the "stalk-root" column



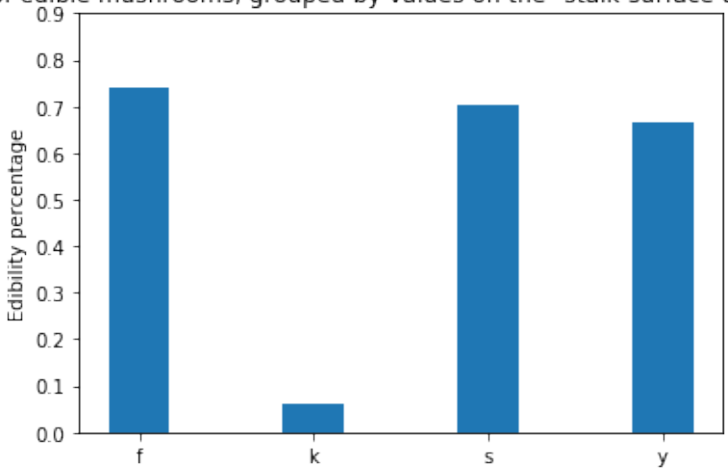
Percentage of edible mushrooms, grouped by values on the "stalk-root" column



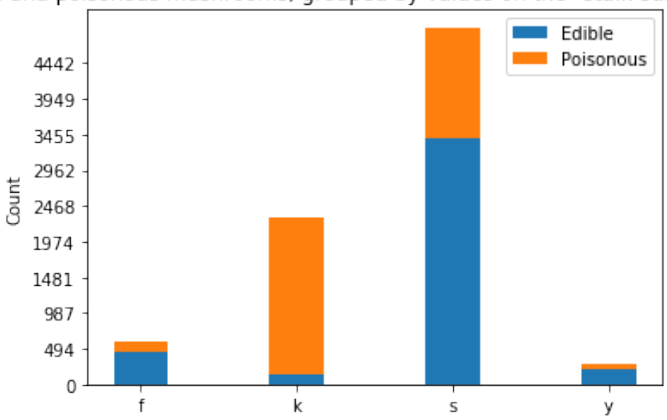
Number of edible and poisonous mushrooms, grouped by values on the "stalk-surface-above-ring" column



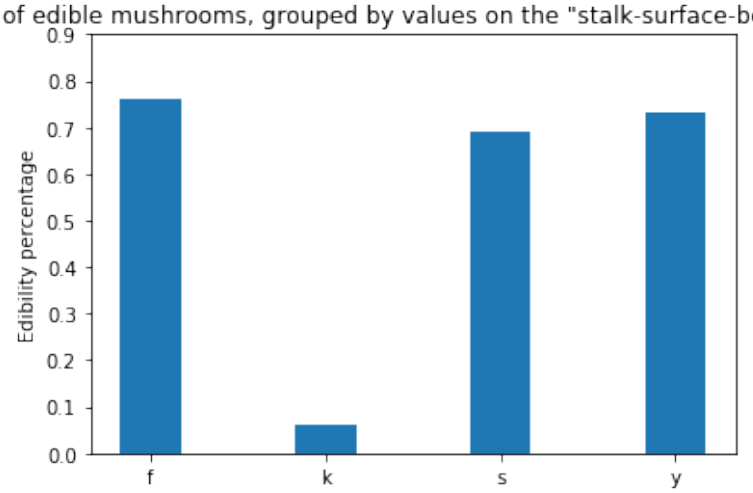
Percentage of edible mushrooms, grouped by values on the "stalk-surface-above-ring" column



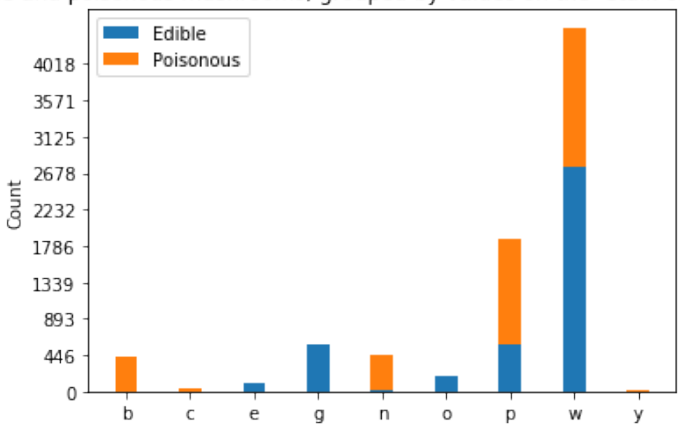
Number of edible and poisonous mushrooms, grouped by values on the "stalk-surface-below-ring" column



