Birla Institute of Technology and Science Pilani, Hyderabad Campus

1st Semester 2024-25, BITS F464: Machine Learning

Assignment No: 1 Date Given: 14.08.2024 Date of Sub: **20.08.2024**

(Note: Data Exploration, Pre-processing and Wrangling using Scikit Learn) Max. Marks: 5

This assignment is about data exploration, pre-processing and wrangling which are some of the important initial steps in the Machine Learning software development life cycle as discussed in the class before the ML model is built, deployed and maintained. Raw data is often messy, inconsistent, and incomplete. Data exploration and wrangling involves understanding the characteristics of your data like imputing mean or median for missing (Not a Number) values, removing duplicates, normalizing or standardizing the data, detecting outliers, encoding categorical values, creating new features, reducing the dimension, finding out how many values are less than 50% percentile etc. in your dataset.

You are given with a CSV file (attached here) containing California's house price dataset (20,640 houses) with features as latitude, longitude, how old, no. of rooms, bedrooms, price, ocean proximity etc. Use this data to carry out the following:

1. Install Anaconda distribution for Python which is free and open source on your local computer (PC or Laptop). Go to the Navigator and launch a Jupyter Notebook as we discussed in the class. Can alternatively utilize Google Colab for performing the tasks mentioned.
2. Import pandas library for dealing with the data. Read data from the data.csv (given here in this assignment) containing housing price data into a Panda's DataFrame. Display how many rows are there in the file and its contents. Reading from the file that is available locally can be done by using:

   ```
   data = pd.read_csv("<path for the data.csv file>")
   ```
3. Display descriptive statistics by normally including those stats that summarize the central tendency, dispersion and shape of a dataset's distribution. Generate descriptive statistics using describe method on the pandas DataFrame. Display stats like mean, median and standard deviation.
4. Develop a Python code snippet that effectively visualizes the descriptive statistics of a given DataFrame. This should include (min, max, mean, median, standard deviation).
5. Find out the columns which have the missing values present in them and also identify the number of tuples which have these missing values. Instead of removing these rows that have missing values in the given CSV file, impute Median of the values of the respective columns by calling the median method on the respective attributes of the DataFrame. You could also impute with mean or mode if you so wish.
6. To better understand the importance of imputing the mean/median/mode values to the given DataFrame, Plot the distribution of the columns which have this missing values before and after filling in the missing values. (Try creating a function for doing the task)
7. Some rows in the CSV file might have duplicates. If so, remove those duplicates by calling drop_duplicates method on the DataFrame.
8. Use Matplotlib to create a Python code snippet to generate a scatter plot visualizing the relationship between longitude and latitude, with the color of each data point representing the corresponding median house value. Optional: - (Enhance the plot's aesthetics by setting the figure size, adding a colorbar, labeling axes, and providing a descriptive title).

   Expected Output: - (Something similar to the Fig.1 as shown in the next page alongwith the actual California map)

