US Hospital Customers Satisfaction - Maryam (updated)

December 5, 2023

1 Importing Modules

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.base import BaseEstimator, TransformerMixin

from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪mean_squared_error, r2_score

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

# Setting environment to ignore future warnings
import warnings
warnings.simplefilter('ignore')
%matplotlib inline
```

2 1. Data Loading and Overview

2.1 1.1 Data Loading

[4]:
```
# File paths for the dataset files
files = ['data/cms_hospital_patient_satisfaction_2020.csv',
    'data/cms_hospital_patient_satisfaction_2019.csv',
    'data/cms_hospital_patient_satisfaction_2018.csv',
    'data/cms_hospital_patient_satisfaction_2017.csv',
    'data/cms_hospital_patient_satisfaction_2016.csv']
```

2.2 1.2 Merging Data

[5]:
```
# Create an empty list to store the dataframes
dataframes = []

# Load each file into a dataframe and append it to the list
for files in files:
    hospital_df = pd.read_csv(files)
    dataframes.append(hospital_df)

# Concatenate all dataframes into one complete dataset
hospital_df = pd.concat(dataframes, ignore_index=True)
```

2.3 1.3 First Look of Data

[6]:
```
display(hospital_df.head(3))
```

```
  Facility ID                     Facility Name                  Address  \
0      010001  SOUTHEAST ALABAMA MEDICAL CENTER  1108 ROSS CLARK CIRCLE
1      010001  SOUTHEAST ALABAMA MEDICAL CENTER  1108 ROSS CLARK CIRCLE
2      010001  SOUTHEAST ALABAMA MEDICAL CENTER  1108 ROSS CLARK CIRCLE

     City State  ZIP Code County Name    Phone Number HCAHPS Measure ID  \
0  DOTHAN    AL     36301     HOUSTON  (334) 793-8701      H_COMP_1_A_P
1  DOTHAN    AL     36301     HOUSTON  (334) 793-8701     H_COMP_1_SN_P
2  DOTHAN    AL     36301     HOUSTON  (334) 793-8701      H_COMP_1_U_P

                                    HCAHPS Question  \
0  Patients who reported that their nurses "Alway…
1  Patients who reported that their nurses "Somet…
2  Patients who reported that their nurses "Usual…

                         HCAHPS Answer Description Patient Survey Star Rating  \
0                Nurses "always" communicated well             Not Applicable
1  Nurses "sometimes" or "never" communicated well             Not Applicable
2               Nurses "usually" communicated well             Not Applicable
```

```
  Patient Survey Star Rating Footnote HCAHPS Answer Percent  \
0                                NaN                    77
1                                NaN                     7
2                                NaN                    16

  HCAHPS Answer Percent Footnote HCAHPS Linear Mean Value  \
0                            NaN           Not Applicable
1                            NaN           Not Applicable
2                            NaN           Not Applicable

  Number of Completed Surveys Number of Completed Surveys Footnote  \
0                         535                                   NaN
1                         535                                   NaN
2                         535                                   NaN

  Survey Response Rate Percent Survey Response Rate Percent Footnote  \
0                           22                                    NaN
1                           22                                    NaN
2                           22                                    NaN

   Start Date    End Date  Year         Hospital Type  \
0  07/01/2018  06/30/2019  2020  Acute Care Hospitals
1  07/01/2018  06/30/2019  2020  Acute Care Hospitals
2  07/01/2018  06/30/2019  2020  Acute Care Hospitals

                          Hospital Ownership Emergency Services  \
0  Government - Hospital District or Authority               Yes
1  Government - Hospital District or Authority               Yes
2  Government - Hospital District or Authority               Yes

  Meets criteria for promoting interoperability of EHRs  \
0                                                     Y
1                                                     Y
2                                                     Y

  Hospital overall rating Hospital overall rating footnote  \
0                       2                              NaN
1                       2                              NaN
2                       2                              NaN

  Mortality national comparison Mortality national comparison footnote  \
0    Below the national average                                    NaN
1    Below the national average                                    NaN
2    Below the national average                                    NaN

  Safety of care national comparison  \
0       Same as the national average
1       Same as the national average
```

```
2       Same as the national average

  Safety of care national comparison footnote Readmission national comparison  \
0                                          NaN      Below the national average
1                                          NaN      Below the national average
2                                          NaN      Below the national average

  Readmission national comparison footnote  \
0                                      NaN
1                                      NaN
2                                      NaN

  Patient experience national comparison  \
0             Below the national average
1             Below the national average
2             Below the national average

  Patient experience national comparison footnote  \
0                                             NaN
1                                             NaN
2                                             NaN

  Effectiveness of care national comparison  \
0              Same as the national average
1              Same as the national average
2              Same as the national average

  Effectiveness of care national comparison footnote  \
0                                                NaN
1                                                NaN
2                                                NaN

  Timeliness of care national comparison  \
0           Same as the national average
1           Same as the national average
2           Same as the national average

  Timeliness of care national comparison footnote  \
0                                             NaN
1                                             NaN
2                                             NaN

  Efficient use of medical imaging national comparison  \
0                       Same as the national average
1                       Same as the national average
2                       Same as the national average

  Efficient use of medical imaging national comparison footnote
```

```
0                                                   NaN
1                                                   NaN
2                                                   NaN
```

3 2. Summary and Understanding of Data

3.1 2.1 Exploratory Data Analysis

[7]:
```
# Function to perform all EDA
def eda(hospital_df, name=""):
    # Displaying basic details
    print(f"EDA of {name} Dataset is........")
    print(f"Size: {hospital_df.size}")
    print(f"Columns: {hospital_df.shape[1]}")
    print(f"Records: {hospital_df.shape[0]}")
    print("*"*50, "\n")

    # Displayng Top 4 records of Data
    print("First Look of Data: ")
    display(hospital_df.head())
    print("*"*50, "\n")

    # Getting Numerical columns and Categorical columns
    cat_col = hospital_df.select_dtypes(object).columns
    num_col = hospital_df.select_dtypes(np.number).columns

    # Displaying the Numerical Columns
    print("Dataset has following Numerical Columns: ")
    if len(num_col) == 0:
        print('''\t"No Numerical Column exist."''', "\n")
    else:
        for i, j in enumerate(num_col):
            print(f"{i+1}- {j}")

    # Displaying the Categorical Columns
    print("*"*50)
    print("Dataset has following Categorical Columns: ")
    if len(cat_col) == 0:
        print("\tNo Categorical Column exist.")
    else:
        for i, j in enumerate(cat_col):
            print(f"{i+1}- {j}")
    print("*"*50, "\n")

    # Displaying info of Data e.g., Null values, data types etc
    print("Information of Data is as follows: ")
    display(hospital_df.info())
```

```
    print("*"*50, "\n")

    # Displaying Statistical properties
    print("Statistical Properties of Data: ")
    display(hospital_df.describe(include="all"))
    print("*"*50, "\n")
```

[8]:
```
eda(hospital_df, "Customer Satisfaction of US Hospital")
```

```
EDA of Customer Satisfaction of US Hospital Dataset is…
Size: 71108369
Columns: 43
Records: 1653683
**************************************************

First Look of Data:
```

	Facility ID	Facility Name	Address	City	State	ZIP Code	County Name	Phone Number	HCAHPS Measure ID	HCAHPS Question	HCAHPS Answer Description	Patient Survey Star Rating	Patient Survey Star Rating Footnote	HCAHPS Answer Percent
0	010001	SOUTHEAST ALABAMA MEDICAL CENTER	1108 ROSS CLARK CIRCLE	DOTHAN	AL	36301	HOUSTON	(334) 793-8701	H_COMP_1_A_P	Patients who reported that their nurses "Alway…	Nurses "always" communicated well	Not Applicable	NaN	77
1	010001	SOUTHEAST ALABAMA MEDICAL CENTER	1108 ROSS CLARK CIRCLE	DOTHAN	AL	36301	HOUSTON	(334) 793-8701	H_COMP_1_SN_P	Patients who reported that their nurses "Somet…	Nurses "sometimes" or "never" communicated well	Not Applicable	NaN	7
2	010001	SOUTHEAST ALABAMA MEDICAL CENTER	1108 ROSS CLARK CIRCLE	DOTHAN	AL	36301	HOUSTON	(334) 793-8701	H_COMP_1_U_P	Patients who reported that their nurses "Usual…	Nurses "usually" communicated well	Not Applicable	NaN	16
3	010001	SOUTHEAST ALABAMA MEDICAL CENTER	1108 ROSS CLARK CIRCLE	DOTHAN	AL	36301	HOUSTON	(334) 793-8701	H_COMP_1_LINEAR_SCORE	Nurse communication - linear mean score	Nurse communication - linear mean score	Not Applicable	NaN	NaN
4	010001	SOUTHEAST ALABAMA MEDICAL CENTER	1108 ROSS CLARK CIRCLE	DOTHAN	AL	36301	HOUSTON	(334) 793-8701	H_COMP_1_STAR_RATING	Nurse communication - star rating	Nurse communication - star rating	3	NaN	NaN

```
2                                NaN                    16
3                                NaN         Not Applicable
4                                NaN         Not Applicable

  HCAHPS Answer Percent Footnote HCAHPS Linear Mean Value  \
0                           NaN           Not Applicable
1                           NaN           Not Applicable
2                           NaN           Not Applicable
3                           NaN                       90
4                           NaN           Not Applicable

  Number of Completed Surveys Number of Completed Surveys Footnote  \
0                         535                                  NaN
1                         535                                  NaN
2                         535                                  NaN
3                         535                                  NaN
4                         535                                  NaN

  Survey Response Rate Percent Survey Response Rate Percent Footnote  \
0                           22                                   NaN
1                           22                                   NaN
2                           22                                   NaN
3                           22                                   NaN
4                           22                                   NaN

   Start Date    End Date  Year         Hospital Type  \
0  07/01/2018  06/30/2019  2020  Acute Care Hospitals
1  07/01/2018  06/30/2019  2020  Acute Care Hospitals
2  07/01/2018  06/30/2019  2020  Acute Care Hospitals
3  07/01/2018  06/30/2019  2020  Acute Care Hospitals
4  07/01/2018  06/30/2019  2020  Acute Care Hospitals

                          Hospital Ownership Emergency Services  \
0  Government - Hospital District or Authority               Yes
1  Government - Hospital District or Authority               Yes
2  Government - Hospital District or Authority               Yes
3  Government - Hospital District or Authority               Yes
4  Government - Hospital District or Authority               Yes

  Meets criteria for promoting interoperability of EHRs  \
0                                                 Y
1                                                 Y
2                                                 Y
3                                                 Y
4                                                 Y

  Hospital overall rating Hospital overall rating footnote  \
0                       2                              NaN
```

```
1                       2                                NaN
2                       2                                NaN
3                       2                                NaN
4                       2                                NaN

  Mortality national comparison Mortality national comparison footnote  \
0    Below the national average                                    NaN
1    Below the national average                                    NaN
2    Below the national average                                    NaN
3    Below the national average                                    NaN
4    Below the national average                                    NaN

  Safety of care national comparison  \
0       Same as the national average
1       Same as the national average
2       Same as the national average
3       Same as the national average
4       Same as the national average

  Safety of care national comparison footnote Readmission national comparison  \
0                                        NaN      Below the national average
1                                        NaN      Below the national average
2                                        NaN      Below the national average
3                                        NaN      Below the national average
4                                        NaN      Below the national average

  Readmission national comparison footnote  \
0                                      NaN
1                                      NaN
2                                      NaN
3                                      NaN
4                                      NaN

  Patient experience national comparison  \
0             Below the national average
1             Below the national average
2             Below the national average
3             Below the national average
4             Below the national average

  Patient experience national comparison footnote  \
0                                             NaN
1                                             NaN
2                                             NaN
3                                             NaN
4                                             NaN

  Effectiveness of care national comparison  \
```

```
0              Same as the national average
1              Same as the national average
2              Same as the national average
3              Same as the national average
4              Same as the national average

  Effectiveness of care national comparison footnote  \
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4                                                 NaN

  Timeliness of care national comparison  \
0          Same as the national average
1          Same as the national average
2          Same as the national average
3          Same as the national average
4          Same as the national average

  Timeliness of care national comparison footnote  \
0                                              NaN
1                                              NaN
2                                              NaN
3                                              NaN
4                                              NaN

  Efficient use of medical imaging national comparison  \
0                      Same as the national average
1                      Same as the national average
2                      Same as the national average
3                      Same as the national average
4                      Same as the national average

  Efficient use of medical imaging national comparison footnote
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4                                                 NaN
**************************************************

Dataset has following Numerical Columns:
1- ZIP Code
2- Year
**************************************************
Dataset has following Categorical Columns:
```

```
1- Facility ID
2- Facility Name
3- Address
4- City
5- State
6- County Name
7- Phone Number
8- HCAHPS Measure ID
9- HCAHPS Question
10- HCAHPS Answer Description
11- Patient Survey Star Rating
12- Patient Survey Star Rating Footnote
13- HCAHPS Answer Percent
14- HCAHPS Answer Percent Footnote
15- HCAHPS Linear Mean Value
16- Number of Completed Surveys
17- Number of Completed Surveys Footnote
18- Survey Response Rate Percent
19- Survey Response Rate Percent Footnote
20- Start Date
21- End Date
22- Hospital Type
23- Hospital Ownership
24- Emergency Services
25- Meets criteria for promoting interoperability of EHRs
26- Hospital overall rating
27- Hospital overall rating footnote
28- Mortality national comparison
29- Mortality national comparison footnote
30- Safety of care national comparison
31- Safety of care national comparison footnote
32- Readmission national comparison
33- Readmission national comparison footnote
34- Patient experience national comparison
35- Patient experience national comparison footnote
36- Effectiveness of care national comparison
37- Effectiveness of care national comparison footnote
38- Timeliness of care national comparison
39- Timeliness of care national comparison footnote
40- Efficient use of medical imaging national comparison
41- Efficient use of medical imaging national comparison footnote
************************************************

Information of Data is as follows:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1653683 entries, 0 to 1653682
Data columns (total 43 columns):
 #   Column                                                    Non-Null
```

```
Count    Dtype
---  ------
--------------    -----
 0   Facility ID                                                1653683 non-
null  object
 1   Facility Name                                              1653683 non-
null  object
 2   Address                                                    1653683 non-
null  object
 3   City                                                       1653683 non-
null  object
 4   State                                                      1653683 non-
null  object
 5   ZIP Code                                                   1653683 non-
null  int64
 6   County Name                                                1651283 non-
null  object
 7   Phone Number                                               1653683 non-
null  object
 8   HCAHPS Measure ID                                          1653683 non-
null  object
 9   HCAHPS Question                                            1653683 non-
null  object
 10  HCAHPS Answer Description                                  1653683 non-
null  object
 11  Patient Survey Star Rating                                 1653683 non-
null  object
 12  Patient Survey Star Rating Footnote                        76826 non-
null    object
 13  HCAHPS Answer Percent                                      1653683 non-
null  object
 14  HCAHPS Answer Percent Footnote                             346726 non-
null   object
 15  HCAHPS Linear Mean Value                                   1653683 non-
null  object
 16  Number of Completed Surveys                                1653683 non-
null  object
 17  Number of Completed Surveys Footnote                       506289 non-
null   object
 18  Survey Response Rate Percent                               1653683 non-
null  object
 19  Survey Response Rate Percent Footnote                      506289 non-
null   object
 20  Start Date                                                 1653683 non-
null  object
 21  End Date                                                   1653683 non-
null  object
 22  Year                                                       1653683 non-
```

```
null  int64
 23  Hospital Type                                                1653683 non-
null  object
 24  Hospital Ownership                                           1653683 non-
null  object
 25  Emergency Services                                           1653683 non-
null  object
 26  Meets criteria for promoting interoperability of EHRs        1440651 non-
null  object
 27  Hospital overall rating                                      1653683 non-
null  object
 28  Hospital overall rating footnote                             441842 non-
null   object
 29  Mortality national comparison                                1653683 non-
null  object
 30  Mortality national comparison footnote                       475951 non-
null   object
 31  Safety of care national comparison                           1653683 non-
null  object
 32  Safety of care national comparison footnote                  744337 non-
null   object
 33  Readmission national comparison                              1653683 non-
null  object
 34  Readmission national comparison footnote                     336592 non-
null   object
 35  Patient experience national comparison                       1653683 non-
null  object
 36  Patient experience national comparison footnote              470302 non-
null   object
 37  Effectiveness of care national comparison                    1653683 non-
null  object
 38  Effectiveness of care national comparison footnote           410079 non-
null   object
 39  Timeliness of care national comparison                       1653683 non-
null  object
 40  Timeliness of care national comparison footnote              373723 non-
null   object
 41  Efficient use of medical imaging national comparison         1653683 non-
null  object
 42  Efficient use of medical imaging national comparison footnote  634119 non-
null   object
dtypes: int64(2), object(41)
memory usage: 542.5+ MB
None
**************************************************