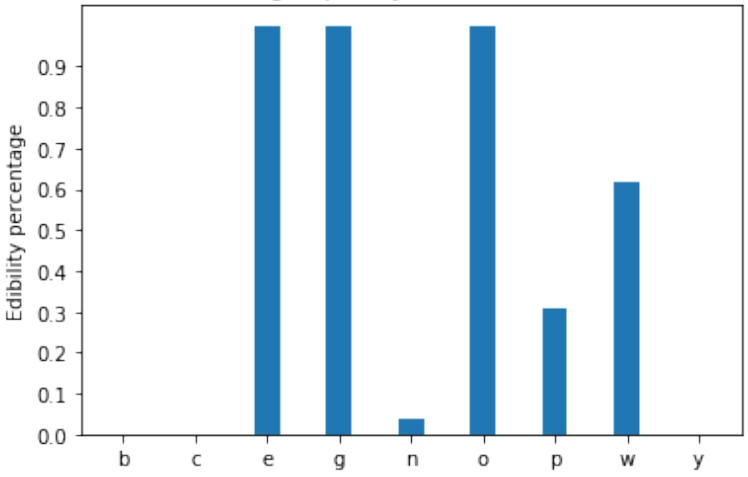
Percentage of edible mushrooms, grouped by values on the "stalk-surface-below-ring" column



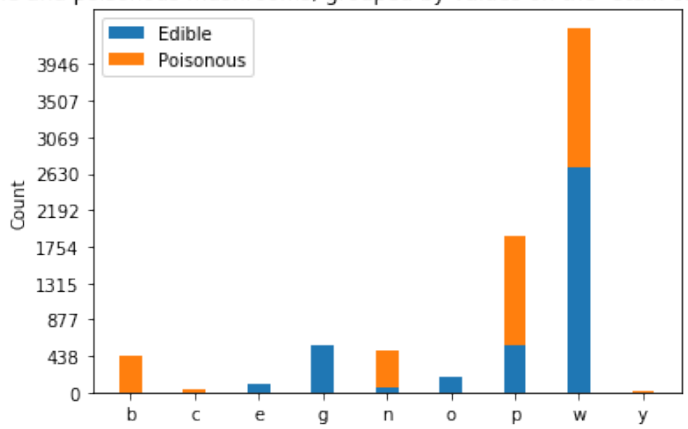
Number of edible and poisonous mushrooms, grouped by values on the "stalk-color-above-ring" column



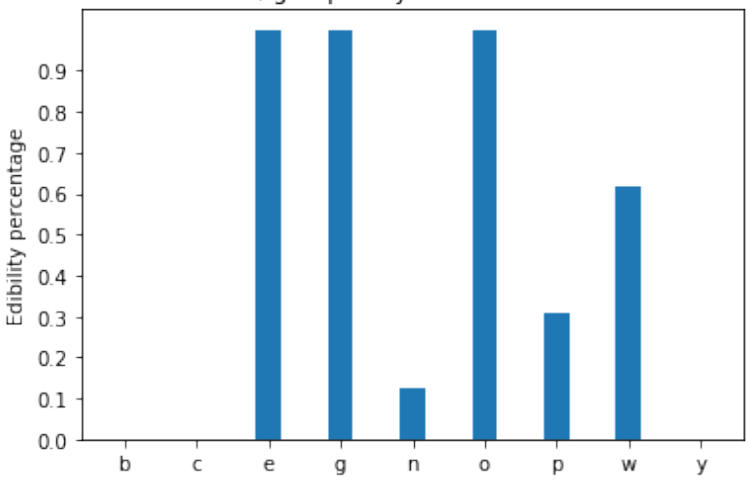
Percentage of edible mushrooms, grouped by values on the "stalk-color-above-ring" column



