databricks bdp_cw2_lihtcdata

LIHTC Data Transformation and Visualization

Geeta Yadav
Student-ID (S2120665)

Data Overview

LIHTC (Low-Income Housing Tax Credit) dataset developed by HUD (Housing Urban Development). The dataset includes information on the mix of housing units, size and location of individual projects. The database columns are HUD ID (which is a Unique Project ID), project details like name, address including street number, zip code, city code and state code, company details like name, address, the total number of units and low-income units, number of bedrooms, the year the credit was allocated, telephone number and company owner name, year of project allocation, whether the project was original construction or rehabilitation. In addition, these data are geocoded and geographically distributed.

Import and display Data

Storing the file in the cloud to perform the analyse, transform and visualise the data.

```
# File location and type
file_location = "/FileStore/tables/LIHTCPUB.CSV"
file_type = "CSV"

# CSV options
infer_schema = "true"
first_row_is_header = "false"
delimiter = ","

# The applied options are for CSV files. For other file types, these will be ignored.
lihtc = spark.read.format(file_type) \
  .option("inferSchema", True) \
  .option("header", True) \
  .option("sep", delimiter) \
  .load(file_location)

display(lihtc)
```

	hud_id	project	proj_add
1	AKA0000X003	EAGLE RIDGE TOWNHOMES	1775 NORTH THUMA
2	AKA0000X018	GATEWAY-SEWARD ASSOCIATES, LTD PTN	1810 PHOENIX ROAD
3	AKA0000X022	JUNEAU AFFORDABLE RENTALS, LLC	SCATTERED SITE
4	AKA0000X024	MILL BAY TOWNHOMES, LLC	1223 MILL BAY ROAD
5	AKA0000X030	TURNAGAIN PLACE APTS	2708 COHO WAY
6	AKA0000X031	VISTA ROSE PHASE I	1250 N LUCILLE ST
7	AKA0000X032	VISTA ROSE PHASE II	1240 N LUCILLE ST

Truncated results, showing first 1000 rows.

Importing Libraries

Import the useful libraries to support and perform the required functions to transform and visualize the data.

```
import numpy as np
import pandas as pd
import pyspark as ps
# # import pyspark class Row from module sql
import pyspark.sql as pysql
from pyspark.sql.functions import avg
from pyspark.sql.functions import isnan, when, count, col

import matplotlib.pyplot as plt
```

Data Analysis and Transformation

Steps to perform analyse and transform the data are as follows:

1. The data's reliability
2. Verify the data's accuracy
3. Verify the data's consistency
4. Check the data consistency

1. Checking numbers of data rows and columns

```
print((lihtc.count(), len(lihtc.columns)))
```

```
(50567, 75)
```

2. Viewing data

```
lihtc.display(10)
```

	hud_id	project	proj_add
1	AKA0000X003	EAGLE RIDGE TOWNHOMES	1775 NORTH THUMA
2	AKA0000X018	GATEWAY-SEWARD ASSOCIATES, LTD PTN	1810 PHOENIX ROAD
3	AKA0000X022	JUNEAU AFFORDABLE RENTALS, LLC	SCATTERED SITE
4	AKA0000X024	MILL BAY TOWNHOMES, LLC	1223 MILL BAY ROAD
5	AKA0000X030	TURNAGAIN PLACE APTS	2708 COHO WAY
6	AKA0000X031	VISTA ROSE PHASE I	1250 N LUCILLE ST
7	AKA0000X032	VISTA ROSE PHASE II	1240 N LUCILLE ST

Truncated results, showing first 1000 rows.

```
print((lihtc.count(), len(lihtc.columns)))
```

```
(50567, 75)
```

3. Dropping unnecessary columns from data

Transforming data by selecting columns the necessary and useful information columns from the data and dropping unnecessary columns. It could provide the summary of useful information from the data.

```
lihtc_1 =
lihtc.select("hud_id","project","proj_cty","proj_st","n_1br","n_2br","n_3br","n_4br","yr_pis","n_units","li_units","ty
pe")
```