Statistical Properties of Data:
```

	Facility ID	Facility Name	Address	City	State	\
count	1653683.0	1653683	1653683	1653683	1653683	
unique	8424.0	5769	5190	3007	56	
top	141350.0	MEMORIAL HOSPITAL	100 HOSPITAL DRIVE	CHICAGO	TX	
freq	346.0	5004	2422	10034	139785	
mean	NaN	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	NaN	

	ZIP Code	County Name	Phone Number	HCAHPS Measure ID	\
count	1.653683e+06	1651283	1653683	1653683	
unique	NaN	1627	9924	98	
top	NaN	LOS ANGELES	(216) 844-1000	H_COMP_1_A_P	
freq	NaN	27962	372	23928	
mean	5.403704e+04	NaN	NaN	NaN	
std	2.693196e+04	NaN	NaN	NaN	
min	6.030000e+02	NaN	NaN	NaN	
25%	3.301300e+04	NaN	NaN	NaN	
50%	5.541500e+04	NaN	NaN	NaN	
75%	7.609200e+04	NaN	NaN	NaN	
max	9.992900e+04	NaN	NaN	NaN	

	HCAHPS Question	\
count	1653683	
unique	100	
top	Patients who reported that their nurses "Alway…	
freq	23928	
mean	NaN	
std	NaN	
min	NaN	
25%	NaN	
50%	NaN	
75%	NaN	
max	NaN	

	HCAHPS Answer Description	Patient Survey Star Rating	\
count	1653683	1653683	
unique	101	7	
top	Nurses "always" communicated well	Not Applicable	
freq	23928	1380856	
mean	NaN	NaN	
std	NaN	NaN	
min	NaN	NaN	
25%	NaN	NaN	

```
50%                                    NaN                          NaN
75%                                    NaN                          NaN
max                                    NaN                          NaN

                     Patient Survey Star Rating Footnote  \
count                                                76826
unique                                                   7
top     15 - The number of cases/patients is too few t…
freq                                                 28385
mean                                                   NaN
std                                                    NaN
min                                                    NaN
25%                                                    NaN
50%                                                    NaN
75%                                                    NaN
max                                                    NaN

       HCAHPS Answer Percent HCAHPS Answer Percent Footnote  \
count                1653683                         346726
unique                   103                             36
top            Not Applicable                              1
freq                  521726                          39914
mean                     NaN                            NaN
std                      NaN                            NaN
min                      NaN                            NaN
25%                      NaN                            NaN
50%                      NaN                            NaN
75%                      NaN                            NaN
max                      NaN                            NaN

       HCAHPS Linear Mean Value Number of Completed Surveys  \
count                   1653683                     1653683
unique                       49                        3035
top              Not Applicable               Not Available
freq                    1404784                      234442
mean                        NaN                         NaN
std                         NaN                         NaN
min                         NaN                         NaN
25%                         NaN                         NaN
50%                         NaN                         NaN
75%                         NaN                         NaN
max                         NaN                         NaN

                     Number of Completed Surveys Footnote  \
count                                                506289
unique                                                   36
top     6 - Fewer than 100 patients completed the HCAH…
freq                                                 57345
```

```
mean                                                    NaN
std                                                     NaN
min                                                     NaN
25%                                                     NaN
50%                                                     NaN
75%                                                     NaN
max                                                     NaN

       Survey Response Rate Percent  \
count                       1653683
unique                           79
top                   Not Available
freq                        234442
mean                           NaN
std                            NaN
min                            NaN
25%                            NaN
50%                            NaN
75%                            NaN
max                            NaN

                  Survey Response Rate Percent Footnote  Start Date  \
count                                             506289     1653683
unique                                                36           5
top     6 - Fewer than 100 patients completed the HCAH…  07/01/2018
freq                                              57345      442587
mean                                                NaN         NaN
std                                                 NaN         NaN
min                                                 NaN         NaN
25%                                                 NaN         NaN
50%                                                 NaN         NaN
75%                                                 NaN         NaN
max                                                 NaN         NaN

          End Date          Year         Hospital Type  \
count      1653683  1.653683e+06               1653683
unique           5           NaN                     4
top     06/30/2019           NaN  Acute Care Hospitals
freq        442587           NaN               1149842
mean           NaN  2.018323e+03                   NaN
std            NaN  1.425935e+00                   NaN
min            NaN  2.016000e+03                   NaN
25%            NaN  2.017000e+03                   NaN
50%            NaN  2.019000e+03                   NaN
75%            NaN  2.020000e+03                   NaN
max            NaN  2.020000e+03                   NaN

                    Hospital Ownership Emergency Services  \
```

count	1653683	1653683
unique	11	2
top	Voluntary non-profit - Private	Yes
freq	724291	1534987
mean	NaN	NaN
std	NaN	NaN
min	NaN	NaN
25%	NaN	NaN
50%	NaN	NaN
75%	NaN	NaN
max	NaN	NaN

	Meets criteria for promoting interoperability of EHRs \
count	1440651
unique	2
top	Y
freq	1440486
mean	NaN
std	NaN
min	NaN
25%	NaN
50%	NaN
75%	NaN
max	NaN

	Hospital overall rating	Hospital overall rating footnote \
count	1653683	441842.0
unique	6	16.0
top	3	16.0
freq	472382	172515.0
mean	NaN	NaN
std	NaN	NaN
min	NaN	NaN
25%	NaN	NaN
50%	NaN	NaN
75%	NaN	NaN
max	NaN	NaN

	Mortality national comparison	Mortality national comparison footnote \
count	1653683	475951.0
unique	7	12.0
top	Same as the national average	5.0
freq	773391	135222.0
mean	NaN	NaN
std	NaN	NaN
min	NaN	NaN
25%	NaN	NaN
50%	NaN	NaN

```
75%                                NaN                                    NaN
max                                NaN                                    NaN

       Safety of care national comparison  \
count                              1653683
unique                                   7
top                          Not Available
freq                                743722
mean                                   NaN
std                                    NaN
min                                    NaN
25%                                    NaN
50%                                    NaN
75%                                    NaN
max                                    NaN

              Safety of care national comparison footnote  \
count                                                744337
unique                                                   12
top     Results are not available for this reporting p…
freq                                                 217470
mean                                                    NaN
std                                                     NaN
min                                                     NaN
25%                                                     NaN
50%                                                     NaN
75%                                                     NaN
max                                                     NaN

       Readmission national comparison  \
count                            1653683
unique                                 7
top          Above the national average
freq                              395717
mean                                 NaN
std                                  NaN
min                                  NaN
25%                                  NaN
50%                                  NaN
75%                                  NaN
max                                  NaN

        Readmission national comparison footnote  \
count                                    336592.0
unique                                       12.0
top                                           5.0
freq                                     116994.0
mean                                          NaN
```

```
std                                          NaN
min                                          NaN
25%                                          NaN
50%                                          NaN
75%                                          NaN
max                                          NaN

       Patient experience national comparison  \
count                                 1653683
unique                                      7
top                           Not Available
freq                                   470302
mean                                      NaN
std                                       NaN
min                                       NaN
25%                                       NaN
50%                                       NaN
75%                                       NaN
max                                       NaN

        Patient experience national comparison footnote  \
count                                            470302.0
unique                                               10.0
top                                                  16.0
freq                                             218271.0
mean                                                  NaN
std                                                   NaN
min                                                   NaN
25%                                                   NaN
50%                                                   NaN
75%                                                   NaN
max                                                   NaN

       Effectiveness of care national comparison  \
count                                    1653683
unique                                         7
top                 Same as the national average
freq                                      930543
mean                                         NaN
std                                          NaN
min                                          NaN
25%                                          NaN
50%                                          NaN
75%                                          NaN
max                                          NaN

        Effectiveness of care national comparison footnote  \
count                                             410079.0
```

```
unique                                                 10.0
top                                                     5.0
freq                                                158472.0
mean                                                     NaN
std                                                      NaN
min                                                      NaN
25%                                                      NaN
50%                                                      NaN
75%                                                      NaN
max                                                      NaN

       Timeliness of care national comparison  \
count                                 1653683
unique                                      7
top              Same as the national average
freq                                   430889
mean                                      NaN
std                                       NaN
min                                       NaN
25%                                       NaN
50%                                       NaN
75%                                       NaN
max                                       NaN

          Timeliness of care national comparison footnote  \
count                                                373723
unique                                                   10
top     Results are not available for this reporting p…
freq                                                 118415
mean                                                    NaN
std                                                     NaN
min                                                     NaN
25%                                                     NaN
50%                                                     NaN
75%                                                     NaN
max                                                     NaN

       Efficient use of medical imaging national comparison  \
count                                                1653683
unique                                                     7
top                                            Not Available
freq                                                  633690
mean                                                     NaN
std                                                      NaN
min                                                      NaN
25%                                                      NaN
50%                                                      NaN
75%                                                      NaN
```

```
max                                                    NaN

       Efficient use of medical imaging national comparison footnote
count                                                 634119
unique                                                    12
top     Results are not available for this reporting p…
freq                                                  202620
mean                                                     NaN
std                                                      NaN
min                                                      NaN
25%                                                      NaN
50%                                                      NaN
75%                                                      NaN
max                                                      NaN

**************************************************
```

4 3. Data Preprocessing

4.1 3.1 Handling Missing Values

[9]:
```
hospital_df.isna().sum()
```

[9]:
```
Facility ID                                0
Facility Name                              0
Address                                    0
City                                       0
State                                      0
ZIP Code                                   0
County Name                             2400
Phone Number                               0
HCAHPS Measure ID                          0
HCAHPS Question                            0
HCAHPS Answer Description                  0
Patient Survey Star Rating                 0
Patient Survey Star Rating Footnote   1576857
HCAHPS Answer Percent                      0
HCAHPS Answer Percent Footnote        1306957
HCAHPS Linear Mean Value                   0
Number of Completed Surveys                0
Number of Completed Surveys Footnote  1147394
Survey Response Rate Percent               0
Survey Response Rate Percent Footnote 1147394
Start Date                                 0
End Date                                   0
Year                                       0
Hospital Type                              0
```

```
Hospital Ownership                                                  0
Emergency Services                                                  0
Meets criteria for promoting interoperability of EHRs          213032
Hospital overall rating                                             0
Hospital overall rating footnote                              1211841
Mortality national comparison                                       0
Mortality national comparison footnote                        1177732
Safety of care national comparison                                  0
Safety of care national comparison footnote                    909346
Readmission national comparison                                     0
Readmission national comparison footnote                      1317091
Patient experience national comparison                              0
Patient experience national comparison footnote               1183381
Effectiveness of care national comparison                           0
Effectiveness of care national comparison footnote            1243604
Timeliness of care national comparison                              0
Timeliness of care national comparison footnote               1279960
Efficient use of medical imaging national comparison                0
Efficient use of medical imaging national comparison footnote 1019564
dtype: int64
```

[10]:
```
# Dropping the Columns with 80% Missing Values
hospital_df = hospital_df.drop(columns=['Patient Survey Star Rating Footnote',␣
 ↪'HCAHPS Answer Percent Footnote', 'Number of Completed Surveys Footnote',␣
 ↪'Survey Response Rate Percent Footnote', 'Meets criteria for promoting␣
 ↪interoperability of EHRs', 'Hospital overall rating footnote', 'Mortality␣
 ↪national comparison footnote', 'Safety of care national comparison␣
 ↪footnote', 'Readmission national comparison footnote', 'Patient experience␣
 ↪national comparison footnote', 'Effectiveness of care national comparison␣
 ↪footnote', 'Timeliness of care national comparison footnote', 'Efficient use␣
 ↪of medical imaging national comparison footnote'])
```

[11]:
```
hospital_df.isna().sum()
```

[11]:
```
Facility ID                                    0
Facility Name                                  0
Address                                        0
City                                           0
State                                          0
ZIP Code                                       0
County Name                                 2400
Phone Number                                   0
HCAHPS Measure ID                              0
HCAHPS Question                                0
HCAHPS Answer Description                      0
Patient Survey Star Rating                     0
HCAHPS Answer Percent                          0
```

```
HCAHPS Linear Mean Value                                  0
Number of Completed Surveys                               0
Survey Response Rate Percent                              0
Start Date                                                0
End Date                                                  0
Year                                                      0
Hospital Type                                             0
Hospital Ownership                                        0
Emergency Services                                        0
Hospital overall rating                                   0
Mortality national comparison                             0
Safety of care national comparison                        0
Readmission national comparison                           0
Patient experience national comparison                    0
Effectiveness of care national comparison                 0
Timeliness of care national comparison                    0
Efficient use of medical imaging national comparison      0
dtype: int64
```

Handling missing values in the Column "County Name"

[12]:
```
hospital_df[['City', 'State', 'County Name']].isna().sum()
```

[12]:
```
City              0
State             0
County Name    2400
dtype: int64
```

Imputing missing values in the Column "County Name" with respect to 'City' & 'State'

[11]:
```
# Create a dictionary to map City-State combinations to County Name
city_state_to_county = hospital_df.dropna(subset=['County Name']).
 ↪drop_duplicates(subset=['City', 'State']).set_index(['City',␣
 ↪'State'])['County Name'].to_dict()

# Function to impute missing County Names
def impute_county(row):
    if pd.isnull(row['County Name']):
        city_state_key = (row['City'], row['State'])
        return city_state_to_county.get(city_state_key)
    return row['County Name']

# Apply the function to impute missing County Names
hospital_df['County Name'] = hospital_df.apply(impute_county, axis=1)
```

[13]:
```
# Remove the remaining missing values in the County Name column
hospital_df = hospital_df.dropna(subset = ['County Name'])
```

4.2 3.2 Analyzing the Categorical Columns for "Not Available" / "Not Applicable" values

[14]:
```
hospital_df.columns
```

[14]:
```
Index(['Facility ID', 'Facility Name', 'Address', 'City', 'State', 'ZIP Code',
       'County Name', 'Phone Number', 'HCAHPS Measure ID', 'HCAHPS Question',
       'HCAHPS Answer Description', 'Patient Survey Star Rating',
       'HCAHPS Answer Percent', 'HCAHPS Linear Mean Value',
       'Number of Completed Surveys', 'Survey Response Rate Percent',
       'Start Date', 'End Date', 'Year', 'Hospital Type', 'Hospital Ownership',
       'Emergency Services', 'Hospital overall rating',
       'Mortality national comparison', 'Safety of care national comparison',
       'Readmission national comparison',
       'Patient experience national comparison',
       'Effectiveness of care national comparison',
       'Timeliness of care national comparison',
       'Efficient use of medical imaging national comparison'],
      dtype='object')
```

[15]:
```
hospital_df['Patient experience national comparison'].value_counts()
```

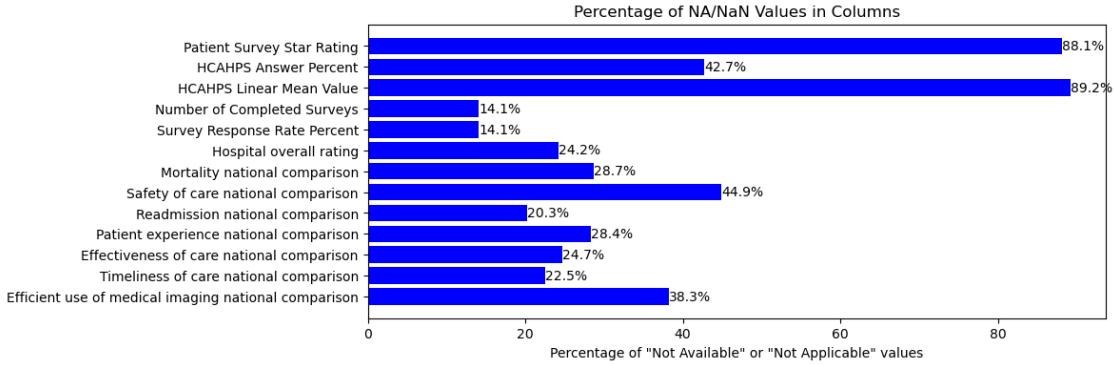
[15]:
```
Patient experience national comparison
Not Available                    468592
Above the national average       343943
Same as the national average     332122
Below the national average       316876
Above the National average        66880
Same as the National average      63360
Below the National average        59510
Name: count, dtype: int64
```

Columns having Not Availabe / Not Applicable values:

- 'Patient Survey Star Rating'
- 'HCAHPS Answer Percent'
- 'HCAHPS Linear Mean Value'
- 'Number of Completed Surveys'
- 'Survey Response Rate Percent'
- 'Hospital overall rating'
- 'Mortality national comparison'
- 'Safety of care national comparison'
- 'Readmission national comparison'
- 'Patient experience national comparison'
- 'Effectiveness of care national comparison'
- 'Timeliness of care national comparison'
- 'Efficient use of medical imaging national comparison'

[16]:
```
# List of columns with 'Not Available' / 'Not Applicable' values
columns_with_na = [
    'Patient Survey Star Rating',
    'HCAHPS Answer Percent',
    'HCAHPS Linear Mean Value',
    'Number of Completed Surveys',
    'Survey Response Rate Percent',
    'Hospital overall rating',
    'Mortality national comparison',
    'Safety of care national comparison',
    'Readmission national comparison',
    'Patient experience national comparison',
    'Effectiveness of care national comparison',
    'Timeliness of care national comparison',
    'Efficient use of medical imaging national comparison'
]