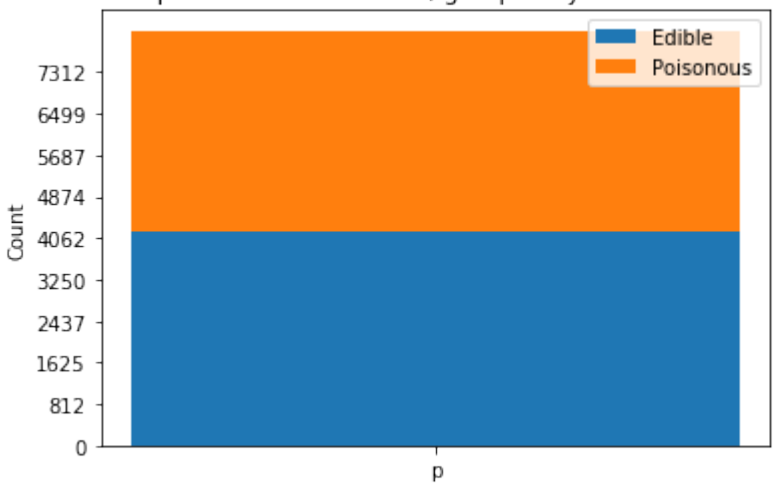
Number of edible and poisonous mushrooms, grouped by values on the "stalk-color-below-ring" column