4. Display transformed data (lihtc_1)

```
display(lihtc_1)
```

	hud_id	project	proj_cty	proj_st	n_1br	n
1	AKA0000X003	EAGLE RIDGE TOWNHOMES	PALMER	AK	0	0
2	AKA0000X018	GATEWAY-SEWARD ASSOCIATES, LTD PTN	SEWARD	AK	null	n
3	AKA0000X022	JUNEAU AFFORDABLE RENTALS, LLC	JUNEAU	AK	0	0
4	AKA0000X024	MILL BAY TOWNHOMES, LLC	KODIAK	AK	null	n
5	AKA0000X030	TURNAGAIN PLACE APTS	ANCHORAGE	AK	null	n
6	AKA0000X031	VISTA ROSE PHASE I	WASILLA	AK	null	n
7	AKA0000X032	VISTA ROSE PHASE II	WASILLA	AK	null	n

Truncated results, showing first 1000 rows.

Describing nature of transformed data (lihtc_1) columns

```
lihtc_1.describe()
```

```
Out[30]: DataFrame[summary: string, hud_id: string, project: string, proj_cty: string, proj_st: string, n_1br: string,
n_2br: string, n_3br: string, n_4br: string, yr_pis: string, n_units: string, li_units: string, type: string]
```

Printing schema of (lihtc_1) data

```
lihtc_1.printSchema()
```

```
root
 |-- hud_id: string (nullable = true)
 |-- project: string (nullable = true)
 |-- proj_cty: string (nullable = true)
 |-- proj_st: string (nullable = true)
 |-- n_1br: integer (nullable = true)
 |-- n_2br: integer (nullable = true)
 |-- n_3br: integer (nullable = true)
 |-- n_4br: integer (nullable = true)
 |-- yr_pis: integer (nullable = true)
 |-- n_units: integer (nullable = true)
 |-- li_units: integer (nullable = true)
 |-- type: integer (nullable = true)
```

Filling null values in the columns

```
lihtc_1.fillna("--")
display(lihtc_1)
```

	hud_id	project	proj_cty	proj_st	n_1br	n
1	AKA0000X003	EAGLE RIDGE TOWNHOMES	PALMER	AK	0	0
2	AKA0000X018	GATEWAY-SEWARD ASSOCIATES, LTD PTN	SEWARD	AK	null	n
3	AKA0000X022	JUNEAU AFFORDABLE RENTALS, LLC	JUNEAU	AK	0	0
4	AKA0000X024	MILL BAY TOWNHOMES, LLC	KODIAK	AK	null	n

5	AKA0000X030	TURNAGAIN PLACE APTS	ANCHORAGE	AK	null	n
6	AKA0000X031	VISTA ROSE PHASE I	WASILLA	AK	null	n
7	AKA0000X032	VISTA ROSE PHASE II	WASILLA	AK	null	n

Truncated results, showing first 1000 rows.

Renaming columns

Giving new names to the columns for the better understanding of columns information.

```
lihtc_2 = lihtc_1.withColumnRenamed("proj_cty","project_city") \
    .withColumnRenamed("n_1br","1bedroom_unit") \
    .withColumnRenamed("n_2br","2bedroom_unit") \
    .withColumnRenamed("n_3br","3bedroom_unit") \
    .withColumnRenamed("n_4br","4bedroom_unit") \
    .withColumnRenamed("yr_pis","serviceplaced_yr") \
    .withColumnRenamed("n_units","total_num_units") \
    .withColumnRenamed("li_units","lowincome_units") \
    .withColumnRenamed("hud_id","project_uniqueID") \
    .withColumnRenamed("proj_st","project_state")

#lihtc_2.printSchema()
```

Display data (lihtc_2) with rename columns

```
display(lihtc_2)
```

	project_uniqueID	project	project_city	project_state	1bedroon
1					

2	AKA0000X018	GATEWAY-SEWARD ASSOCIATES, LTD PTN	SEWARD	AK	null
3	AKA0000X022	JUNEAU AFFORDABLE RENTALS, LLC	JUNEAU	AK	0
4	AKA0000X024	MILL BAY TOWNHOMES, LLC	KODIAK	AK	null
5	AKA0000X030	TURNAGAIN PLACE APTS	ANCHORAGE	AK	null
6	AKA0000X031	VISTA ROSE PHASE I	WASILLA	AK	null
7	AKA0000X032	VISTA ROSE PHASE II	WASILLA	AK	null

Truncated results, showing first 1000 rows.

