100 XP

Exercise - Work with data to predict missing values

8 minutes

This module requires a sandbox to complete. You have used 1 of 10 sandboxes for today. More sandboxes will be available tomorrow.

Activate sandbox

Runtime File Edit View

Run all Kernel Compute not connected

Alt + A

Exercise: Titanic Dataset - Visualising Different Types of Data

To build better machine learning models we should understand the available data. This usually means both:

1. data visualization
2. understanding the kind of data we have available

In this module, we'll practice cleaning our Titanic dataset, and visualization of different kinds of data, especially

- continuous
- ordinal
- categorical
- simple identity column

data types.

```
import pandas as pd

# Load data from our dataset file into a pandas dataframe
!wget https://raw.githubusercontent.com/MicrosoftDocs/mslearn-introduction-to-machine-learning,
!wget https://raw.githubusercontent.com/MicrosoftDocs/mslearn-introduction-to-machine-learning,
dataset = pd.read_csv('titanic.csv', index_col=False, sep=",", header=0)

# Let's take a look at the data
dataset.head()
```

Take a careful look at the columns, and try to identify those columns holding continuous, ordinal, categorical, or identity data.

We can display a brief summary of the *datatypes* with panda's `info()` method:

```
dataset.info()
```

We can see several columns stored as numerical data (the `int64` or `float64` types), while others contain more complex data types (those with `object` as Dtype)

Visualising Ordinal Data

Let's visualize some ordinal data. We have available:

1. `Pclass` - the ticket class
2. `Parch` - the number of parents or children on the ship
3. `sibsp` - the number of siblings or spouses on the ship

We can view ordinal data with almost any kind of graph. We'll start with a simple histogram that describes relationships between the ticket class and the likelihood of survival.

```
import graphing

graphing.histogram(dataset, label_x='Pclass', label_y='Survived', histfunc='avg', include_boxp
```

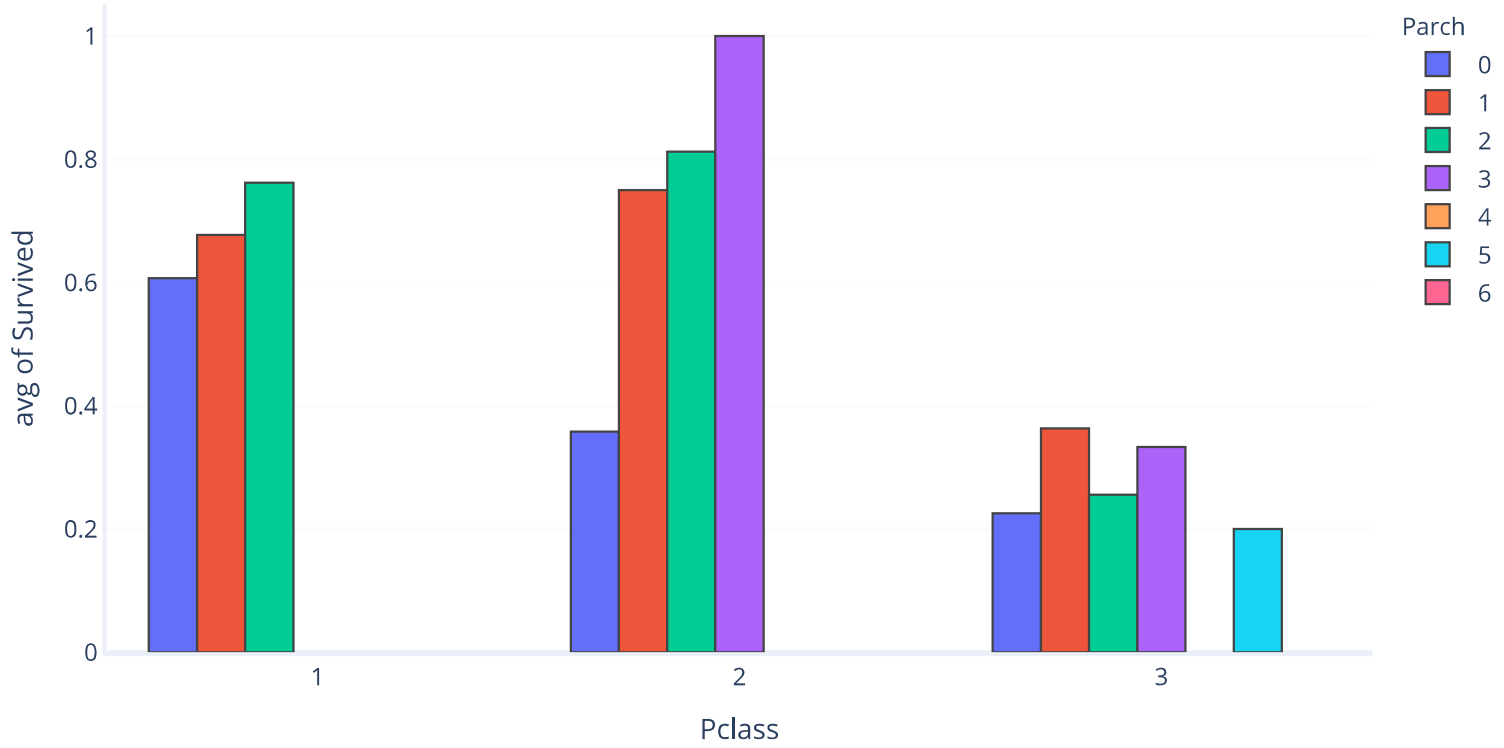
The box and whisker plot (top) shows that at least half the people had third-class tickets - note how the median and maximum of the plot both sit at `Pclass = 3` .