# Calculate percentage of 'Not Available' or 'Not Applicable' values in each␣
 ↪column
na_percentages = {}
for column in columns_with_na:
    na_count = hospital_df[hospital_df[column].isin(['Not Available', 'Not␣
 ↪Applicable'])][column].count()
    total_count = len(hospital_df[column])
    percentage = (na_count / total_count) * 100
    na_percentages[column] = percentage

# Display percentage of 'Not Available' or 'Not Applicable' values in each␣
 ↪column
for column, percentage in na_percentages.items():
    print(f"Column '{column}': {percentage:.2f}%")

# Create a bar chart to visualize the percentages
plt.figure(figsize=(10, 4))
bars = plt.barh(list(na_percentages.keys()), list(na_percentages.values()),␣
 ↪color='blue')
plt.xlabel('Percentage of "Not Available" or "Not Applicable" values')
plt.title('Percentage of NA/NaN Values in Columns')
plt.gca().invert_yaxis()  # Invert y-axis to display the columns from top to␣
 ↪bottom
# Annotate bars with their respective values
for bar in bars:
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2, f'{bar.
 ↪get_width():.1f}%',
             va='center', ha='left', color='black')
plt.show()
```

```
Column 'Patient Survey Star Rating': 88.14%
Column 'HCAHPS Answer Percent': 42.72%
Column 'HCAHPS Linear Mean Value': 89.18%
Column 'Number of Completed Surveys': 14.09%
Column 'Survey Response Rate Percent': 14.09%
Column 'Hospital overall rating': 24.16%
Column 'Mortality national comparison': 28.70%
Column 'Safety of care national comparison': 44.92%
Column 'Readmission national comparison': 20.27%
Column 'Patient experience national comparison': 28.38%
Column 'Effectiveness of care national comparison': 24.73%
Column 'Timeliness of care national comparison': 22.50%
Column 'Efficient use of medical imaging national comparison': 38.25%
```



As Patient experience national comparison is our target variable. We are dropping the values with "Not Available" from our target variable.

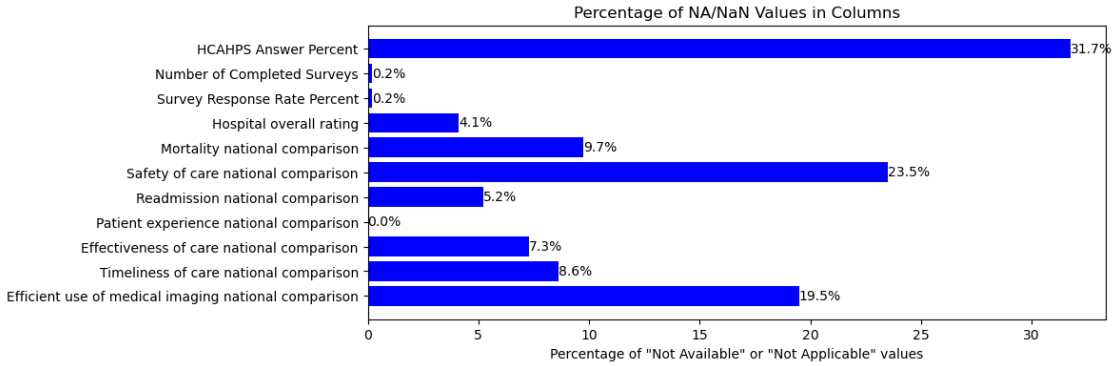
Also, as 'Patient Survey Star Rating', and HCAHPS Linear Mean Value has more than 60% Not Available values, we will be dropping the columns.

[17]:
```
hospital_df = hospital_df[hospital_df['Patient experience national comparison']␣
↪!= 'Not Available']
```

[18]:
```
# List of columns with 'Not Available' / 'Not Applicable' values
columns_with_na = [
    'HCAHPS Answer Percent',
    'Number of Completed Surveys',
    'Survey Response Rate Percent',
    'Hospital overall rating',
    'Mortality national comparison',
    'Safety of care national comparison',
    'Readmission national comparison',
    'Patient experience national comparison',
    'Effectiveness of care national comparison',
    'Timeliness of care national comparison',
```

```
    'Efficient use of medical imaging national comparison'
]

# Calculate percentage of 'Not Available' or 'Not Applicable' values in each␣
 ↪column
na_percentages = {}
for column in columns_with_na:
    na_count = hospital_df[hospital_df[column].isin(['Not Available', 'Not␣
 ↪Applicable'])][column].count()
    total_count = len(hospital_df[column])
    percentage = (na_count / total_count) * 100
    na_percentages[column] = percentage

# Display percentage of 'Not Available' or 'Not Applicable' values in each␣
 ↪column
for column, percentage in na_percentages.items():
    print(f"Column '{column}': {percentage:.2f}%")

# Create a bar chart to visualize the percentages
plt.figure(figsize=(10, 4))
bars = plt.barh(list(na_percentages.keys()), list(na_percentages.values()),␣
 ↪color='blue')
plt.xlabel('Percentage of "Not Available" or "Not Applicable" values')
plt.title('Percentage of NA/NaN Values in Columns')
plt.gca().invert_yaxis()  # Invert y-axis to display the columns from top to␣
 ↪bottom
# Annotate bars with their respective values
for bar in bars:
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2, f'{bar.
 ↪get_width():.1f}%',
             va='center', ha='left', color='black')
plt.show()
```

```
Column 'HCAHPS Answer Percent': 31.74%
Column 'Number of Completed Surveys': 0.20%
Column 'Survey Response Rate Percent': 0.20%
Column 'Hospital overall rating': 4.12%
Column 'Mortality national comparison': 9.74%
Column 'Safety of care national comparison': 23.49%
Column 'Readmission national comparison': 5.21%
Column 'Patient experience national comparison': 0.00%
Column 'Effectiveness of care national comparison': 7.29%
Column 'Timeliness of care national comparison': 8.63%
Column 'Efficient use of medical imaging national comparison': 19.49%
```



Dropping the rows having less than 10% values "Not Available"

[19]:
```
columns_to_drop_na = [
    'HCAHPS Answer Percent',
    'Number of Completed Surveys',
    'Survey Response Rate Percent',
    'Hospital overall rating',
    'Mortality national comparison',
    'Safety of care national comparison',
    'Readmission national comparison',
    'Patient experience national comparison',
    'Effectiveness of care national comparison',
    'Timeliness of care national comparison',
    'Efficient use of medical imaging national comparison'
]

threshold = 10

for column in columns_to_drop_na:
    na_percentage = (hospital_df[column].value_counts(normalize=True).get('Not␣
 ↪Available', 0) * 100)
    if na_percentage < threshold:
        hospital_df = hospital_df[hospital_df[column] != 'Not Available']
```

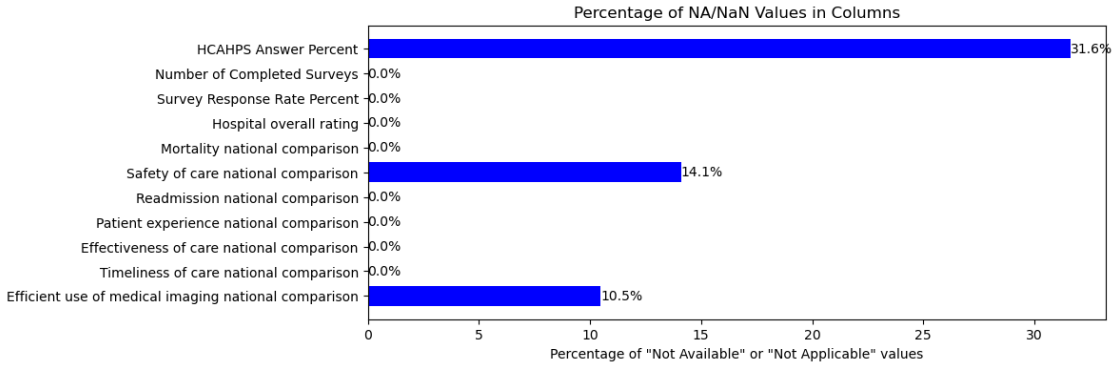
Imputing the value "Not Available" with Mode of columns "HCAHPS Answer Percent", "Safety of care national comparison" and "Efficient use of medical imaging national comparison"

[20]:
```
# List of columns with 'Not Available' / 'Not Applicable' values
columns_with_na = [
    'HCAHPS Answer Percent',
    'Number of Completed Surveys',
    'Survey Response Rate Percent',
    'Hospital overall rating',
    'Mortality national comparison',
```

```
    'Safety of care national comparison',
    'Readmission national comparison',
    'Patient experience national comparison',
    'Effectiveness of care national comparison',
    'Timeliness of care national comparison',
    'Efficient use of medical imaging national comparison'
]

# Calculate percentage of 'Not Available' or 'Not Applicable' values in each␣
 ↪column
na_percentages = {}
for column in columns_with_na:
    na_count = hospital_df[hospital_df[column].isin(['Not Available', 'Not␣
 ↪Applicable'])][column].count()
    total_count = len(hospital_df[column])
    percentage = (na_count / total_count) * 100
    na_percentages[column] = percentage

# Display percentage of 'Not Available' or 'Not Applicable' values in each␣
 ↪column
for column, percentage in na_percentages.items():
    print(f"Column '{column}': {percentage:.2f}%")

# Create a bar chart to visualize the percentages
plt.figure(figsize=(10, 4))
bars = plt.barh(list(na_percentages.keys()), list(na_percentages.values()),␣
 ↪color='blue')
plt.xlabel('Percentage of "Not Available" or "Not Applicable" values')
plt.title('Percentage of NA/NaN Values in Columns')
plt.gca().invert_yaxis()  # Invert y-axis to display the columns from top to␣
 ↪bottom
# Annotate bars with their respective values
for bar in bars:
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2, f'{bar.
 ↪get_width():.1f}%',
             va='center', ha='left', color='black')
plt.show()
```

```
Column 'HCAHPS Answer Percent': 31.62%
Column 'Number of Completed Surveys': 0.00%
Column 'Survey Response Rate Percent': 0.00%
Column 'Hospital overall rating': 0.00%
Column 'Mortality national comparison': 0.00%
Column 'Safety of care national comparison': 14.09%
Column 'Readmission national comparison': 0.00%
Column 'Patient experience national comparison': 0.00%
Column 'Effectiveness of care national comparison': 0.00%
```

```
Column 'Timeliness of care national comparison': 0.00%
Column 'Efficient use of medical imaging national comparison': 10.49%
```



[21]:
```
hospital_df['Safety of care national comparison'].value_counts()
```

[21]:
```
Safety of care national comparison
Above the national average      310117
Below the national average      239783
Same as the national average    174377
Not Available                   141595
Same as the National average     60555
Above the National average       42075
Below the National average       36080
Name: count, dtype: int64
```

[22]:
```
# Impute values in "HCAHPS Answer Percent Categories", "Safety of care national␣
 ↪comparison" and
# "Efficient use of medical imaging national comparison"

# Standardizing string formats
hospital_df['HCAHPS Answer Percent'] = hospital_df['HCAHPS Answer Percent'].
 ↪astype(str).str.strip().str.lower()

# Choosing the imputation value
mode_value = hospital_df['HCAHPS Answer Percent'][~hospital_df['HCAHPS Answer␣
 ↪Percent'].isin(['not applicable', 'not available'])].mode()[0]

# Imputing 'Not Available' and 'Not Applicable'
hospital_df['HCAHPS Answer Percent'] = hospital_df['HCAHPS Answer Percent'].
 ↪replace(['not applicable', 'not available'], mode_value)

# Standardizing string formats
hospital_df['Safety of care national comparison'] = hospital_df['Safety of care␣
 ↪national comparison'].astype(str).str.strip().str.lower()
```

```
# Choosing the imputation value
mode_value2 = hospital_df['Safety of care national␣
 ↪comparison'][~hospital_df['Safety of care national comparison'].isin(['not␣
 ↪applicable', 'not available'])].mode()[0]

# Imputing 'Not Available' and 'Not Applicable'
hospital_df['Safety of care national comparison'].replace(['not applicable',␣
 ↪'not available'], mode_value2, inplace=True)

# Standardizing string formats
hospital_df['Efficient use of medical imaging national comparison'] =␣
 ↪hospital_df['Efficient use of medical imaging national comparison'].
 ↪astype(str).str.strip().str.lower()

# Choosing the imputation value
mode_value3 = hospital_df['Efficient use of medical imaging national␣
 ↪comparison'][~hospital_df['Efficient use of medical imaging national␣
 ↪comparison'].isin(['not applicable', 'not available'])].mode()[0]

# Imputing 'Not Available' and 'Not Applicable'
hospital_df['Efficient use of medical imaging national comparison'].
 ↪replace(['not applicable', 'not available'], mode_value3, inplace=True)
```

5 4. Data Analysis

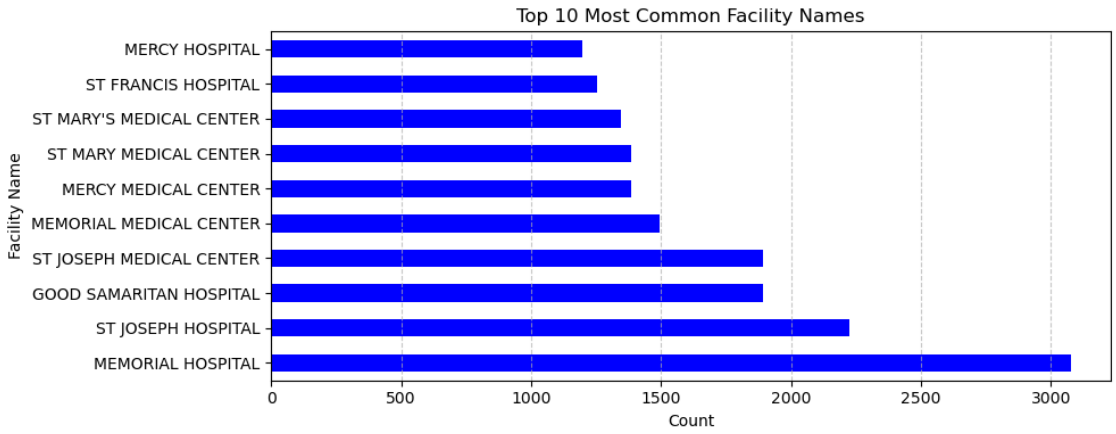
5.1 4.1 Facility Information

5.1.1 Question 1: What are the most common facility names, cities, or states?

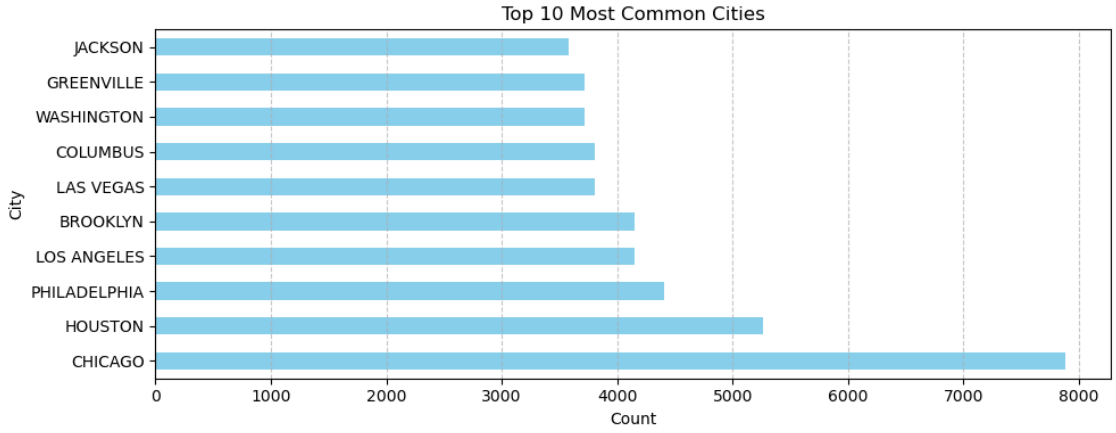
[23]:
```
# Convert Facility Name, City, and State columns to string to avoid issues
hospital_df['Facility Name'] = hospital_df['Facility Name'].astype(str)
hospital_df['City'] = hospital_df['City'].astype(str)
hospital_df['State'] = hospital_df['State'].astype(str)

