This code reads in two JSON files containing Yelp business and review data and uses them to analyze and visualize the reviews of restaurants in the dataset. The code starts by importing pandas and numpy libraries, and then reads in the Yelp business and review data files using the `pd.read\_json()` method, which reads JSON files into a pandas DataFrame. It sets `chunksize=1000000` to read the files in chunks of 1 million records, which is useful when working with very large files.

The code then iterates over the first chunk of businesses and reviews and selects a subset of businesses that are open, filtering out those that are closed. It further filters these businesses to only include those with the category "Restaurant" or "Restaurants" in the category column.

The code then uses seaborn and matplotlib libraries to create two bar plots. The first bar plot shows the count of reviews for each restaurant in the dataset, grouped by their star rating. The second bar plot shows the top 20 restaurants with the highest number of reviews that have a 5-star rating, and the top 20 restaurants with the highest number of reviews that have a rating of less than 3.

Next, the code defines a function `extract\_keys()` that takes two arguments, `attr` and `key`. This function checks if the attribute is None and returns an empty dictionary if so. Otherwise, it checks if the key is present in the attribute dictionary and pops it out if it is, returning the value for that key. It also defines a function `str\_to\_dict()` that takes a string argument and converts it to a dictionary using the ast library. These two functions are used to extract nested attributes from the attributes column of the restaurants DataFrame.

The code then creates a new DataFrame `df\_attr` by concatenating various attributes extracted from the attributes column of the restaurants DataFrame. It further creates dummy variables for each of the columns in `df\_attr` using the `pd.get\_dummies()` method. It also creates dummy variables for the categories column of the restaurants DataFrame.

Finally, the code creates a new DataFrame `df\_final` by concatenating `df\_attr\_dummies`, `df\_categories\_dummies`, and the `stars` and `name` columns of the original DataFrame. It then maps the star ratings to integers from 1 to 5 using a dictionary.

KNN

The code implements the k-nearest neighbors (k-NN) algorithm for a classification task on a dataset df\_final which contains information about restaurants, including their features (attributes) such as ambience, food, service, etc. and their associated stars ratings (1-5). The goal is to predict the star rating of a restaurant given its features.

Here is how the code works:

First, the dataset is split into the feature matrix X (all columns except the last two) and the target vector y (the stars column).

The train\_test\_split function from sklearn.model\_selection is used to randomly split the data into a training set and a test set, with 80% of the data used for training and 20% used for testing. The random\_state parameter is set to 1 to ensure reproducibility.

An instance of the KNeighborsClassifier class is created with n\_neighbors=20, which means the algorithm will consider the 20 closest neighbors to each point in the dataset when making predictions.

The fit method of the KNeighborsClassifier object is called on the training set to train the model.

The predict method is called on the test set to make predictions.

The accuracy of the model is computed on both the training set and the test set using the score method of the KNeighborsClassifier object.

The output shows that the KNN model achieved an accuracy of approximately 0.59 (59%) on the training set and 0.53 (53%) on the test set. This suggests that the model may be overfitting the training data, as the accuracy is significantly higher on the training set than the test set.

The validation set, which is the last row of the original dataset, is extracted and stored in the test\_set variable.

Output: Validation set (Restaurant name): The Plum Pit

A new instance of the KNeighborsClassifier class is created and trained on the entire dataset except for the validation set.

The kneighbors method of the KNeighborsClassifier object is called on the validation set to find the k-nearest neighbors (in this case k=20) to the validation set. The distances and indices of the nearest neighbors are stored in the distances and indices variables, respectively.

The indices of the nearest neighbors are used to extract the corresponding rows from the original dataset.

A new DataFrame final\_table is created that contains the distances and indices of the nearest neighbors, and this DataFrame is joined with the original dataset to get the corresponding restaurant names and star ratings.

The resulting DataFrame is displayed, showing the five closest restaurants to the validation set and their star ratings.