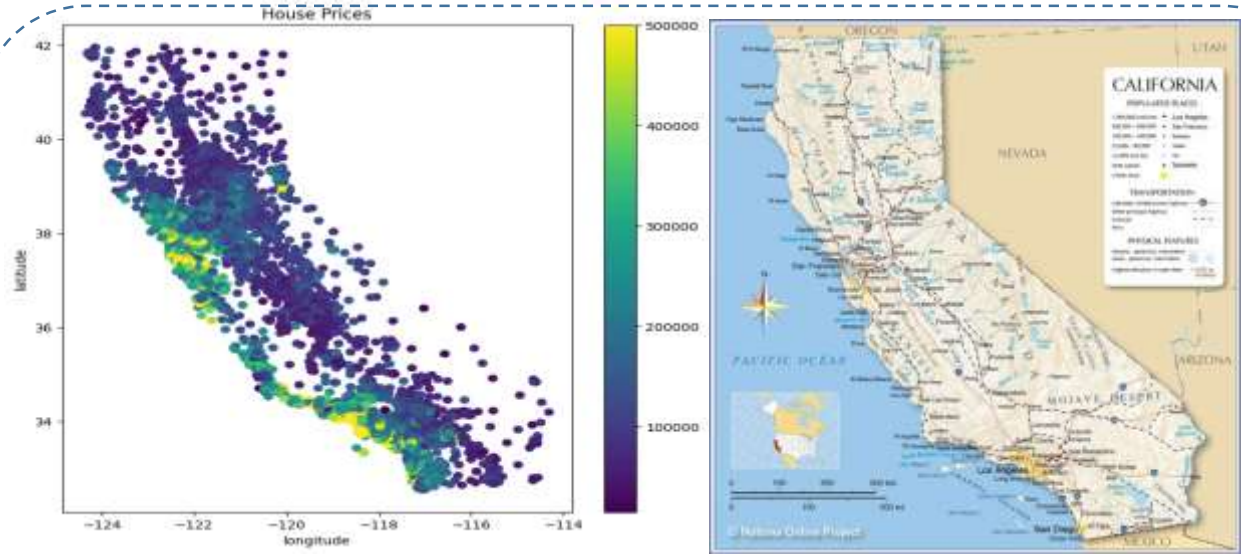


Fig.1: using plt.scatter(df['longitude'] , df['latitude'] , c = df['median_house_value'])

9. Now try making out some observations out of specific columns in the DataFrame and plot them either in form of numerical observations or as a plot showing the observations you made. (Ex: Plot of number of houses having total Bedrooms less than 1500, median income more than 10, etc.)

```
#one example of how to do the above task. (Try doing something else with the data)
data[data['total_rooms']>=1500].shape
```

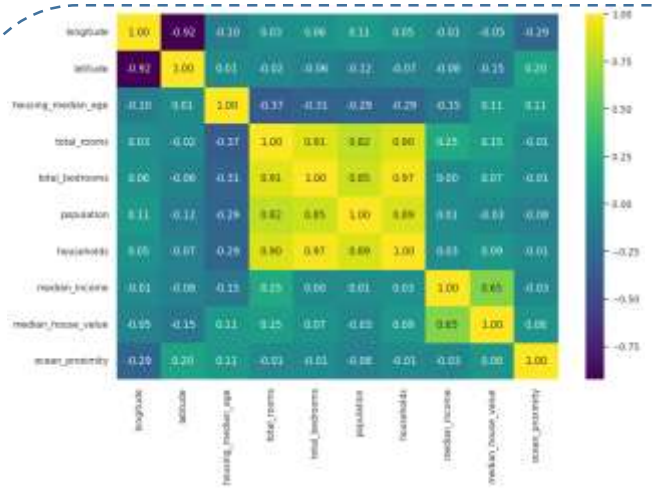
10. Analyze the relationship between ocean_proximity and other numerical features within the DataFrame. Perform exploratory data analysis by (on subgroups):
    - Grouping the data by ocean_proximity and calculating summary statistics for each group.
    - Visualizing the distribution of a target variable (e.g., median_house_value) across different ocean_proximity categories.
    - Consider encoding categorical ocean_proximity values into numerical representations for potential modeling purposes.
    - Below are some snippets for your understanding:

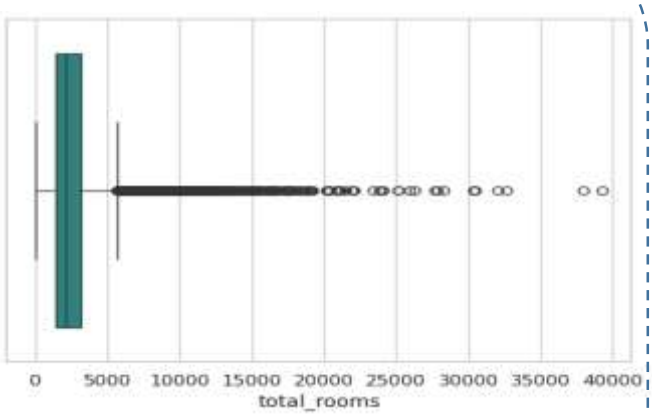
```
#grouping the given DataFrame according to `ocean_proximity.
grouped = data.groupby('ocean_proximity')
#encoding the values into labels.
ocean_proximity_le = LabelEncoder()
data['ocean_proximity'] = ocean_proximity_le.fit_transform(data['ocean_proximity'])
```