# Finding the most common facility names
top_facility_names = hospital_df['Facility Name'].value_counts().head(10)

# Finding the most common cities
top_cities = hospital_df['City'].value_counts().head(10)

# Finding the most common states
top_states = hospital_df['State'].value_counts().head(10)
```

[24]:
```
# Visualization - Top Facility Names
plt.figure(figsize=(10, 4))
top_facility_names.plot(kind='barh', color='blue')  # Set color for the bars
```

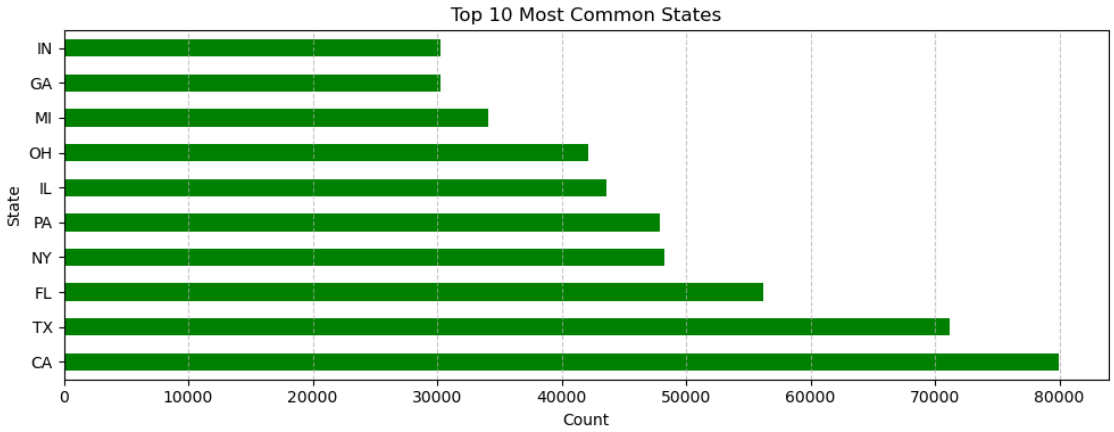
```
plt.title('Top 10 Most Common Facility Names')
plt.xlabel('Count')
plt.ylabel('Facility Name')
plt.grid(axis='x', linestyle='--', alpha=0.7)  # Add grid lines for better␣
 ↪readability
plt.tight_layout()
plt.show()
```



[25]:
```
# Visualization - Top Cities
plt.figure(figsize=(10, 4))
top_cities.plot(kind='barh', color='skyblue')
plt.title('Top 10 Most Common Cities')
plt.xlabel('Count')
plt.ylabel('City')
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



[26]:
```
# Visualization - Top States
plt.figure(figsize=(10, 4))
top_states.plot(kind='barh', color='green')
plt.title('Top 10 Most Common States')
plt.xlabel('Count')
plt.ylabel('State')
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



[27]:
```
# Summary
print("Top 10 Most Common Facility Names:\n", top_facility_names)
print("*"*50)
print()
print("\nTop 10 Most Common Cities:\n", top_cities)
print("*"*50)
print()
print("\nTop 10 Most Common States:\n", top_states)
```

```
Top 10 Most Common Facility Names:
 Facility Name
MEMORIAL HOSPITAL            3076
ST JOSEPH HOSPITAL           2224
GOOD SAMARITAN HOSPITAL      1890
ST JOSEPH MEDICAL CENTER     1890
MEMORIAL MEDICAL CENTER      1494
MERCY MEDICAL CENTER         1384
ST MARY MEDICAL CENTER       1384
ST MARY'S MEDICAL CENTER     1346
ST FRANCIS HOSPITAL          1253
```

```
MERCY HOSPITAL              1198
Name: count, dtype: int64
**********************************************


Top 10 Most Common Cities:
 City
CHICAGO         7882
HOUSTON         5262
PHILADELPHIA    4405
LOS ANGELES     4152
BROOKLYN        4152
LAS VEGAS       3806
COLUMBUS        3806
WASHINGTON      3713
GREENVILLE      3713
JACKSON         3577
Name: count, dtype: int64
**********************************************


Top 10 Most Common States:
 State
CA    79970
TX    71168
FL    56196
NY    48234
PA    47923
IL    43571
OH    42100
MI    34083
GA    30282
IN    30243
Name: count, dtype: int64
```

Facility Names: - Memorial Hospital appears as the most frequent facility name with 3076 occurrences. St Joseph Hospital and Good Samaritan Hospital follow with 2224 and 1890 occurrences, respectively. Several hospitals with similar names like St Joseph Medical Center, Memorial Medical Center, Mercy Medical Center, and St Mary Medical Center exhibit notable occurrences ranging from 1384 to 1890. Other hospitals such as St Mary's Medical Center, St Francis Hospital, and Mercy Hospital complete the top 10 list with occurrences ranging between 1198 to 1346.

Cities: - Chicago stands out significantly with 7882 occurrences, leading the list. Houston, Philadelphia, Los Angeles, and Brooklyn display a similar number of occurrences around 4152 to 5262. Las Vegas, Columbus, Washington, Greenville, and Jackson also appear in the top 10 list, each with 3577 to 3806 occurrences.

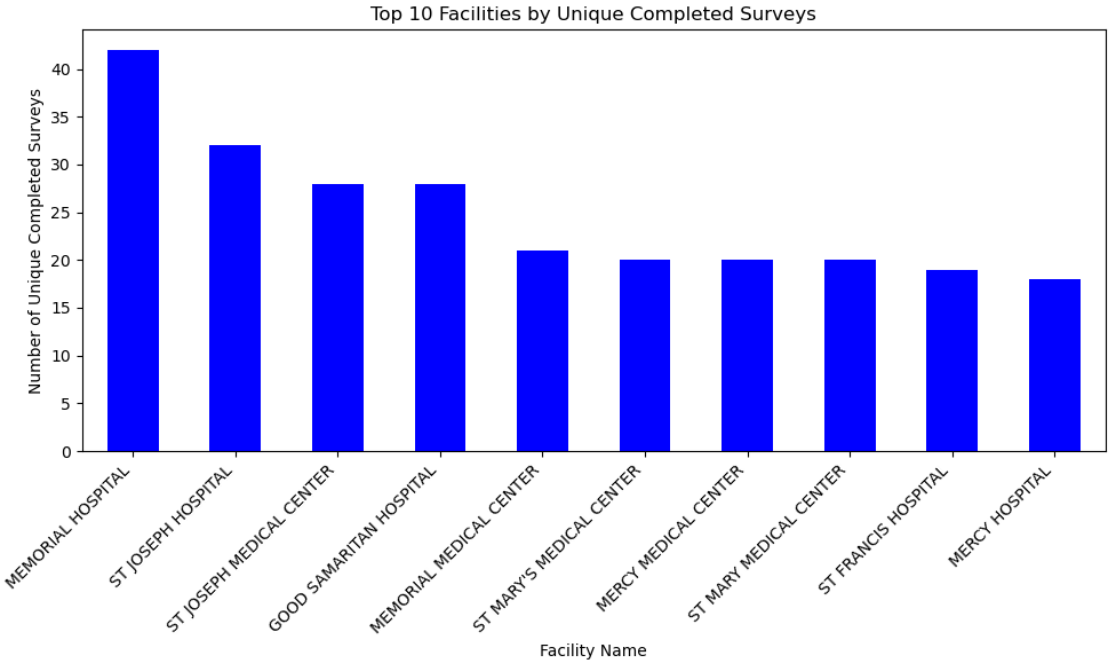
States: - California (CA) tops the list with a substantial count of 79970. Texas (TX) follows closely with 71168 occurrences. Florida (FL), New York (NY), and Pennsylvania (PA) exhibit significant

numbers, ranging from 47923 to 56196. Illinois (IL), Ohio (OH), Michigan (MI), Georgia (GA), and Indiana (IN) complete the top 10 list with occurrences between 30243 to 43571.

5.1.2 Question 2: How many completed surveys are available for each facility?

[28]:
```
# Convert 'Number of Completed Surveys' column to numeric
hospital_df['Number of Completed Surveys'] = pd.to_numeric(hospital_df['Number␣
 ↪of Completed Surveys'], errors='coerce')

# Grouping by Facility Name and aggregating unique completed surveys
unique_completed_surveys_per_facility = hospital_df.groupby('Facility␣
 ↪Name')['Number of Completed Surveys'].nunique().sort_values(ascending=False)

# Displaying summary
print(unique_completed_surveys_per_facility.head(10))  # Display top 10␣
 ↪facilities with most unique completed surveys

# Visualization - Bar chart
plt.figure(figsize=(10, 6))
unique_completed_surveys_per_facility.head(10).plot(kind='bar', color='blue')
plt.title('Top 10 Facilities by Unique Completed Surveys')
plt.xlabel('Facility Name')
plt.ylabel('Number of Unique Completed Surveys')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
Facility Name
MEMORIAL HOSPITAL           42
ST JOSEPH HOSPITAL          32
ST JOSEPH MEDICAL CENTER    28
GOOD SAMARITAN HOSPITAL     28
MEMORIAL MEDICAL CENTER     21
ST MARY'S MEDICAL CENTER    20
MERCY MEDICAL CENTER        20
ST MARY MEDICAL CENTER      20
ST FRANCIS HOSPITAL         19
MERCY HOSPITAL              18
Name: Number of Completed Surveys, dtype: int64
```



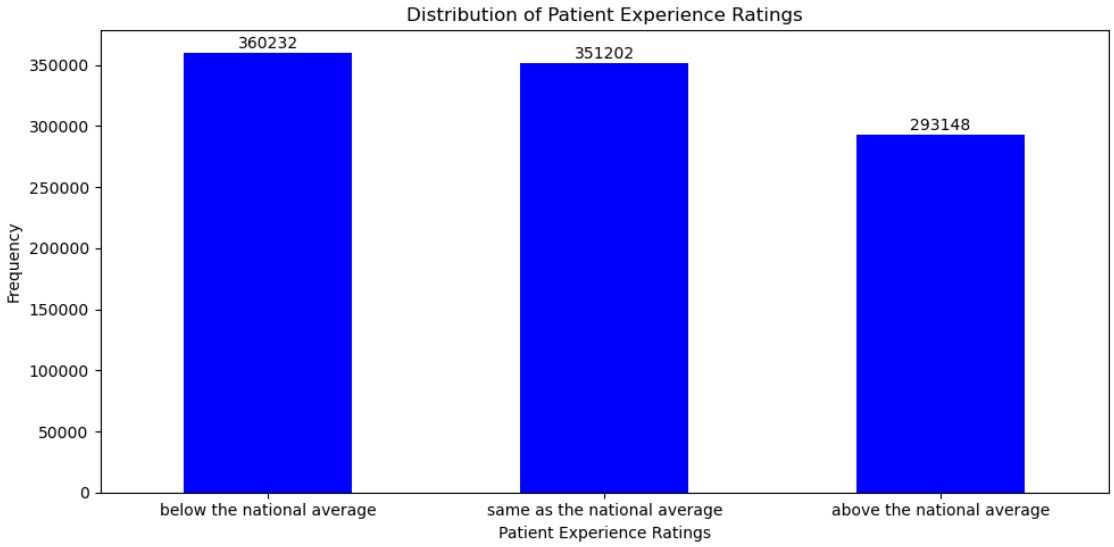
The top 10 facilities with the highest number of completed surveys are listed, revealing that Memorial Hospital holds the highest count with 42 completed surveys, followed closely by St Joseph Hospital with 32. Good Samaritan Hospital and St Joseph Medical Center share 28 completed surveys each, while several other facilities, including Memorial Medical Center, Mercy Medical Center, St Mary Medical Center, and St Mary's Medical Center, range between 20 to 21 completed surveys.

5.2 4.2 Patient Experience Ratings

5.2.1 Question 1: What is the overall distribution of patient experience ratings?

[29]:
```
# Lowercase the values in the column 'Patient experience national comparison'
hospital_df['Patient experience national comparison'] = hospital_df['Patient␣
 ↪experience national comparison'].str.lower()

# Plotting the distribution of patient experience ratings
plt.figure(figsize=(10, 5))
ax = hospital_df['Patient experience national comparison'].value_counts().
 ↪plot(kind='bar', color='blue')

plt.title('Distribution of Patient Experience Ratings')
plt.xlabel('Patient Experience Ratings')
plt.ylabel('Frequency')
plt.xticks(rotation=0)
plt.tight_layout()
```

```
# Adding annotations to the bars
for i in ax.patches:
    plt.text(i.get_x() + i.get_width() / 2, i.get_height() + 1000, str(i.
 ↪get_height()), ha='center', va='bottom')

plt.show()
```



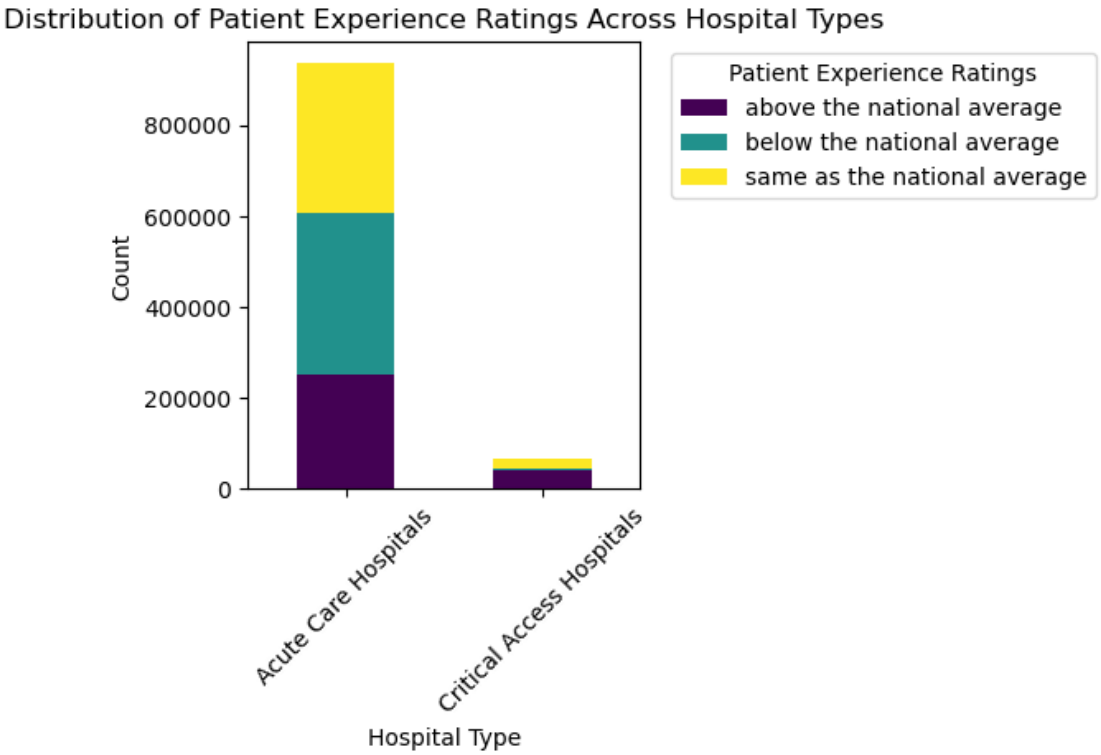
The distribution of patient experience ratings across different hospitals reveals an intriguing pattern. The majority of hospitals fall within the "Below the National Average," "Same as the National Average," and "Above the National Average" categories, with significant counts in each classification.

5.2.2 Question 2: How does the distribution of patient experience ratings vary across different hospital types?

[30]:
```
# Grouping the data by Hospital Type and Patient Experience Rating
rating_by_type = hospital_df.groupby(['Hospital Type', 'Patient experience␣
 ↪national comparison']).size().unstack()

# Plotting the distribution
plt.figure(figsize=(12, 6))
rating_by_type.plot(kind='bar', stacked=True, colormap='viridis')
plt.title('Distribution of Patient Experience Ratings Across Hospital Types')
plt.xlabel('Hospital Type')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Patient Experience Ratings', bbox_to_anchor=(1.05, 1),␣
 ↪loc='upper left')
```