Printing count of rows and columns of (lihtc_2) transformed data

```
print((lihtc_2.count(), len(lihtc_2.columns)))
```

```
(50567, 12)
```

Dropping Duplicates

Dropping duplicates from the project_uniqueID column if any. project_uniqueID column should not have any duplicate in the column so if there is any duplicate in the column it will get drop.

```
dropdup_lihtc = lihtc_2.dropDuplicates(["project_uniqueID"])
```

```
dropdup_lihtc.show(truncate=False)
```

```
+----------------+--------------------------------------------------+------------+-------------+-------------+--------
-----+-------------+-------------+---------------+--------------+---------------+----+
|project_uniqueID|project                                           |project_city|project_state|1bedroom_unit|2bedroom
_unit|3bedroom_unit|4bedroom_unit|serviceplaced_yr|total_num_units|lowincome_units|type|
```

```
+---------------+--------------------------------------------------+-----------+-------------+-------------+--------
-----+-------------+-------------+---------------+---------------+---------------+----+
|AKA0000X003    |EAGLE RIDGE TOWNHOMES                            |PALMER     |AK           |0            |0
|0            |0            |8888           |33             |0              |null|
|AKA0000X018    |GATEWAY-SEWARD ASSOCIATES, LTD PTN                |SEWARD     |AK           |null         |null
|null         |null         |9999           |20             |null           |null|
|AKA0000X022    |JUNEAU AFFORDABLE RENTALS, LLC                    |JUNEAU     |AK           |0            |0
|20           |5            |9999           |25             |25             |1   |
|AKA0000X024    |MILL BAY TOWNHOMES, LLC                           |KODIAK     |AK           |null         |null
|null         |null         |9999           |20             |null           |null|
|AKA0000X030    |TURNAGAIN PLACE APTS                              |ANCHORAGE  |AK           |null         |null
|null         |null         |9999           |29             |null           |null|
|AKA0000X031    |VISTA ROSE PHASE I                                |WASILLA    |AK           |null         |null
|null         |null         |9999           |44             |null           |null|
|AKA0000X032    |VISTA ROSE PHASE II                               |WASILLA    |AK           |null         |null
|null         |null         |9999           |36             |null           |null|
```

Printing count of dropdup_lihtc transformed data

There is no duplicate row found in the project_uniqueID cloumn.

```
print((dropdup_lihtc.count(), len(dropdup_lihtc.columns)))
```

```
(50567, 12)
```

Removing rows

Removing all the rows containing fake data. 8888 number in serviceplaced_yr giving a data of the units which are not confirmed so it is considering as a fake data.

```
lihtc2b = lihtc_2[ (lihtc_2['serviceplaced_yr'] != 8888)]
print((lihtc2b.count(), len(lihtc2b.columns)))
```

```
(47458, 12)
```

Display lihtc2b tranformed data

```
display(lihtc2b)
```

	project_uniqueID	project	project_city	project_state	1bedroon
1	AKA0000X018	GATEWAY-SEWARD ASSOCIATES, LTD PTN	SEWARD	AK	null
2	AKA0000X022	JUNEAU AFFORDABLE RENTALS, LLC	JUNEAU	AK	0
3	AKA0000X024	MILL BAY TOWNHOMES, LLC	KODIAK	AK	null
4	AKA0000X030	TURNAGAIN PLACE APTS	ANCHORAGE	AK	null
5	AKA0000X031	VISTA ROSE PHASE I	WASILLA	AK	null
6	AKA0000X032	VISTA ROSE PHASE II	WASILLA	AK	null
7	AKA0000X034	YENLO PHASE I AND II	WASILLA	AK	null

Truncated results, showing first 1000 rows.