11. After performing the label encoding of the ocean_proximity feature: -
    a) Create a correlation heatmap of the dataset to identify relationships between numerical features.
    b) Generate box plots for all numerical features to visualize their distribution and potential outliers.
    c) Analyze the heatmap and box plots for all the features to draw insights about the dataset, such as:
        i) Strong correlations between features
        ii) Presence of outliers in the numerical data
        iii) Potential feature importance based on their distribution

d) Identify and visualize characteristics of premium houses based on ocean proximity, cost per square foot, and median income. (Plot a Box-Plot for doing so)

Expected Output (for b and c): - (you should be getting something similar to figures shown below)

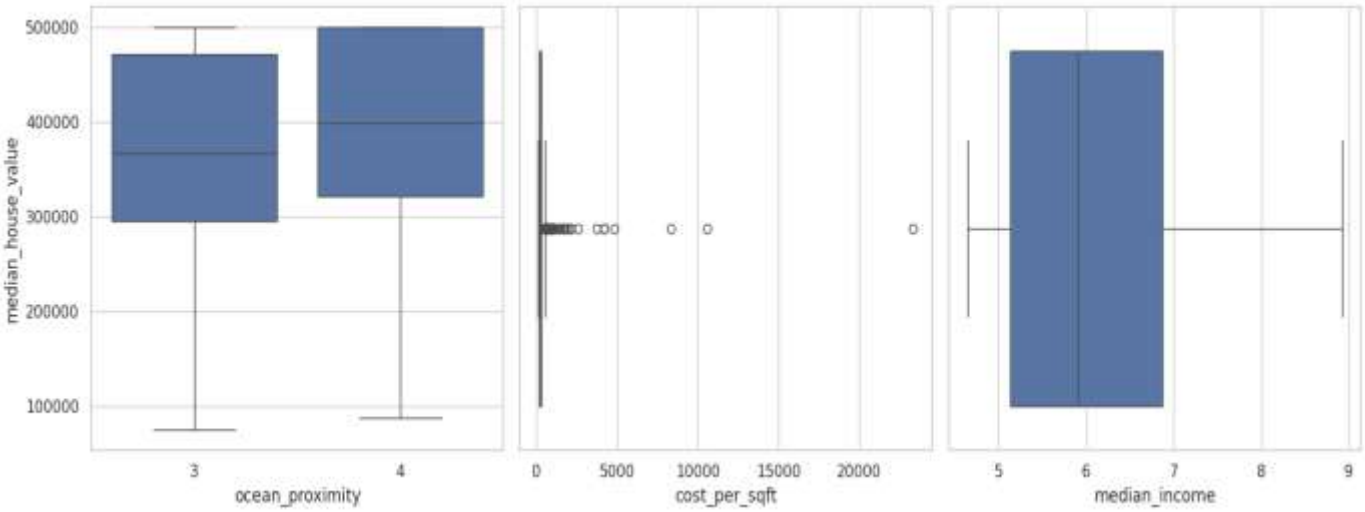




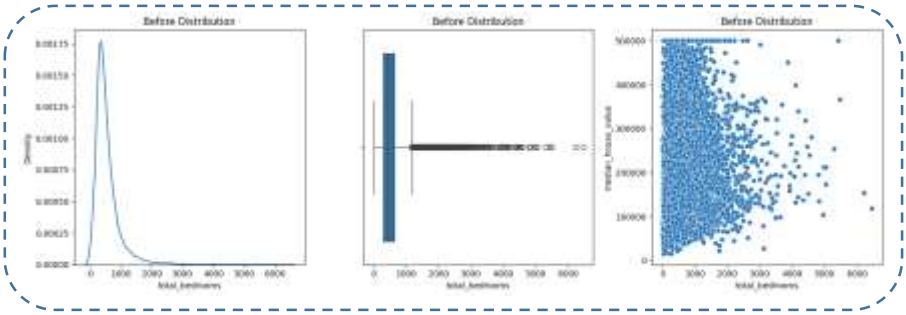
(Fig.2)

```
import seaborn as sns
sns.heatmap(data.corr(), annot=True)
sns.boxplot(data=df, x='total_rooms', palette='viridis')
```