```
plt.tight_layout()
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```



[40]:
```
# Displaying the results
print("Results - Distribution of Patient Experience Ratings Across Hospital␣
  ↪Types:")
display(rating_by_type)
```

```
Results - Distribution of Patient Experience Ratings Across Hospital Types:

Patient experience national comparison  above the national average  \
Hospital Type
Acute Care Hospitals                                        250810
Critical Access Hospitals                                    42338

Patient experience national comparison  below the national average  \
Hospital Type
Acute Care Hospitals                                        357512
Critical Access Hospitals                                     2720

Patient experience national comparison  same as the national average
Hospital Type
```

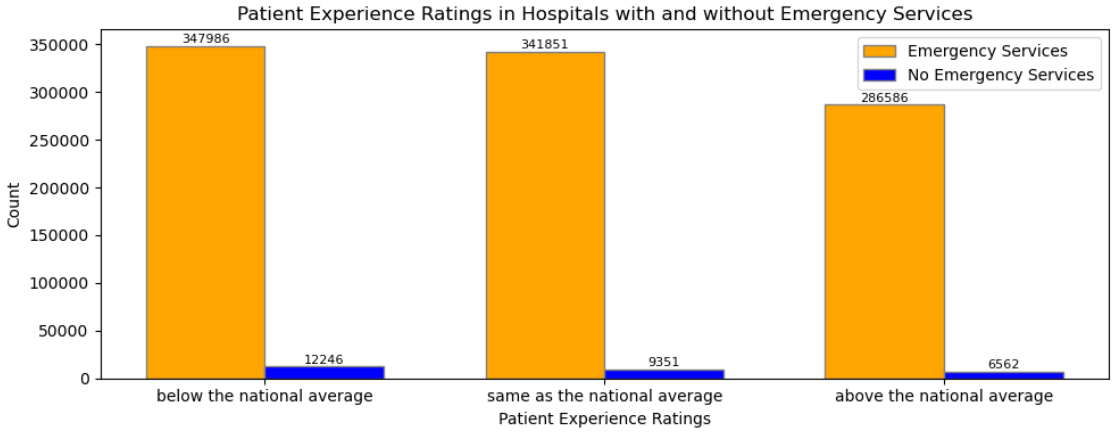
```
Acute Care Hospitals                                    328014
Critical Access Hospitals                                23188
```

The analysis reveals distinct variations in patient experience ratings across different hospital types. Among Acute Care Hospitals, the count for ratings above the national average stands at 250,810, surpassing those below the national average, which amount to 357,512. Conversely, Critical Access Hospitals show a considerably lower count of ratings above the national average, totaling 42,338, with a minor count of 2,720 below the national average. Moreover, Acute Care Hospitals display a considerable number of ratings aligning with the national average, reaching 328,224, whereas Critical Access Hospitals demonstrate a notably smaller count of ratings at 23,188 that meet the national average criteria. These findings underscore notable disparities in patient experience ratings between Acute Care Hospitals and Critical Access Hospitals.

5.2.3 Question 3: Among the hospitals providing emergency services and those that don't, how do patient experience ratings differ?

[32]:
```
# Filtering data for hospitals providing emergency services and those that don't
emergency_vs_nonemergency = hospital_df[['Emergency Services', 'Patient␣
 ↪experience national comparison']]
emergency_services =␣
 ↪emergency_vs_nonemergency[emergency_vs_nonemergency['Emergency Services'] ==␣
 ↪'Yes']
non_emergency_services =␣
 ↪emergency_vs_nonemergency[emergency_vs_nonemergency['Emergency Services'] ==␣
 ↪'No']

# Count occurrences of patient experience ratings for hospitals with emergency␣
 ↪services
emergency_services_counts = emergency_services['Patient experience national␣
 ↪comparison'].value_counts()

# Count occurrences of patient experience ratings for hospitals without␣
 ↪emergency services
non_emergency_services_counts = non_emergency_services['Patient experience␣
 ↪national comparison'].value_counts()

# Plotting the comparison between emergency and non-emergency hospitals
plt.figure(figsize=(10, 4))
barWidth = 0.35

r1 = np.arange(len(emergency_services_counts))
r2 = [x + barWidth for x in r1]

plt.bar(r1, emergency_services_counts, color='orange', width=barWidth,␣
 ↪edgecolor='grey', label='Emergency Services')
plt.bar(r2, non_emergency_services_counts, color='blue', width=barWidth,␣
 ↪edgecolor='grey', label='No Emergency Services')
```

```
plt.xlabel('Patient Experience Ratings')
plt.ylabel('Count')
plt.title('Patient Experience Ratings in Hospitals with and without Emergency␣
 ↪Services')
plt.xticks([r + barWidth / 2 for r in range(len(emergency_services_counts))],␣
 ↪emergency_services_counts.index)

# Adding figures on each bar
for i in range(len(emergency_services_counts)):
    plt.text(r1[i], emergency_services_counts[i] + 1000,␣
 ↪str(emergency_services_counts[i]), ha='center', va='bottom', color='black',␣
 ↪fontsize=8)
    plt.text(r2[i], non_emergency_services_counts[i] + 1000,␣
 ↪str(non_emergency_services_counts[i]), ha='center', va='bottom',␣
 ↪color='black', fontsize=8)

plt.legend()
plt.tight_layout()
plt.show()
```



[39]:
```
# Printing the results
print("Patient Experience Ratings for Hospitals with Emergency Services:\n",␣
 ↪emergency_services_counts)
print("\nPatient Experience Ratings for Hospitals without Emergency Services:
 ↪\n", non_emergency_services_counts)
```

```
Patient Experience Ratings for Hospitals with Emergency Services:
 Patient experience national comparison
below the national average      347986
same as the national average    341851
above the national average      286586
```

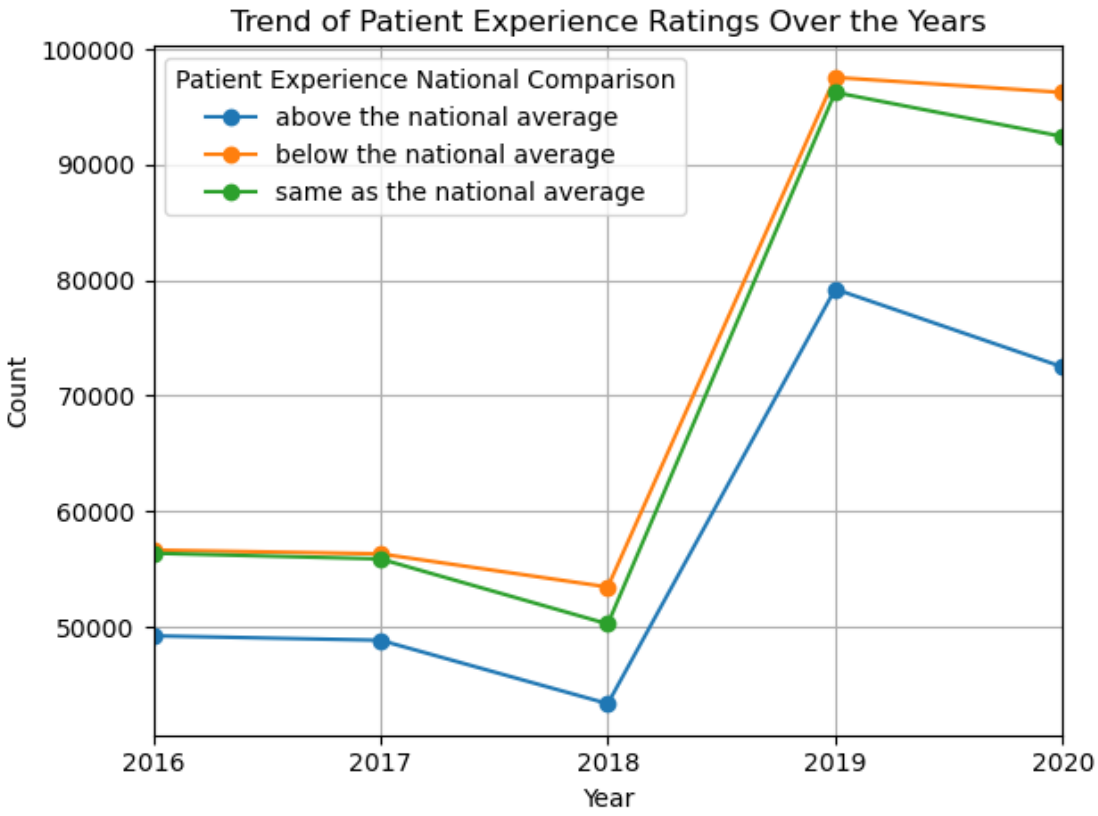
```
Name: count, dtype: int64

Patient Experience Ratings for Hospitals without Emergency Services:
 Patient experience national comparison
below the national average      12246
same as the national average     9351
above the national average       6562
Name: count, dtype: int64
```

The comparison of patient experience ratings between hospitals with and without emergency services reveals notable disparities. Hospitals providing emergency services exhibit substantially higher counts across all rating categories—below, same as, and above the national average—compared to those without such services. Specifically, hospitals with emergency services show significantly higher patient experience ratings below and same as the national average, with markedly elevated figures compared to hospitals lacking emergency services. Conversely, hospitals without emergency services showcase considerably lower counts across all rating categories, indicating a contrast in patient experiences favoring institutions with emergency services.

5.2.4 Question 3: What is the trend of patient experience ratings over the years? Are there any significant changes observed?

[37]:
```
# Convert 'Year' column to datetime format
hospital_df['Year'] = pd.to_datetime(hospital_df['Year'], format='%Y')

# Grouping by Year and Patient experience ratings to get counts
ratings_over_time = hospital_df.groupby(['Year', 'Patient experience national␣
 ↪comparison']).size().unstack()

# Plotting the trend of patient experience ratings over the years
plt.figure(figsize=(10, 4))
ratings_over_time.plot(kind='line', marker='o')
plt.title('Trend of Patient Experience Ratings Over the Years')
plt.xlabel('Year')
plt.ylabel('Count')
plt.legend(title='Patient Experience National Comparison')
plt.grid(True)
plt.tight_layout()
plt.show()

# Printing the results
display(ratings_over_time)
```

```
<Figure size 1000x400 with 0 Axes>
```



Year	Patient experience national comparison above the national average
2016-01-01	49225
2017-01-01	48840
2018-01-01	43350
2019-01-01	79236
2020-01-01	72497

Year	Patient experience national comparison below the national average
2016-01-01	56650
2017-01-01	56320
2018-01-01	53450
2019-01-01	97557
2020-01-01	96255

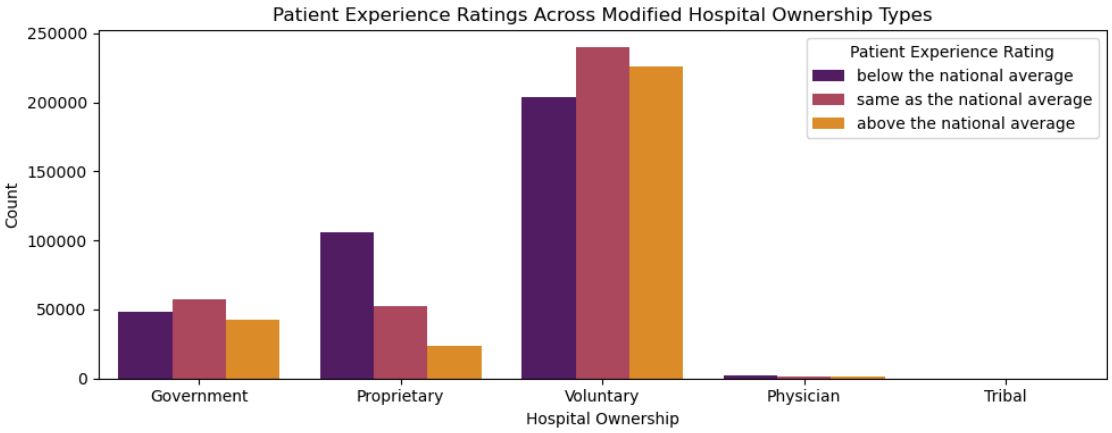
Year	Patient experience national comparison same as the national average
2016-01-01	56375
2017-01-01	55880
2018-01-01	50250

2019-01-01	96255
2020-01-01	92442

The results depict a comprehensive overview of patient experience ratings categorized as above, below, and the same as the national average across different years. The data highlights fluctuations and patterns in patient experiences from 2016 to 2020. Across these years, there are noticeable variations in the counts of ratings, indicating potential shifts in patient perceptions. Specifically, while there are fluctuations in the counts of ratings above and below the national average, the counts of ratings similar to the national average showcase a decreasing trend in recent years. This insight suggests a potential change in patient experiences, warranting a closer examination of factors influencing these variations across different years.

5.2.5 Question 4: Is there a notable difference in patient experience ratings among hospitals with different ownership types?

[41]:
```
# Combine 'Government' and 'Voluntary' categories
hospital_df['Hospital Ownership'] = hospital_df['Hospital Ownership'].replace({
    'Voluntary non-profit - Private': 'Voluntary',
    'Voluntary non-profit - Other': 'Voluntary',
    'Voluntary non-profit - Church': 'Voluntary',
    'Government - Hospital District or Authority': 'Government',
    'Government - Local': 'Government',
    'Government - State': 'Government',
    'Government - Federal': 'Government'
})

# Visualization of patient experience ratings among hospitals with modified␣
 ↪ownership types
plt.figure(figsize=(10, 4))
sns.countplot(data=hospital_df, x='Hospital Ownership', hue='Patient experience␣
 ↪national comparison', palette='inferno')
plt.title('Patient Experience Ratings Across Modified Hospital Ownership Types')
plt.xlabel('Hospital Ownership')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.legend(title='Patient Experience Rating')
plt.tight_layout()
plt.show()

# Printing the results
ownership_ratings = pd.crosstab(hospital_df['Hospital Ownership'],␣
 ↪hospital_df['Patient experience national comparison'])
display(ownership_ratings)
```



Patient experience national comparison	above the national average
Hospital Ownership	
Government	42780
Physician	1224
Proprietary	23486
Tribal	0
Voluntary	225658

Patient experience national comparison	below the national average
Hospital Ownership	
Government	48137
Physician	1949
Proprietary	106170
Tribal	0
Voluntary	203976

Patient experience national comparison	same as the national average
Hospital Ownership	
Government	57572
Physician	1131
Proprietary	52283
Tribal	110
Voluntary	240106

The distribution of patient experience ratings across the modified hospital ownership types demonstrates that, among the ownership categories, 'Voluntary' hospitals have the highest count of ratings above, below, and the same as the national average. 'Government' hospitals follow, with a sizable count of patient experience ratings across all three categories—above, below, and same as the national average. 'Proprietary' hospitals exhibit a notable count across all three categories as well, though relatively fewer compared to 'Government' and 'Voluntary' hospitals. 'Physician' and 'Tribal' categories show significantly lower counts of patient experience ratings across these three national comparison categories, with 'Tribal' hospitals having no ratings below or above the national average.