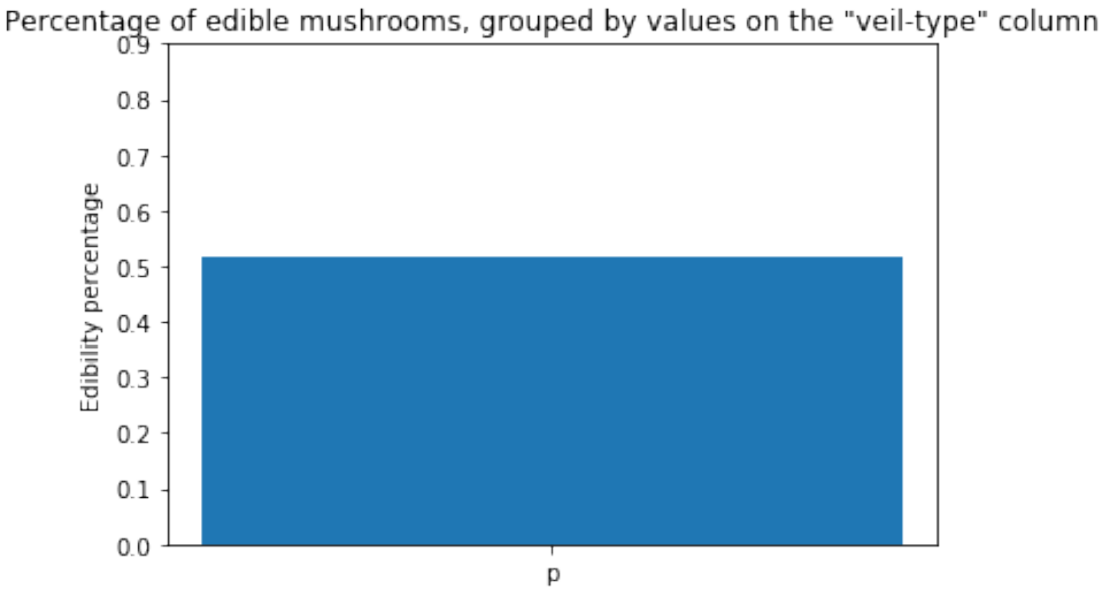


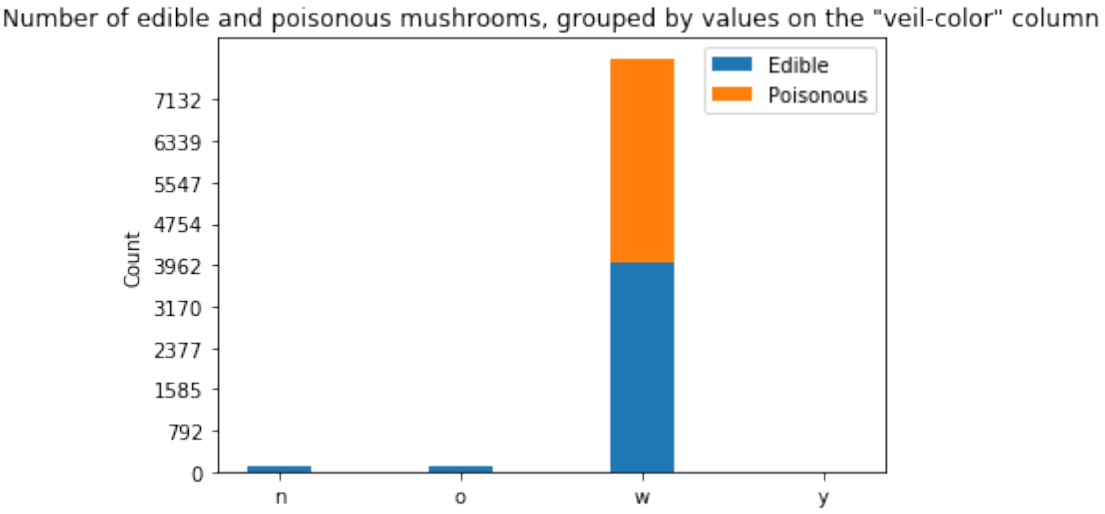
Percentage of edible mushrooms, grouped by values on the "stalk-color-below-ring" column



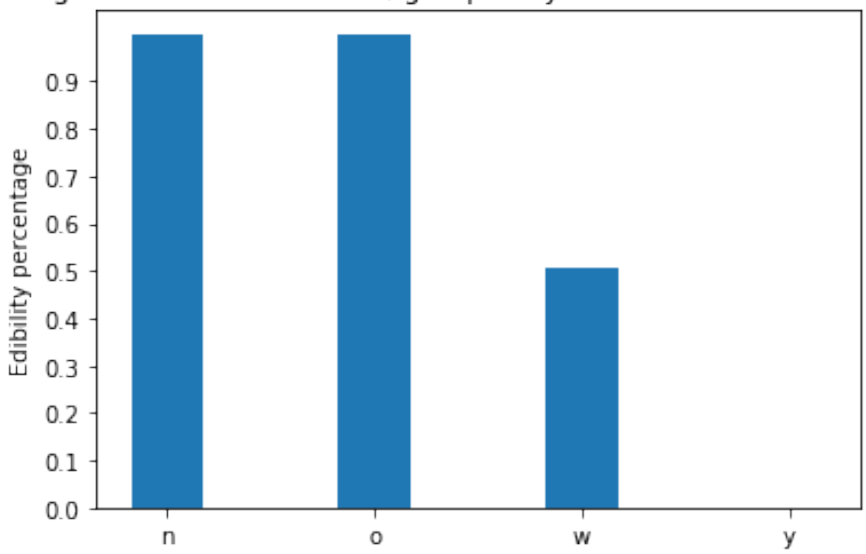
Number of edible and poisonous mushrooms, grouped by values on the "veil-type" column



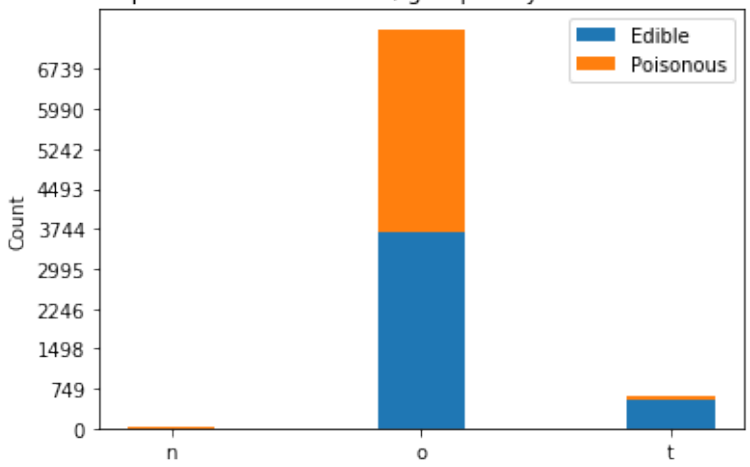

Percentage of edible mushrooms, grouped by values on the "veil-type" column
Edibility percentage
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
p