Removing rows

Removing all the rows with 9999 number in serviceplaced_yr its giving a data of the units which years of service placed is not given.

```
lihtc_3 = lihtc2b[ (lihtc2b['serviceplaced_yr'] != 9999)]
print((lihtc_3.count(), len(lihtc_3.columns)))
```

```
(46489, 12)
```

Display (lihtc3) data

```
display (lihtc_3)
```

	project_uniqueID	project	project_city	project_state	1bedroon
1	AKA19870005	GLACIER PARK-KETCHIKAN ASSOCIATES, LTD PARTNERSHIP	KETCHIKAN	AK	0
2	AKA19890010	PARK WEST APTS	FAIRBANKS	AK	41
3	AKA19900005	TYSON'S TERRACE	SITKA	AK	16
4	AKA19910005	NORTHWOOD APTS	SOLDOTNA	AK	23
5	AKA19920005	DUSTY TRAILS APTS	HAINES	AK	12
6	AKA19930005	CHINOOK VILLA	WASILLA	AK	32
7	AKA19930010	STRASBAUGH PLACE	JUNEAU	AK	2

Truncated results, showing first 1000 rows.

Data Storage

Storing tranformed data

```
lihtc_3.write.format("csv") \
  .option("inferSchema", True) \
  .option("header", True) \
  .save("/FileStore/tables/LIHTC_3.CSV")
```

```
dbutils.fs.rm("/FileStore/tables/LIHTC_3.CSV", True)
```

```
Out[52]: True
```

Reading stored csv

```
lihtc4 = spark.read.format("csv") \
  .option("inferSchema", True) \
  .option("header", True) \
  .option("sep", delimiter) \
  .load("/FileStore/tables/LIHTC_3.CSV")
```

```
display(lihtc4)
```

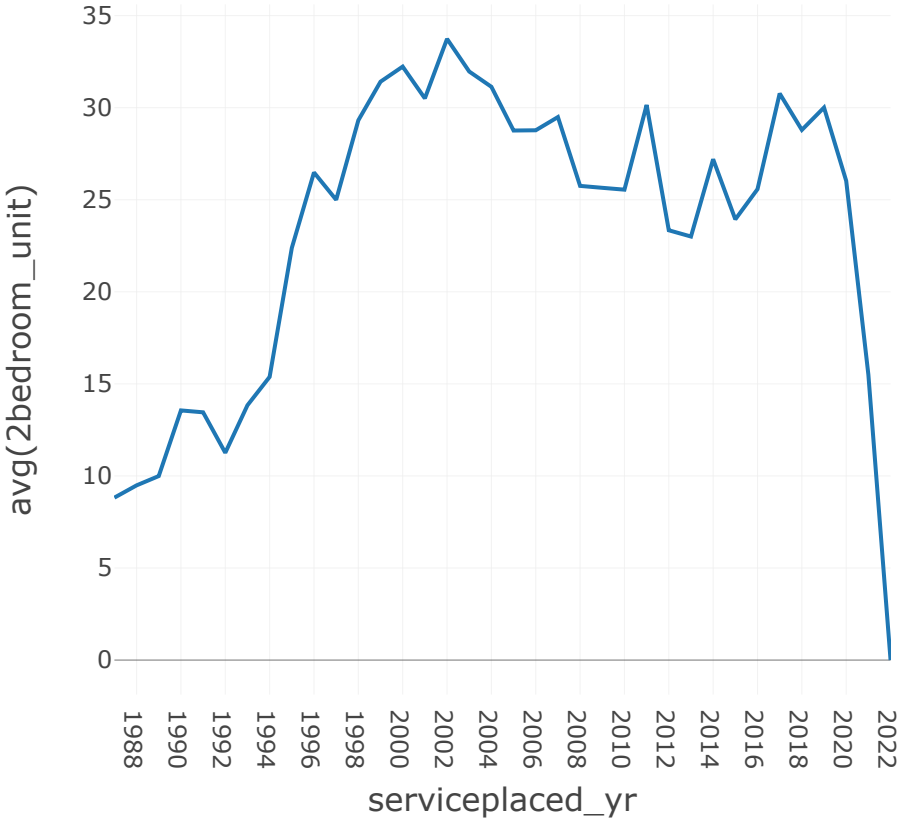
	project_uniqueID	project	project_city	project_state	1bedroon
1	AKA19870005	GLACIER PARK-KETCHIKAN ASSOCIATES, LTD PARTNERSHIP	KETCHIKAN	AK	0
2	AKA19890010	PARK WEST APTS	FAIRBANKS	AK	41
3	AKA19900005	TYSON'S TERRACE	SITKA	AK	16
4	AKA19910005	NORTHWOOD APTS	SOLDOTNA	AK	23
5	AKA19920005	DUSTY TRAILS APTS	HAINES	AK	12
6	AKA19930005	CHINOOK VILLA	WASILLA	AK	32
7					

Truncated results, showing first 1000 rows.