5.2.6 Question 5: Which facilities have the highest patient survey ratings?

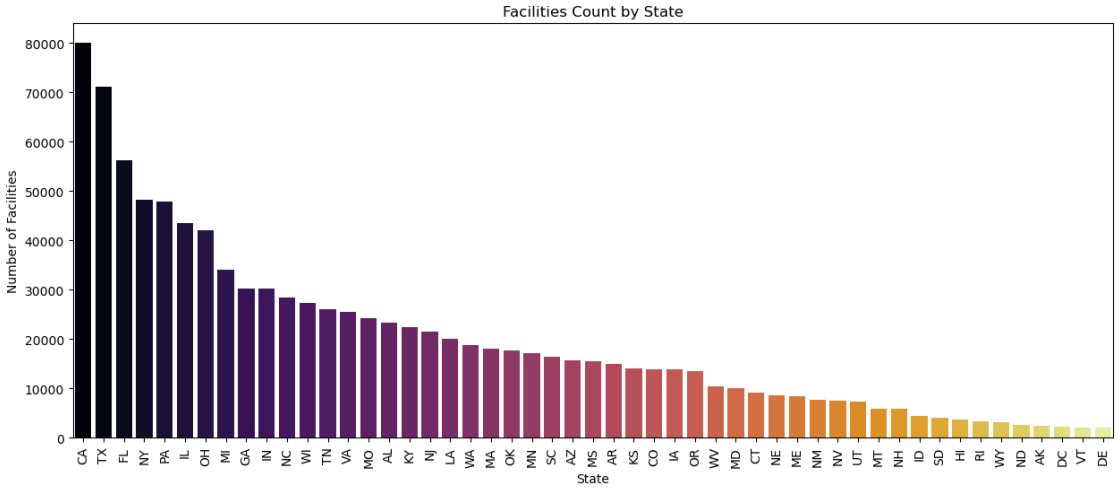
[42]:
```
# Filter the rows with numeric patient survey ratings
ratings = hospital_df['Patient Survey Star Rating'].str.isnumeric()

# Convert the filtered values to numeric (integer)
hospital_df['Patient Survey Star Rating'] = hospital_df['Patient Survey Star␣
 ↪Rating'][ratings].astype(int)

# Group by facility name and calculate the mean patient survey star rating
facility_ratings = hospital_df.groupby('Facility Name')['Patient Survey Star␣
 ↪Rating'].mean()

# Sort the facilities by the mean rating in descending order to get the highest␣
 ↪ratings first
highest_rated_facilities = facility_ratings.sort_values(ascending=False)

# Display the top facilities with the highest ratings
top_facilities = highest_rated_facilities.head(10)
print(top_facilities)
```

```
Facility Name
T J HEALTH COLUMBIA                          5.000000
OKLAHOMA HEART HOSPITAL, LLC                 4.947368
SAUK PRAIRIE HOSPITAL                        4.934783
CROSSING RIVERS HEALTH MEDICAL CENTER        4.869565
DOOR COUNTY MEDICAL CENTER                   4.863636
OKLAHOMA HEART HOSPITAL SOUTH, LLC           4.859649
COMMUNITY MEDICAL CENTER, INC                4.857143
MARINERS HOSPITAL                            4.847826
HEART HOSPITAL OF LAFAYETTE                  4.842105
BRODSTONE MEMORIAL HOSP                      4.818182
Name: Patient Survey Star Rating, dtype: float64
```

5.3 4.3 Geographic Distribution

5.3.1 Question 1: What is the geographic distribution of facilities by states?

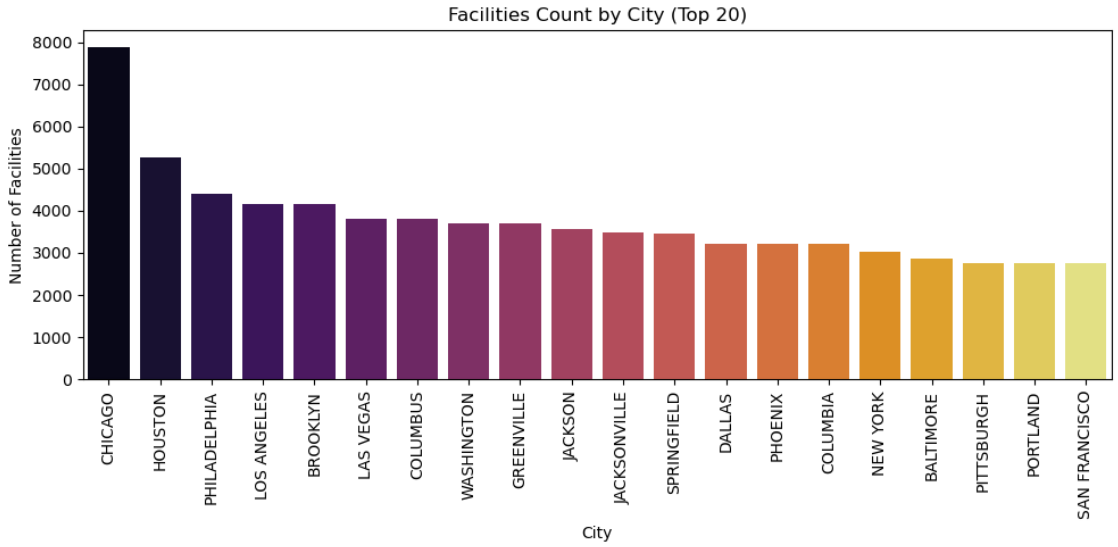
[43]:
```
# Plotting the count of facilities by State
plt.figure(figsize=(15, 6))
sns.countplot(data=hospital_df, x='State', order=hospital_df['State'].
 ↪value_counts().index, palette='inferno')
plt.title('Facilities Count by State')
plt.xlabel('State')
plt.ylabel('Number of Facilities')
plt.xticks(rotation=90)
plt.show()
```



The distribution of healthcare facilities across different states shows significant disparities in the dataset. California (CA) has the highest count of facilities with approximately 79,970, followed closely by Texas (TX) with 71,168 facilities and Florida (FL) with 56,196 facilities. Conversely, the District of Columbia (DC) and Vermont (VT) have the lowest count, each having only 2,329 and 2,076 facilities, respectively.

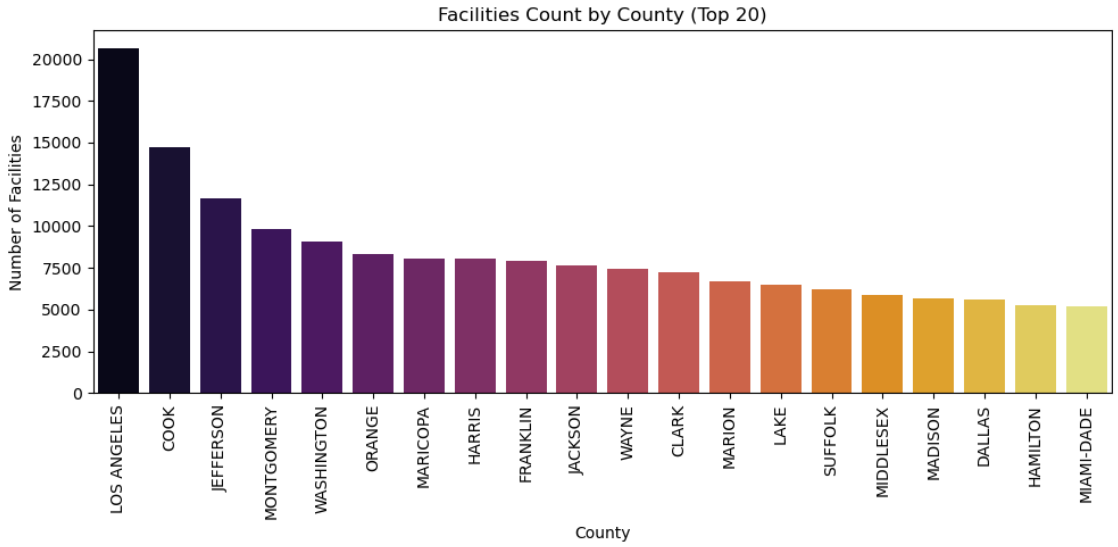
5.4 Question 2: What is the geographic distribution of facilities by cities?

[44]:
```
# Get the top 20 cities by facility count
top_cities = hospital_df['City'].value_counts().nlargest(20)

# Create a horizontal bar plot for the top cities
plt.figure(figsize=(10, 5))
sns.barplot(x=top_cities.index, y=top_cities.values, palette='inferno')
plt.title('Facilities Count by City (Top 20)')
plt.xlabel('City')
plt.ylabel('Number of Facilities')
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
```



5.4.1 Question 3: What is the geographic distribution of facilities by counties?

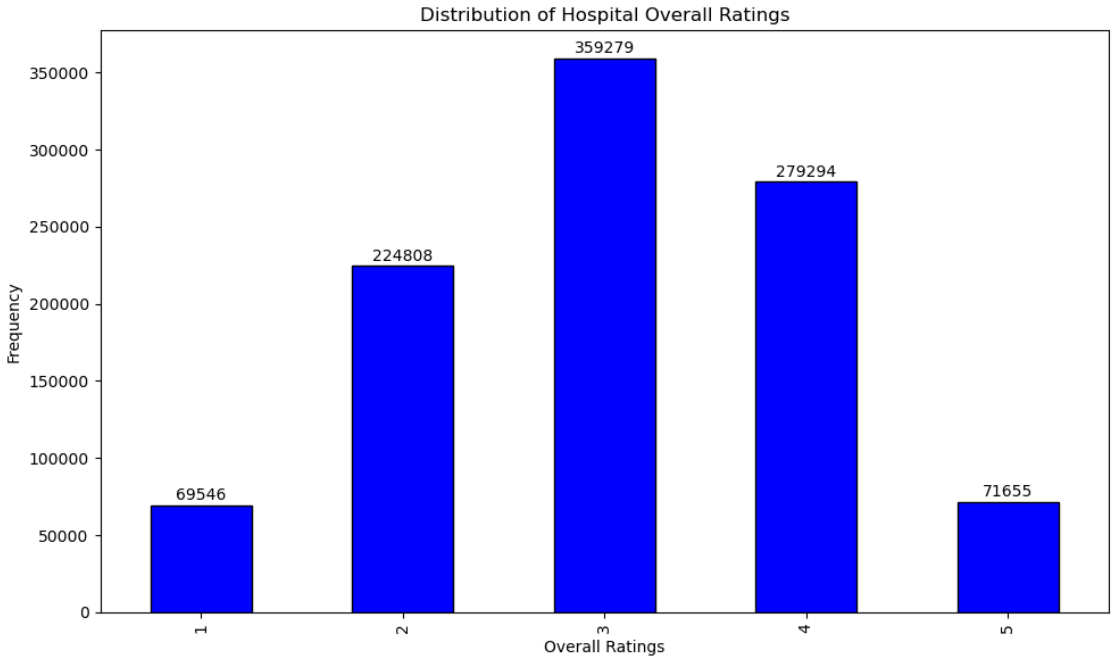
[45]:
```
# Get the top 20 Counties by facility count
top_county = hospital_df['County Name'].value_counts().nlargest(20)

# Create a horizontal bar plot for the top cities
plt.figure(figsize=(10, 5))
sns.barplot(x=top_county.index, y=top_county.values, palette='inferno')
plt.title('Facilities Count by County (Top 20)')
plt.xlabel('County')
plt.ylabel('Number of Facilities')
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
```



5.5 4.4 Hospital Information

5.5.1 Question 1: What is the distribution of Hospital overall ratings?

[46]:
```
# Plotting the distribution of Hospital overall ratings
plt.figure(figsize=(10, 6))
hospital_df['Hospital overall rating'].value_counts().sort_index().
 ↪plot(kind='bar', color='blue', edgecolor='black')
plt.title('Distribution of Hospital Overall Ratings')
plt.xlabel('Overall Ratings')
plt.ylabel('Frequency')

# Adding annotations
for i, value in enumerate(hospital_df['Hospital overall rating'].value_counts().
 ↪sort_index()):
    plt.text(i, value + 1000, str(value), ha='center', va='bottom')

# Show plot
plt.tight_layout()
plt.show()
```

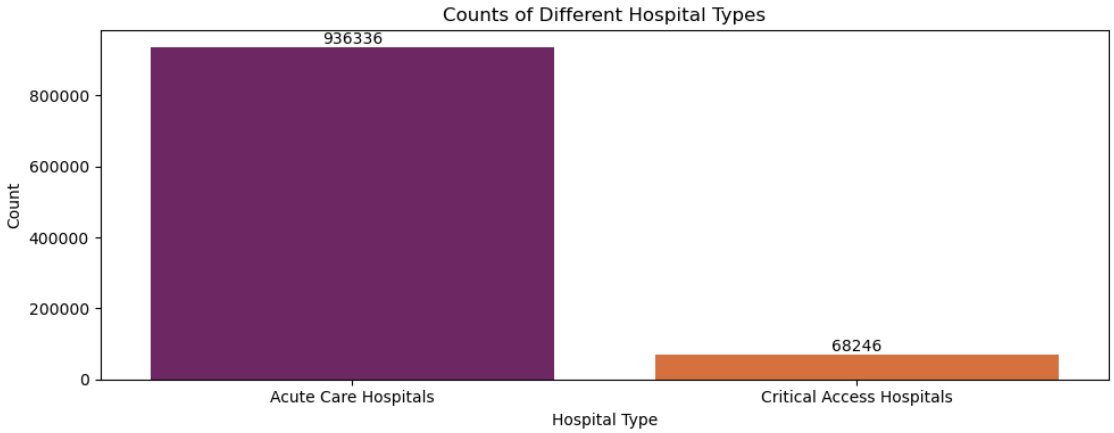


5.5.2 Question 2: What are the different types of hospitals present?

[47]:
```
# Get unique hospital types present in the dataset
hospital_types = hospital_df['Hospital Type'].unique()

# Display the unique hospital types
print(hospital_types)

# Count occurrences of each hospital type
hospital_type_counts = hospital_df['Hospital Type'].value_counts()

# Plotting the count of each hospital type
plt.figure(figsize=(10, 4))
ax = sns.barplot(x=hospital_type_counts.index, y=hospital_type_counts.values,␣
 ↪palette='inferno')
plt.title('Counts of Different Hospital Types')
plt.xlabel('Hospital Type')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.tight_layout()

# Adding figure labels on top of the bars
for i, count in enumerate(hospital_type_counts.values):
    ax.text(i, count + 50, str(count), ha='center', va='bottom', fontsize=10)
```

```
plt.show()
```

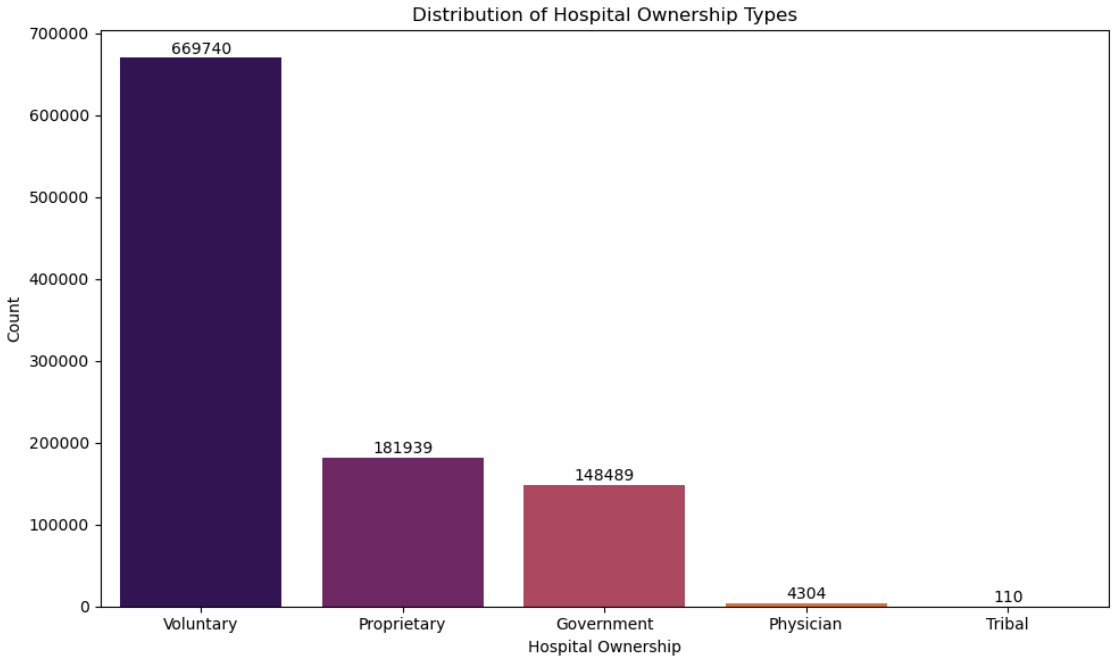
```
['Acute Care Hospitals' 'Critical Access Hospitals']
```



The dataset comprises two main types of hospitals: Acute Care Hospitals and Critical Access Hospitals. Acute Care Hospitals make up the majority, accounting for a significantly higher count of 936,546, while Critical Access Hospitals are notably fewer in number, constituting 68,246 instances in the dataset.

5.5.3 Question 3: What is the distribution of hospital ownership types?

[48]:
```
# Count occurrences of each hospital ownership type
ownership_counts = hospital_df['Hospital Ownership'].value_counts()

# Plotting the count of each hospital ownership type
plt.figure(figsize=(10, 6))
sns.barplot(x=ownership_counts.index, y=ownership_counts.values,␣
 ↪palette='inferno')
plt.title('Distribution of Hospital Ownership Types')
plt.xlabel('Hospital Ownership')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.tight_layout()

# Adding figure labels on top of the bars
for i, count in enumerate(ownership_counts.values):
    plt.text(i, count + 1000, str(count), ha='center', va='bottom', fontsize=10)

plt.show()
```



The distribution of hospital ownership types in the dataset portrays a varied landscape. "Voluntary" hospitals dominate the dataset, accounting for 669,950 instances, representing a substantial portion. Following this, "Proprietary" hospitals make up around 181,939 instances, establishing a significantly smaller presence. "Government" hospitals emerge as another noteworthy category, comprising approximately 148,489 instances. In contrast, "Physician" and "Tribal" hospital ownership types are less prevalent, with 4,304 and 110 instances, respectively. Overall, the dataset showcases a diverse range of hospital ownership types, with "Voluntary" hospitals occupying the most substantial proportion.