Number of edible and poisonous mushrooms, grouped by values on the "veil-color" column
Count
0
792
1585
2377
3170
3962
4754
5547
6339
7132
Edible
Poisonous
n
o
w
y

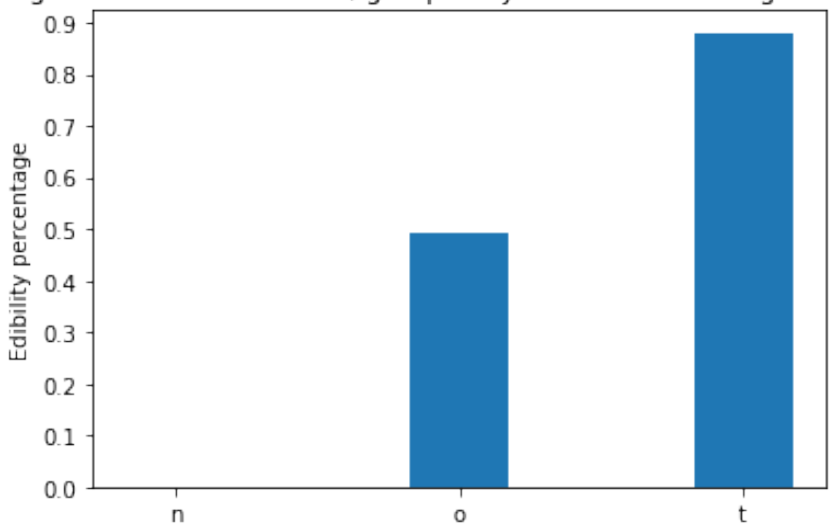
Percentage of edible mushrooms, grouped by values on the "veil-color" column



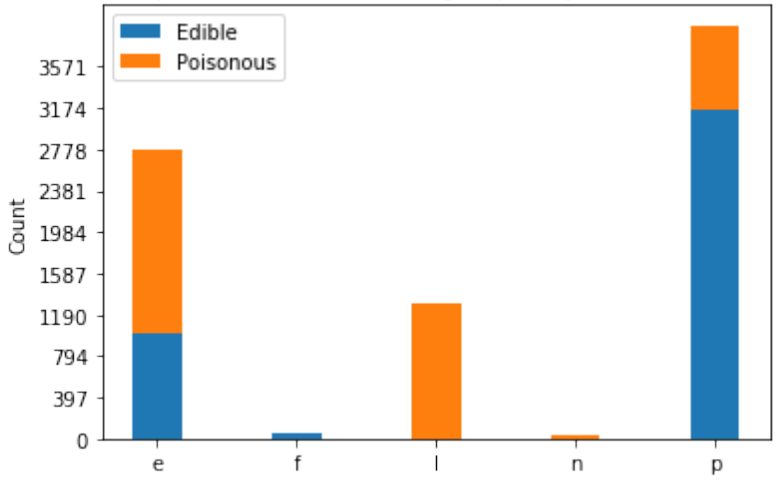
Number of edible and poisonous mushrooms, grouped by values on the "ring-number" column



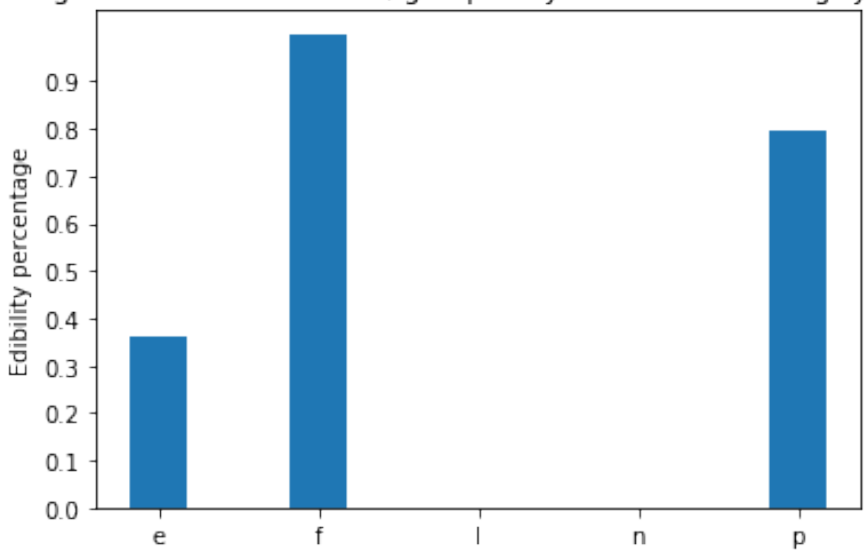
Percentage of edible mushrooms, grouped by values on the "ring-number" column