The histogram shows that people in second and third class tended not to survive the wreck.

Let's look at how survival varies, depending on whether a passenger had parents or children on the ship

```
graphing.multiple_histogram(dataset,
                            label_x='Pclass', # group by ticket class
                            label_group="Parch", # colour by no parents or children
                            label_y='Survived',
                            histfunc="avg")
```

[7] ✓



For first and second class ticket holders, people in larger family groups appear to have had better rates of survival. However, this doesn't seem to be the case for third class passengers.

Lastly, let's see if those with different ticket types tended to be in different sized families. For data spread analysis, a box and whisker is a nice alternative to histograms.

```
graphing.box_and_whisker(dataset, label_x="Pclass", label_y="SibSp")
```

Most values are zero. This shows that most people traveled without siblings and without a partner. There are no obvious differences in this value between the different ticket classes.

Visualising Continuous Data

Continuous data are usually best viewed using either:

1. An XY scatter plot, especially for relationships between two continuous features
2. Histograms or Box and Whisker plots, to look at the spread of data

Our dataset has `Age` and `Fare` as continuous data columns. Let's view them:

```
graphing.scatter_2D(dataset, label_x="Age", label_y="Fare")
```

We don't see an obvious relationship between `Age` and `Fare`.

Does the cost of a fare, or the person's age, have any relationship with likelihood of survival?

```
# Plot Fare vs Survival
graphing.histogram(dataset, label_x="Fare", label_y="Survived", histfunc="avg", nbins=30, titl

# Plot Age vs Survival
graphing.histogram(dataset, label_x="Age", label_y="Survived", histfunc="avg", title="Age vs S
```

The boxplot (top) of the first figure shows us that most people held tickets that cost less than £25, and the histogram shows us that people with more expensive tickets tended to survive.

Our second figure indicates passengers were about 30 years old on average, and that most children under 10 years old survived, unlike most adults.

Visualising Categorical Data

Our Titanic dataset has the following *categorical* columns:

- Sex (Male, Female)
- `Embarked` - the port of ambarkation (C, Q, or S)
- `Cabin` (many options)
- `Survival` (0 = no, 1 = yes)

Categorical data are usually viewable in a similar way to ordinal data, but with data viewed as order-less groups. Alternatively, categories appear as colors, or groups, in other kinds of plots.

Plotting categorical data against other categorical data shows how data is clustered. This is little more than a colored table. Let's do this now:

```
import plotly.graph_objects as go
import numpy as np

# Create some simple functions
# Read their descriptions to find out more
def get_rows(sex, port):
    '''Returns rows that match in terms of sex and embarkment port'''
    return dataset[(dataset.Embarked == port) & (dataset.Sex == sex)]

def proportion_survived(sex, port):
    '''Returns the proportion of people meeting criteria who survived'''
    survived = get_rows(sex, port).Survived
    return np.mean(survived)

# Make two columns of data - together these represent each combination
# of sex and embarkment port
```

```
# Of sex and embarkment port
sexes = ["male", "male", "male", "female","female", "female"]
ports = ["C", "Q", "S" ] * 2

# Calculate the number of passengers at each port + sex combination
passenger_count = [len(get_rows(sex, port)) for sex,port in zip(sexes, ports)]

# Calculate the proportion of passengers from each port + sex combination who survived
passenger_survival = [proportion_survived(sex, port) for sex,port in zip(sexes, ports)]

# Combine into a single data frame
table = pd.DataFrame(dict(
    sex=sexes,
    port=ports,
    passenger_count=passenger_count,
    passenger_survival_rate=passenger_survival
))

# Make a bubble plot
# This is just a scatter plot but each entry in the plot
# has a size and colour. We set colour to passenger_survival
# and size to the number of passengers
graphing.scatter_2D(table,
                    label_colour="passenger_survival_rate",
                    label_size="passenger_count",
                    size_multiplier=0.3,
                    title="Bubble Plot of Categorical Data")
```

It appears that women have a much higher survival rate than men, but there were more men on the ship.

We can also see that most people boarded at Port `S` ("Southampton"). It does seem that there is a weak relationship between the port of boarding and survival.

No compute Compute not connected Viewing Kernel not connected

Next unit: One-hot vectors

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