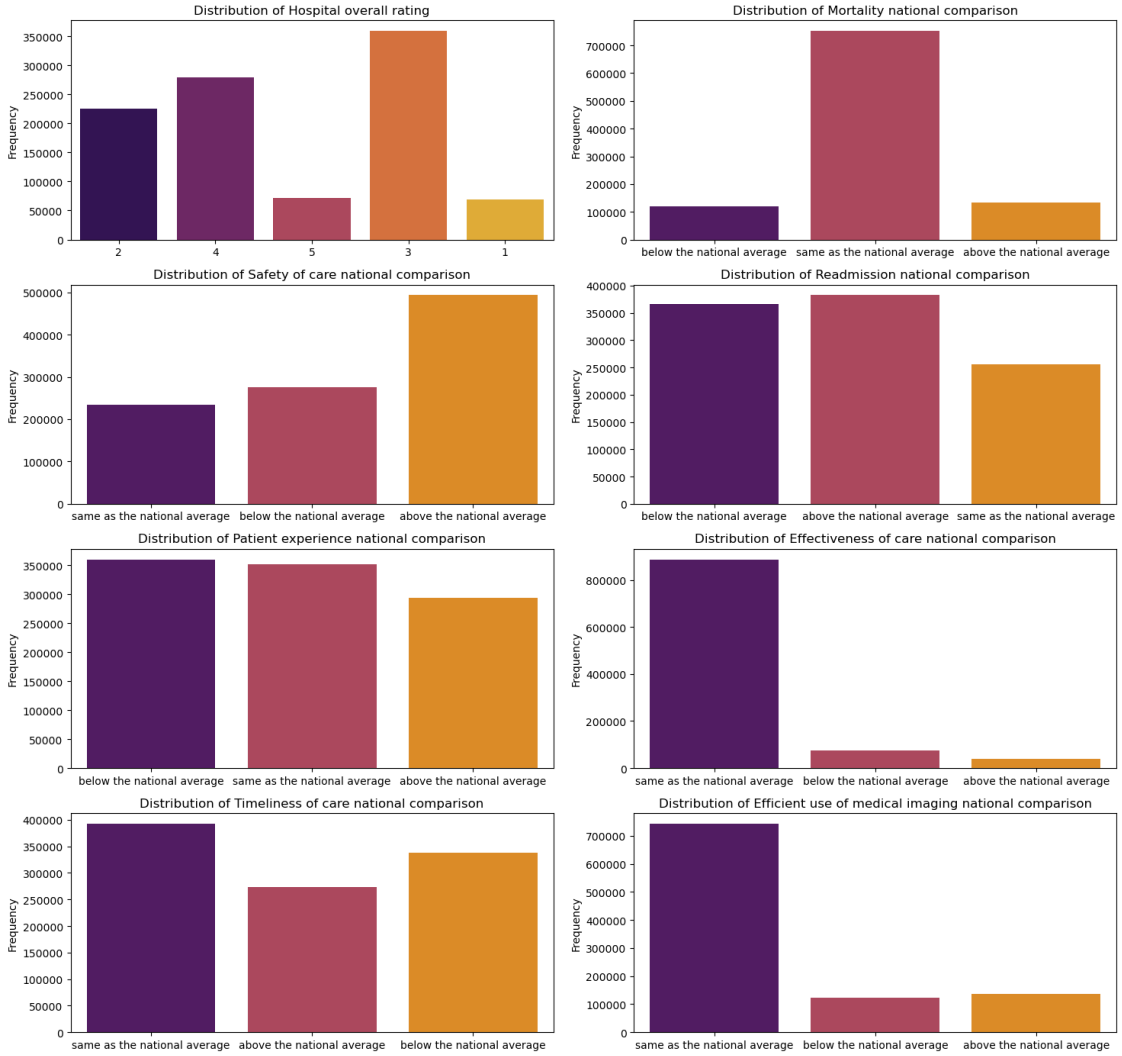
5.6 4.5 Rating Comparisons

5.6.1 Question 1: What are the different rating comparisons available for each category?

- Categories: Hospital Overall Rating, Mortality National Comparison, Safety of Care National Comparison, Readmission National Comparison, Patient Experience National Comparison, Effectiveness of Care National Comparison, Timeliness of Care National Comparison, Efficient Use of Medical Imaging National Comparison

[49]:
```
columns_to_visualize = [
    'Hospital overall rating',
    'Mortality national comparison',
    'Safety of care national comparison',
    'Readmission national comparison',
    'Patient experience national comparison',
    'Effectiveness of care national comparison',
```

```
    'Timeliness of care national comparison',
    'Efficient use of medical imaging national comparison'
]

plt.figure(figsize=(15, 14))

for i, col in enumerate(columns_to_visualize, 1):
    plt.subplot(4, 2, i)
    sns.countplot(data=hospital_df, x=hospital_df[col].str.lower(),␣
 ↪palette='inferno')
    plt.title(f'Distribution of {col}')
    plt.xlabel('')
    plt.ylabel('Frequency')
    plt.xticks(rotation=0)

plt.tight_layout()
plt.show()
```



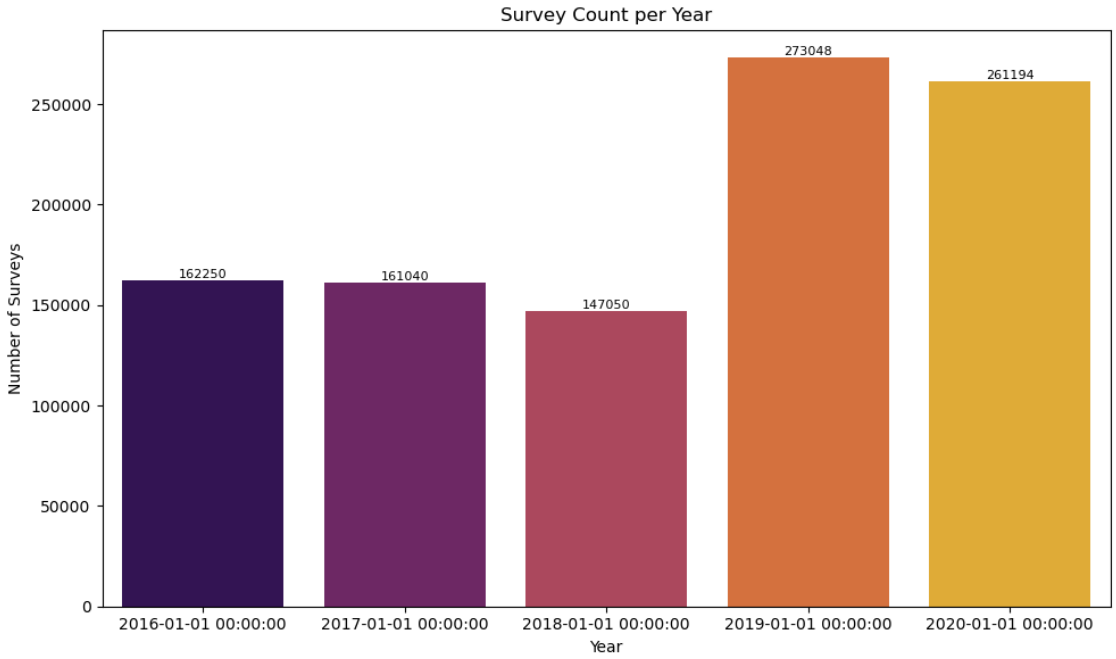
5.7 4.6 Time Analysis

5.7.1 Question 1: How many unique years are covered in the dataset and survey count in each year?

[50]:
```
unique_years = hospital_df['Year'].unique()

# Counting the number of unique years
num_unique_years = len(unique_years)

# Displaying the unique years and the count
print(f"Unique Years: {unique_years}")
print(f"Number of Unique Years: {num_unique_years}")
```

```
Unique Years: <DatetimeArray>
['2020-01-01 00:00:00', '2019-01-01 00:00:00', '2018-01-01 00:00:00',
 '2017-01-01 00:00:00', '2016-01-01 00:00:00']
Length: 5, dtype: datetime64[ns]
Number of Unique Years: 5
```

[51]:
```
# Count the frequency of surveys for each unique year
survey_count_per_year = hospital_df['Year'].value_counts().sort_index()

# Plotting the count of surveys per unique year
plt.figure(figsize=(10, 6))
ax = sns.barplot(x=survey_count_per_year.index, y=survey_count_per_year.values,␣
 ↪palette='inferno')
plt.title('Survey Count per Year')
plt.xlabel('Year')
plt.ylabel('Number of Surveys')

# Adding figure labels on top of the bars
for i, count in enumerate(survey_count_per_year.values):
    ax.text(i, count + 50, str(count), ha='center', va='bottom', fontsize=8)

plt.tight_layout()
plt.show()
```



The year 2019 amassed the highest number of surveys, totaling 273,048, closely followed by 2020 with 261,194 surveys. The years 2016 and 2017 recorded relatively similar counts, each comprising approximately 162,305 and 161,095 surveys, respectively, while 2018 accounted for 147,150 surveys. The distribution illustrates a gradual increase in survey participation from 2016 onwards, reaching its peak in 2019 before a relatively consistent count in the subsequent years, with a slight decline in 2020.

5.8 4.7 Top & Lowest Facilities

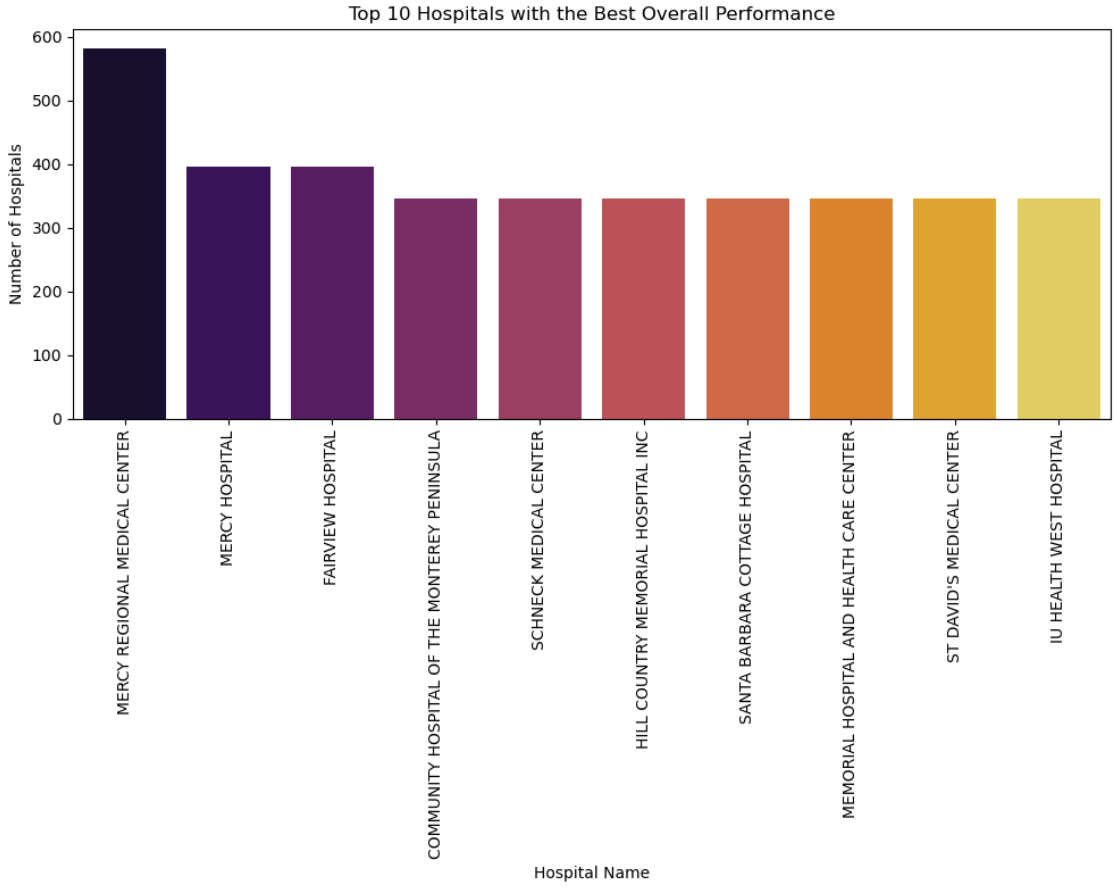
5.8.1 Question 1: Which facility has best performance overall? Top 20 Facilities.

[52]:
```
# Find the hospitals with the highest overall rating
top_hospitals_overall = hospital_df[hospital_df['Hospital overall rating'] ==␣
 ↪'5']

# Get unique hospitals with the best overall performance
unique_top_hospitals_overall = top_hospitals_overall['Facility Name'].unique()

# Display the total number of hospitals with the best overall performance
print(f"Total Number of Hospitals with Best Overall Performance:␣
 ↪{len(unique_top_hospitals_overall)}")

print("*"*60)
print("  ")

# Show the names of unique top hospitals with the best overall performance
```

```
print("Hospitals with the Best Overall Performance:(First 20)")
display(unique_top_hospitals_overall[:20])
```

```
Total Number of Hospitals with Best Overall Performance: 543
**********************************************************

Hospitals with the Best Overall Performance:(First 20)

array(['FAYETTE MEDICAL CENTER', 'SOUTH BALDWIN REGIONAL MEDICAL CENTER',
       'PRATTVILLE BAPTIST HOSPITAL', 'LAKELAND COMMUNITY HOSPITAL',
       'BAPTIST MEDICAL CENTER EAST', 'FLAGSTAFF MEDICAL CENTER',
       'BANNER BOSWELL MEDICAL CENTER', 'MAYO CLINIC HOSPITAL',
       'MERCY HOSPITAL NORTHWEST ARKANSAS',
       'BAPTIST HEALTH MEDICAL CENTER-STUTTGART',
       'ARKANSAS HEART HOSPITAL, LLC', 'PENINSULA MEDICAL CENTER',
       'ALTA BATES SUMMIT MEDICAL CENTER', 'SHARP MEMORIAL HOSPITAL',
       'SANTA MONICA - UCLA MED CTR & ORTHOPAEDIC HOSPITAL',
       'COMMUNITY HOSPITAL OF THE MONTEREY PENINSULA',
       'ST JUDE MEDICAL CENTER',
       'JOHN MUIR MEDICAL CENTER - WALNUT CREEK CAMPUS',
       'SEQUOIA HOSPITAL', 'SHARP CHULA VISTA MEDICAL CENTER'],
      dtype=object)
```

[53]:
```
# Get the top 10 hospitals with the best overall performance
top_10_hospitals_overall = top_hospitals_overall['Facility Name'].
 ↪value_counts().nlargest(10)

# Create a horizontal bar plot for the top 10 hospitals with the best overall␣
 ↪performance
plt.figure(figsize=(10, 8))
sns.barplot(x=top_10_hospitals_overall.index, y=top_10_hospitals_overall.
 ↪values, palette="inferno")
plt.title("Top 10 Hospitals with the Best Overall Performance")
plt.xlabel("Hospital Name")
plt.ylabel("Number of Hospitals")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



5.8.2 Question 2: Which facility has lowest performance overall?

[54]:
```
# Sort the facilities by the mean rating in ascending order to get the lowest␣
↪ratings first
lowest_rated_facilities = facility_ratings.sort_values(ascending=True)

# Display the facilities with the lowest ratings
bottom_facilities = lowest_rated_facilities.head(10)
print(bottom_facilities)
```

```
Facility Name
LOS ANGELES COMMUNITY HOSPITAL                      1.000000
ADVENTIST HEALTHCARE FORT WASHINGTON MEDICAL CTR    1.000000
UMD PRINCE GEORGE'S HOSPITAL CTR                    1.000000
UNIVERSITY OF MD PRINCE GEORGE'S HOSPITAL CTR       1.000000
SOUTHERN CALIFORNIA HOSPITAL AT HOLLYWOOD           1.000000
JACKSON PARK HOSPITAL                               1.017544
MEMORIAL HOSPITAL OF GARDENA                        1.017544
VICTOR VALLEY GLOBAL MEDICAL CENTER                 1.035088
```