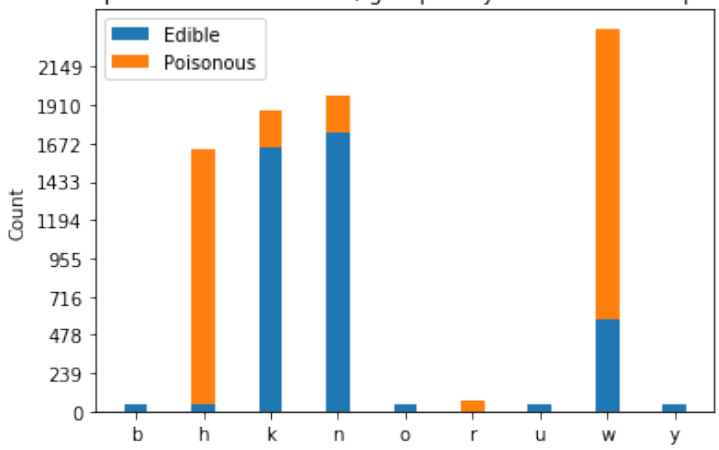
Number of edible and poisonous mushrooms, grouped by values on the "ring-type" column



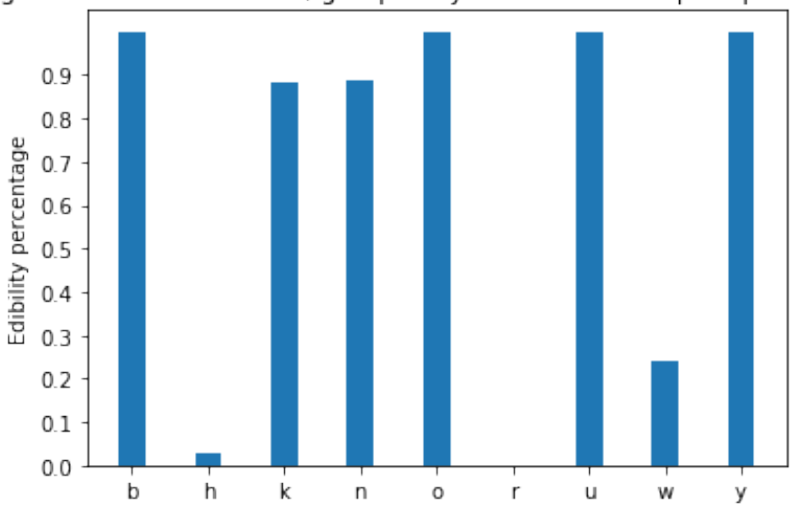
Percentage of edible mushrooms, grouped by values on the "ring-type" column



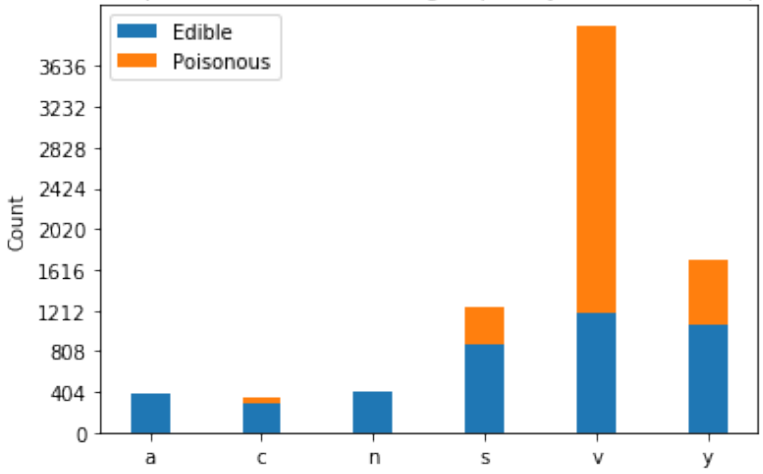
Number of edible and poisonous mushrooms, grouped by values on the "spore-print-color" column



