Filtering Data

This lesson focuses on how to filter data with Pandas.

WE'LL COVER THE FOLLOWING

Filtering

Filtering is the process of extracting a subset of your data based on some condition or constraint. These conditions can be on the values that the data items take. We filter data when we wish to look at a smaller part of the whole data. For instance, we may want:

- the data in a particular period of the year
- the data of the highest selling items
- the data for a specific group of items
- to remove extra or useless data

Data filtering is done on almost every dataset before doing any analysis. Let's look at some examples using our California Housing Dataset.

housing.csv

```
import pandas as pd
df = pd.read_csv('housing.csv')
print(df.head())
```

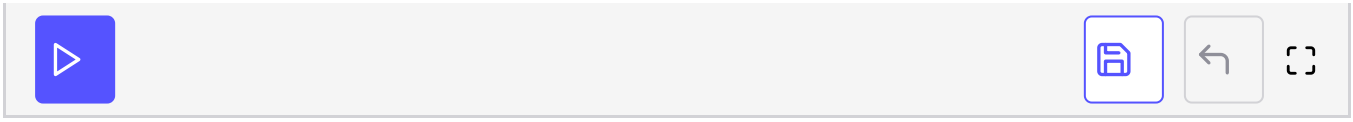
The California Housing Dataset

Let's say we want to see the data for all the housing blocks that are close to the

ocean. From the above code block, we know that there is text in the `ocean_proximity` column instead of numbers. We will first find out how many distinct values there are in this column and then decide how to filter rows for our requirement.

```
import pandas as pd
df = pd.read_csv('housing.csv')

# Find all distinct values in ocean_proximity
unique_values = df['ocean_proximity'].unique()
print(unique_values)
```



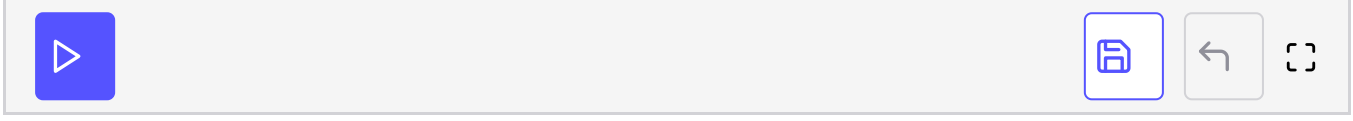
We have used the function `unique()` on the `ocean_proximity` column in **line 5** to obtain all the unique values in this column. We see that there are 5 distinct values in the `ocean_requirement` column, so we have to find those rows that have `NEAR OCEAN` in them.

```
import pandas as pd
df = pd.read_csv('housing.csv')

print('Shape of original dataframe : ', df.shape)

# filter dataframe
condition = df['ocean_proximity'] == 'NEAR OCEAN'
filtered_df = df[condition]

print('Shape of filtered dataframe : ', filtered_df.shape)
print(filtered_df.head())
```



To filter, we have to provide a condition to the dataframe which we set in **line 7**. It gives us a list of booleans (True/False) against each row number. The values in the list are `True` for rows that satisfy our condition. When we provide this list to the dataframe in **line 8**, it gives us all the rows that have `True` against them, therefore giving us a filtered dataframe. We see from the output of the code that only 2658 rows out of total 20640 satisfy our condition.

Now let's look at some other examples. Can you point out what lines 6, 11, 16, 22 and 26 give in the code below without looking at the

explanation?

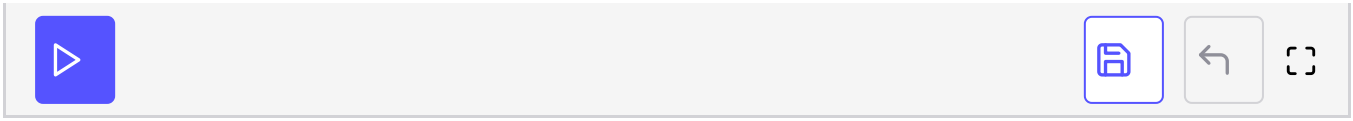
```
import pandas as pd
df = pd.read_csv('housing.csv')
print('Shape of original dataframe : ', df.shape)

condition_1 = df['population'] < 10000
filtered_df = df[condition_1]
print('Shape of filtered dataframe 1: ', filtered_df.shape)


condition_2 = df['ocean_proximity'] != 'INLAND'
filtered_df = df[condition_2]
print('Shape of filtered dataframe 2: ', filtered_df.shape)


condition_3 = (df['population'] > 15000) & (df['households'] < 5000)
filtered_df = df[condition_3]
print('Shape of filtered dataframe 3: ', filtered_df.shape)


condition_list = ['NEAR OCEAN', 'NEAR BAY']
condition_4 = df['ocean_proximity'].isin(condition_list)
filtered_df = df[condition_4]
print('Shape of filtered dataframe 4: ', filtered_df.shape)
```



In **line 5** we have written the condition that the `population` column should have values less than 10000. While `condition_2` in **line 10** says that the values in the `ocean_proximity` column should not be `INLAND.`

We can combine two or more conditions using the `&` operator, which we have done in **line 15**. This condition will filter the dataframe so that all the rows that have `population` greater than 15000 and `households` less than 5000 will be extracted.

We can also provide a list of values and filter them so that the values of the specified column must be one of the values in the provided list. This is done using the `isin()` function as we have done in **line 21**. This filtration will extract those rows that have `ocean_proximity` values from `condition_list.`

Now that we know how to filter we will focus on how to apply functions to data.