```
GARDENS REGIONAL HOSPITAL AND MEDICAL CENTER        1.041667
UNIVERSITY OF MD LAUREL REGIONAL HOSPITAL           1.045455
Name: Patient Survey Star Rating, dtype: float64
```

6 5. Feature Engineering

6.1 5.1 Selecting Columns

Patient Survey Star Rating is having more than 80% missing values so we're dropping it

[55]:
```
hospital_df = hospital_df.drop(columns = 'Patient Survey Star Rating', axis = 1)
```

Removing the missing values from 'Number of Completed Surveys' Column

[56]:
```
hospital_df.dropna(subset=['Number of Completed Surveys'], inplace=True)
```

Selecting the columns which will be used in the development of ML model

[57]:
```
hospital_df = hospital_df.drop(columns = ['Facility ID', 'Facility Name',␣
 ↪'Address','State', 'City', 'ZIP Code',
           'County Name', 'Phone Number', 'HCAHPS Measure ID', 'HCAHPS Answer␣
 ↪Percent', 'HCAHPS Linear Mean Value',
           'Start Date', 'End Date', 'Number of Completed Surveys', 'Survey␣
 ↪Response Rate Percent'])
```

Making Categories more subtle

[58]:
```
value_dict = {'Voluntary non-profit - Church': 'Voluntary non-profit',␣
 ↪'Voluntary non-profit - Private': 'Voluntary non-profit', 'Voluntary␣
 ↪non-profit - Other': 'Voluntary non-profit',
   'Government - Hospital District or Authority': 'Government', 'Government -␣
 ↪Local': 'Government', 'Government - State': 'Government', 'Government -␣
 ↪Federal': 'Government'}
# Replace values in column
hospital_df['Hospital Ownership'] = hospital_df['Hospital Ownership'].
 ↪replace(value_dict)
```

6.2 5.2 Encoding the Categorical Values

[59]:
```
en_hosp_df = hospital_df.copy()
```

6.2.1 5.2.1 Separating Features X and Target Variable Y

[60]:
```
# Separate features (X) and target variable (y)
X = en_hosp_df.drop(columns = ['Patient experience national comparison'])
y = en_hosp_df['Patient experience national comparison']
```

6.2.2 5.2.2 Categorizing columns based on data

[61]:
```
# Categorizing columns based on data type
categorical_cols = ['HCAHPS Question', 'HCAHPS Answer Description', 'Hospital␣
 ↪Type', 'Hospital Ownership',
                    'Emergency Services',
                    'Mortality national comparison', 'Safety of care national␣
 ↪comparison',
                    'Readmission national comparison', 'Effectiveness of care␣
 ↪national comparison',
                    'Timeliness of care national comparison', 'Efficient use of␣
 ↪medical imaging national comparison']

# Numeric columns excluding Year and Hospital overall rating
numerical_cols = [col for col in X.columns if col not in categorical_cols +␣
 ↪['Hospital overall rating']]
```

6.3 5.3 ML Classifiers Implementation

6.3.1 5.3.1 Spliting the data and applying transformation

[62]:
```
# Custom Encoder for Label Encoding
class MultiColumnLabelEncoder(BaseEstimator, TransformerMixin):
    def __init__(self, columns=None):
        self.columns = columns  # list of columns to encode

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        output = X.copy()
        if self.columns:
            for col in self.columns:
                output[col] = LabelEncoder().fit_transform(output[col])
        else:
            for colname, col in output.iteritems():
                output[colname] = LabelEncoder().fit_transform(col)
        return output

# Create preprocessor using ColumnTransformer
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_cols),
        ('cat', MultiColumnLabelEncoder(columns=categorical_cols),␣
 ↪categorical_cols)
    ], remainder='passthrough')
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Apply transformation to the training and test sets
X_train_preprocessed = preprocessor.fit_transform(X_train)
X_test_preprocessed = preprocessor.transform(X_test)

print("Shape of X_train_preprocessed: ", X_train_preprocessed.shape)
print("Shape of X_test_preprocessed: ", X_test_preprocessed.shape)
print("Shape of y_train: ", y_train.shape)
print("Shape of y_test: ", y_test.shape)

X_train = X_train_preprocessed
X_test = X_test_preprocessed
```

```
Shape of X_train_preprocessed:  (803621, 13)
Shape of X_test_preprocessed:  (200906, 13)
Shape of y_train:  (803621,)
Shape of y_test:  (200906,)
```

6.3.2 5.3.2 Classifier Implementation

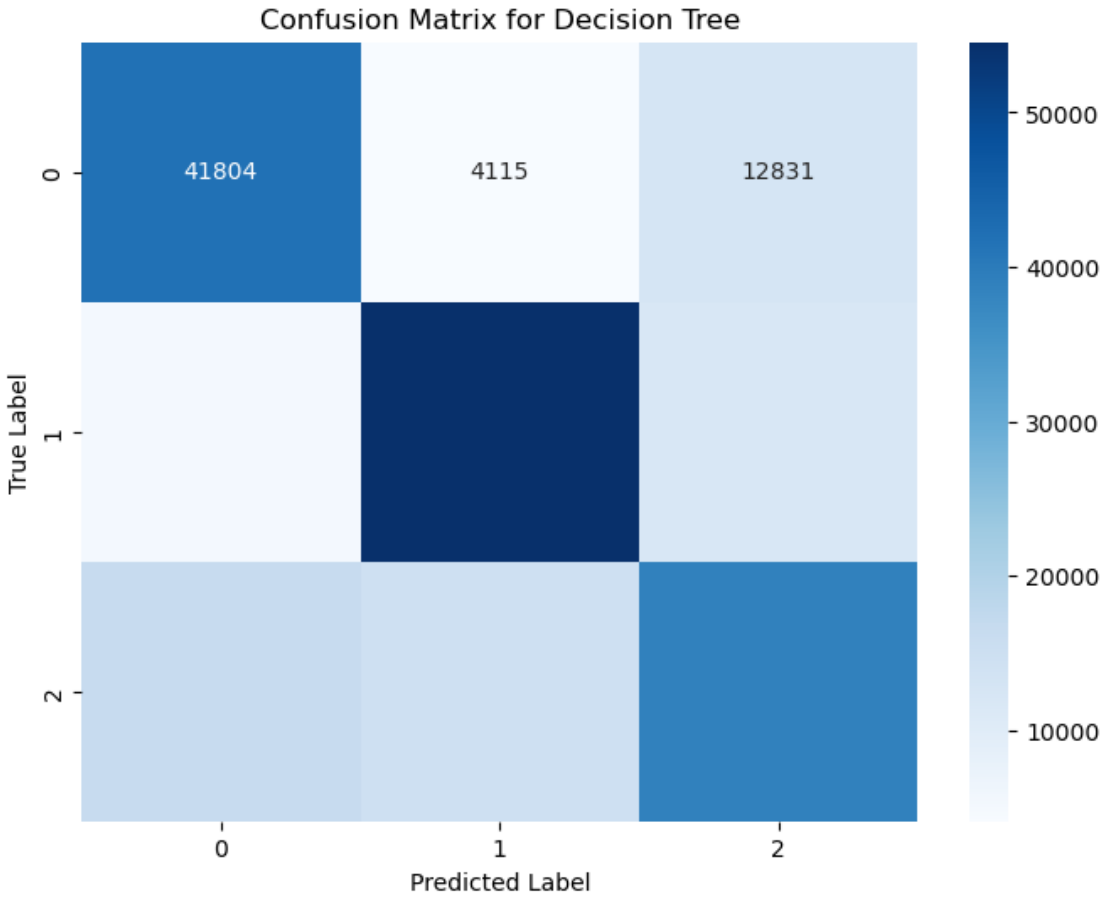
[63]:
```
# Importing Evaluation matrces
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

# check the performance on diffrent regressor
models = []
models.append(('Decision Tree', DecisionTreeClassifier()))
models.append(('LogisitcRegression', LogisticRegression()))

# metrices to store performance
acc = []
pre = []
f1 = []
con = []
rec = []


import time
i = 0
for name,model in models:
```

```
    i = i+1
    start_time = time.time()

    # Fitting model to the Training set
    clf = model
    clf.fit(X_train, y_train)

    # predict values
    y_pred = clf.predict(X_test)

    # Accuracy
    accuracy = accuracy_score(y_test, y_pred)
    acc.append(accuracy)
    # Precision
    precision = precision_score(y_test, y_pred, average=None)
    pre.append(precision)
    # Recall
    recall = recall_score(y_test, y_pred, average=None)
    rec.append(recall)
    # F1 Score
    f1_sco = f1_score(y_test, y_pred, average=None)
    f1.append(f1_sco)
    # Confusion Matrix
    confusion_mat = confusion_matrix(y_test, y_pred)
    con.append(confusion_mat)
    # Report
    report = classification_report(y_test, y_pred)


    print("+","="*100,"+")
    print('\033[1m' + f"\t\t\t{i}-For {name} The Performance result is: " +␣
↪'\033[0m')
    print("+","="*100,"+")
    print('Accuracy : ', accuracy)
    print("-"*50)
    print('F1 : ', f1_sco)
    print("-"*50)
    print('Reacll : ', recall)
    print("-"*50)
    print('Precision : ', precision)
    print("-"*50)
    print('Confusion Matrix....\n', confusion_mat)
    print("-"*50)
    print('Classification Report....\n', report)
    print("-"*50)
    print('Plotting Confusion Matrix...\n')
    # Plotting the confusion matrix using seaborn
```

```
    plt.figure(figsize=(8, 6))
    sns.heatmap(confusion_mat, annot=True, fmt='d', cmap='Blues')
    plt.title(f'Confusion Matrix for {name}')
    plt.ylabel('True Label')
    plt.xlabel('Predicted Label')
    plt.show()



  ␣
 ↪print("\t\t\t\t\t\t\t----------------------------------------------------------------")
    print(f"\t\t\t\t\t\t\t Time for detection ({name}) : {round((time.time() -␣
 ↪start_time), 3)} seconds...")
  ␣
 ↪print("\t\t\t\t\t\t\t----------------------------------------------------------------")
    print()

pd.DataFrame({"Model": dict(models).keys(), "Accuracy": acc, "Precision": pre,␣
 ↪"Recall": rec, "F1_Score": f1, "Confusion Matrix": con})
```

```
+ ================================================================================
====================== +
                    1-For Decision Tree The Performance result is:
+ ================================================================================
====================== +
Accuracy :  0.6731406727524315
--------------------------------------------------
F1 :  [0.68319469 0.75112898 0.57953343]
--------------------------------------------------
Reacll :  [0.71155745 0.75914562 0.55310068]
--------------------------------------------------
Precision :  [0.65700635 0.74327988 0.60861942]
--------------------------------------------------
Confusion Matrix…
 [[41804  4115 12831]
 [ 5139 54556 12170]
 [16685 14728 38878]]
--------------------------------------------------
Classification Report…
                              precision    recall  f1-score   support

  above the national average       0.66      0.71      0.68     58750
  below the national average       0.74      0.76      0.75     71865
same as the national average       0.61      0.55      0.58     70291

                    accuracy                           0.67    200906
                   macro avg       0.67      0.67      0.67    200906
```

```
                weighted avg       0.67      0.67      0.67     200906

---------------------------------------------------
Plotting Confusion Matrix…
```

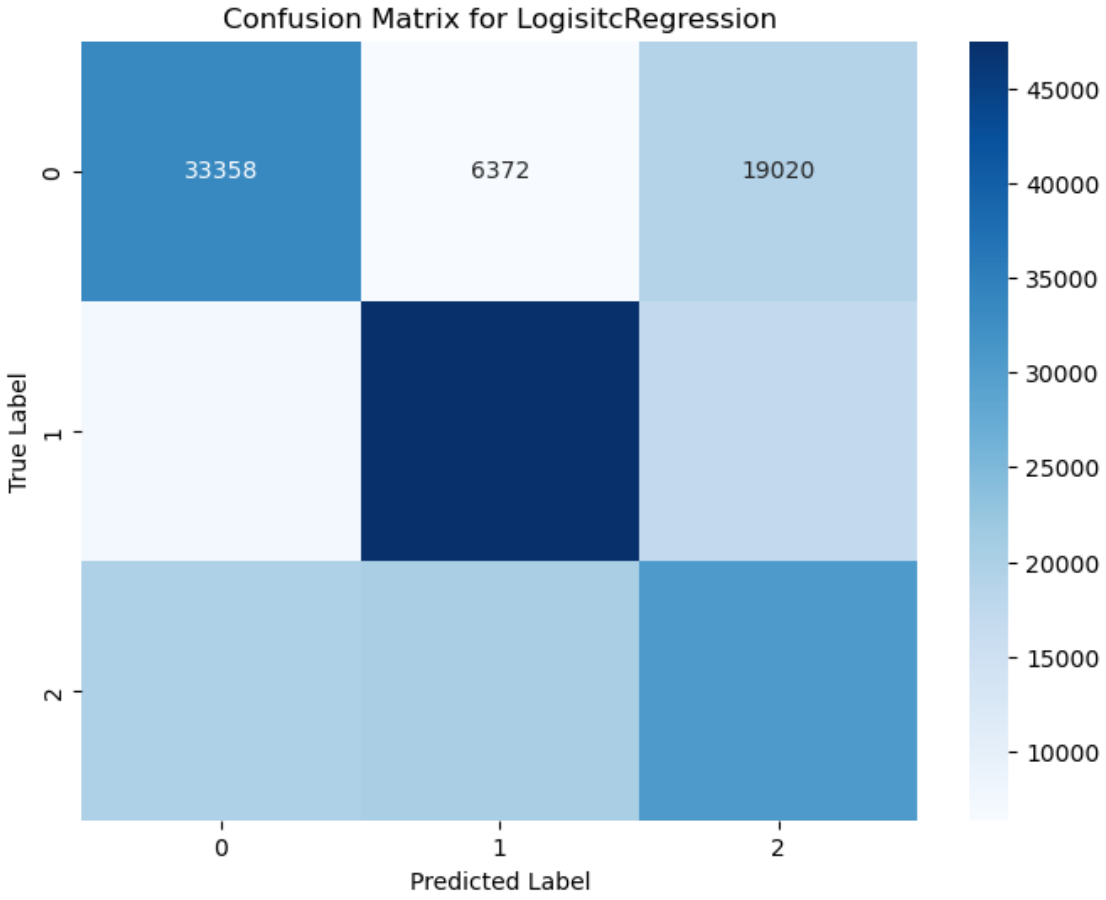



```
-----------------------------------------------------------
                                                  Time for detection
(Decision Tree) : 43.195 seconds…
-----------------------------------------------------------

+ ================================================================================
====================== +
                    2-For LogisitcRegression The Performance result is:
+ ================================================================================
====================== +
Accuracy :  0.5540999273291988
---------------------------------------------------
F1 :  [0.5613793  0.65050898 0.44505113]
```

```
--------------------------------------------------
Reacll :  [0.56779574 0.66068323 0.43368283]
--------------------------------------------------
Precision :  [0.55510625 0.64064334 0.45703148]
--------------------------------------------------
Confusion Matrix…
 [[33358  6372 19020]
 [ 7189 47480 17196]
 [19546 20261 30484]]
--------------------------------------------------
Classification Report…
```

	precision	recall	f1-score	support
above the national average	0.56	0.57	0.56	58750
below the national average	0.64	0.66	0.65	71865
same as the national average	0.46	0.43	0.45	70291
accuracy			0.55	200906
macro avg	0.55	0.55	0.55	200906
weighted avg	0.55	0.55	0.55	200906

```
--------------------------------------------------
Plotting Confusion Matrix…
```




```
------------------------------------------------------------
                                                  Time for detection
(LogisitcRegression) : 57.615 seconds…
------------------------------------------------------------
```

[63]:

```
                Model  Accuracy  \
0       Decision Tree  0.673141
1  LogisitcRegression  0.554100

                                          Precision  \
0  [0.657006349405922, 0.7432798811972915, 0.6086…
1  [0.5551062519761037, 0.640643341923819, 0.4570…

                                             Recall  \
0  [0.7115574468085106, 0.7591456202602102, 0.553…
1  [0.567795744680851, 0.6606832254922425, 0.4336…

                                           F1_Score  \
```

```
0   [0.6831946918563794, 0.7511289789624409, 0.579…
1   [0.561379298738672, 0.6505089808053268, 0.4450…

                                        Confusion Matrix
0   [[41804, 4115, 12831], [5139, 54556, 12170], […
1   [[33358, 6372, 19020], [7189, 47480, 17196], […
```

6.3.3 5.3.3 Results Visualizations

[64]:
```
plt.figure(figsize=(14, 6))
sns.barplot(x=list(dict(models).keys()), y=acc, palette="inferno")
plt.title("Model's Accuracies", fontsize=22)
plt.xlabel("Models", fontsize=17)
plt.ylabel("Accuracy", fontsize=17)
plt.show()
```