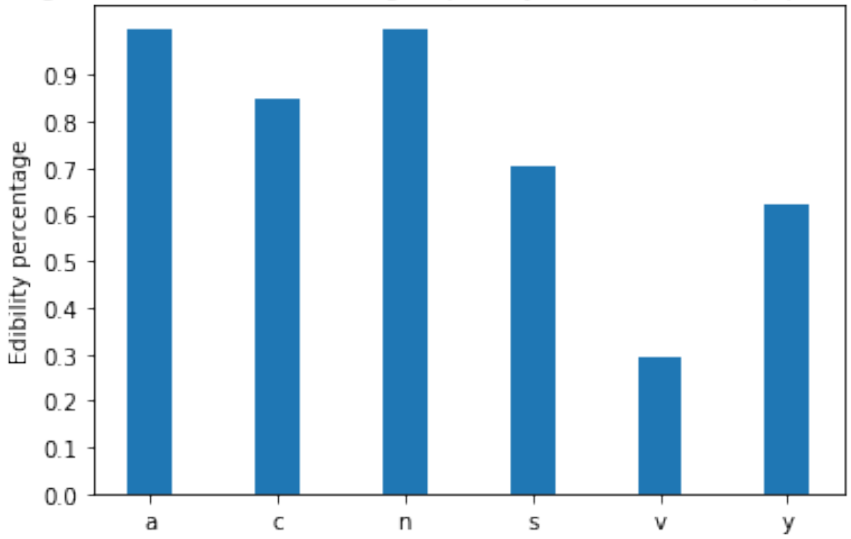
Percentage of edible mushrooms, grouped by values on the "spore-print-color" column



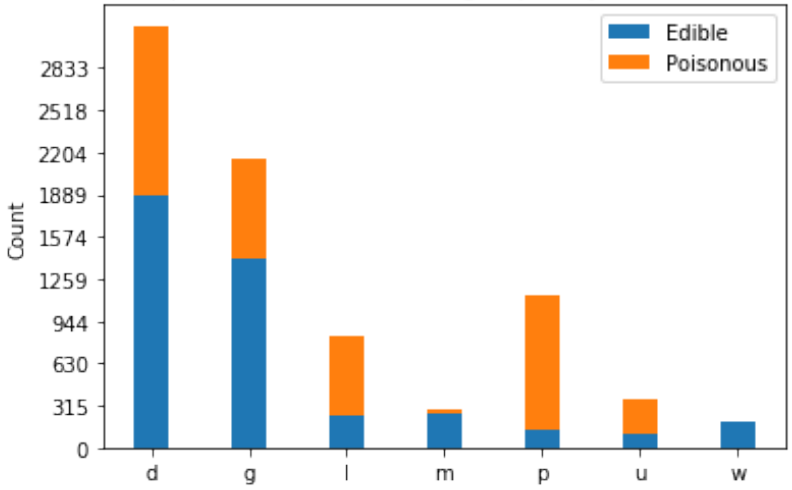
Number of edible and poisonous mushrooms, grouped by values on the "population" column