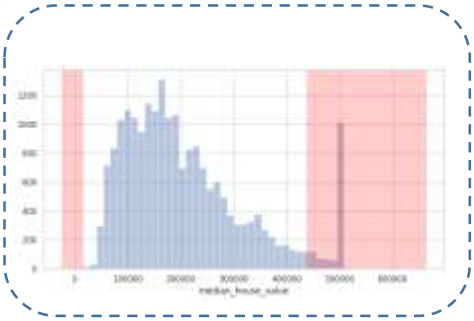
Expected Output: - (Fig.3 for (d))



12. Develop a Python function to visualize the distribution and relationship of a specified numerical column within a dataset. Create a figure with three subplots: Plot the Kernel Density Estimate (KDE) of the column in the first subplot, plot a boxplot of the column in the second subplot, and plot a scatterplot of the column against 'median_house_value' in the third subplot. Present the plots in a clear and informative manner. Expected Output: - (Something similar to Fig.4)



(Fig.4)



(Fig.5)

13. Identify and visualize outliers in a numerical column of a dataset based on standard deviation.
    - Assuming a normal distribution, calculate the lower and upper bounds for data points within approximately 95% of the data using two standard deviations from the mean.
    - Identify data points that fall outside these bounds as outliers.
    - Calculate the total number of outliers.
    - Visually represent the data distribution, highlighting the outlier regions.

    Expected Output (sample shown in Fig. 5).

14. Perform feature scaling on the given DataFrame using StandardScaler and MinMaxScaler from Scikit-learn.
    - Apply both scaling techniques to the numerical features of the DataFrame.
    - Compare the resulting ranges of the scaled data for both methods.
    - Explore the potential benefits and drawbacks of each scaling technique in the context of the given dataset. (Just know what each scaling technique actually does and how are they different from one another)
    - Investigate additional scaling methods commonly employed in practice, such as RobustScaler, QuantileScaler, and PowerTransformer.
    - Justify and analyze the selection of these alternative scaling methods based on their theoretical underpinnings and potential advantages over StandardScaler and MinMaxScaler.

```
#splitting the data into x and y for better analysis.
x = data_.drop(['median_house_value'] , axis = 1).values
y= data_['median_house_value' ].values

#Fitting the given x values using minmax and standard scaler.
from sklearn.preprocessing import MinMaxScaler,StandardScaler
mm_scaler = MinMaxScaler()
x_mm = mm_scaler.fit_transform(x)
s_scaler = StandardScaler()
x_s = s_scaler.fit_transform(x)
```

15. Submission Instructions:

You are free to form your own group of MAXIMUM three students. Clean your Notebook code (ipynb) and name your submission file as IDNo.ipynb. Write a readme.text with your group members name and ID nos. Compress these two files into a single Zip, and name your Zip file using your idno (in lowercase). Make only one submission on behalf of your group in Google Class Assignment Submission page. Deadline for submitting your work is 20th August 2024 midnight.

References:

1. https://www.w3schools.com/python/pandas/default.asp
2. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html
3. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
4. https://www.statology.org/matplotlib-distribution-plot/
5. https://docs.anaconda.com/free/anaconda/index.html
6. https://www.kaggle.com/datasets/camnugent/california-housing-prices
7. https://www.youtube.com/playlist?list=PLZoTAELRMXVPQyArDHyQVjQxjj_YmEuO9
8. https://www.youtube.com/watch?v=kUsNb_gOo_s

(Chittaranjan Hota, I/C)