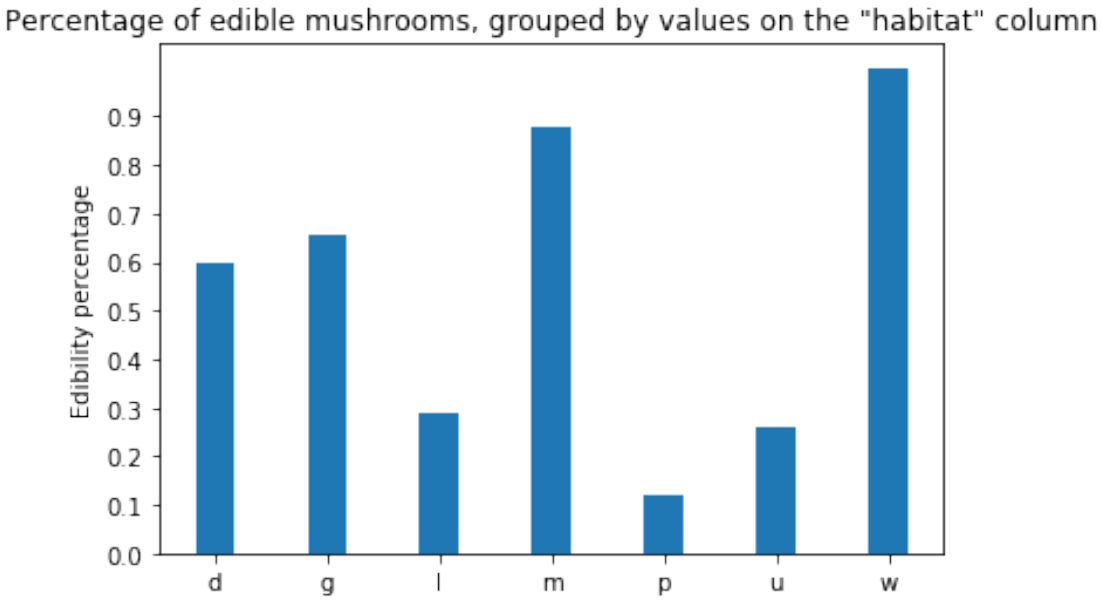


Percentage of edible mushrooms, grouped by values on the "population" column



Number of edible and poisonous mushrooms, grouped by values on the "habitat" column






```
In [7]: # Create anoter random forest using only the odor as feature.
        Xo = mushrooms['odor'].copy()
        Xo = pd.get_dummies(Xo)
        Xo_train, Xo_test, yo_train, yo_test = train_test_split(Xo, y, test_size=0.33, random_s

        odor_classifier = RandomForestClassifier(n_estimators=4, max_leaf_nodes=7, random_state
        odor_classifier.fit(Xo_train, yo_train)

        yo_pred_rf = odor_classifier.predict(Xo_test)

        accuracy_score(y_true = yo_test, y_pred = yo_pred_rf)

Out[7]: 0.9858261842596047

In [8]: # Check the accuracy when using all features except for odor.
        Xno = mushrooms.drop('class', axis=1).drop('odor', axis=1)
        Xno = pd.get_dummies(Xno)
        Xno_train, Xno_test, yno_train, yno_test = train_test_split(Xno, y, test_size=0.33, ra

        no_odor_classifier = RandomForestClassifier(n_estimators=4, max_leaf_nodes=7, random_st
        no_odor_classifier.fit(Xno_train, yno_train)

        yno_pred_rf = no_odor_classifier.predict(Xno_test)

        accuracy_score(y_true = yno_test, y_pred = yno_pred_rf)

Out[8]: 0.9306229019022753
```

1.2.3 Check mushroom hunting tips

In [9]:
```
# Define filters for each tip:
# 1. Choose mushrooms without white gills.
gills_tip = mushrooms['gill-color'] != 'w'

# 2. Select mushrooms without red on the cap or stem.
cap_color_tip = mushrooms['cap-color'] != 'r'
stem_tip = (mushrooms['stalk-color-above-ring'] != 'r') & (mushrooms['stalk-color-belo

# 3. Look for mushrooms without scales on the cap.
cap_surface_tip = mushrooms['cap-surface'] != 'y'

# 4. Seek out mushrooms without a ring around the stem
cap_surface_tip = mushrooms['ring-number'] == 'n'

# Combine filters and make predictions.
hunt_filter = gills_tip & cap_color_tip & stem_tip & cap_surface_tip & cap_surface_tip
hunt_pred = np.where(hunt_filter == True, 'e', 'p')

# Check accuracy.
accuracy_score(y_true = y, y_pred = hunt_pred)
```

Out[9]:
```
0.4798129000492368
```

In [10]:
```
# Check accuracy of (only) poisonous mushrooms.
poisonous = y == 'p'
accuracy_score(y_true = y[poisonous], y_pred = hunt_pred[poisonous])
```

Out[10]:
```
0.9954034729315628
```

In [11]:
```
# Count number of mushroms predicted as edibles.
np.count_nonzero(hunt_pred == 'e')
```

Out[11]:
```
18
```

In [12]:
```
# Count number of mushroms predicted as poisonous.
np.count_nonzero(hunt_pred == 'p')
```

Out[12]:
```
8106
```