Visualization

1. Find the average number of 2 bedrooms units placed in all years.

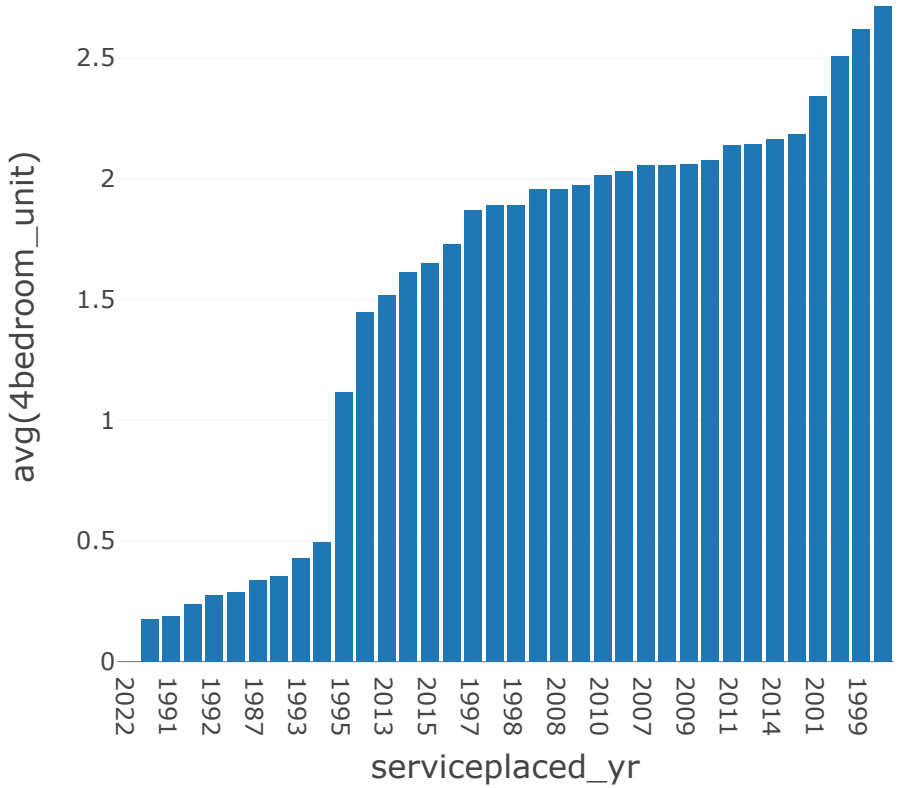
```
vis1 = lihtc4.groupby("serviceplaced_yr").avg("2bedroom_unit").orderBy(avg("2bedroom_unit"),ascending=True)
display(vis1)
```



It can be seen that the number of 2 bed room unit has been increased in the last 22 years. In year 2002 the number of unit placed by the the projects was the highest in comparison of last 22 years which means the rquirements of 2bedroom units was increased.

2. Find the average number of 4 bedroom units placed in all years.

```
vis1 = lihtc4.groupby("serviceplaced_yr").avg("4bedroom_unit").orderBy(avg("4bedroom_unit"),ascending=True)
display(vis1)
```



The bar graph shows that average number of the 4-bedroom units. It can be seen that the minimum number of units placed in year 2022 and the highest number of the units placed in year 2004.

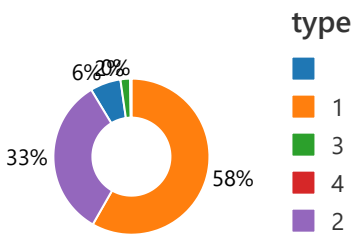
3.Find the types of the 2 bedroom units.

```
vis1 = lihtc4.groupBy("type").agg(count("2bedroom_unit"))
display(vis1)
```

	type	count(2bedroom_unit)
1	null	2520
2	1	23052
3	3	787
4	4	107
5	2	13110

Showing all 5 rows.

```
display(vis1)
```



The above Pie chart illustrates the percentage of the types of 2 bedroom units, it can be seen that the maximum fifty eight percent

units are of new construction units, Acquisition and rehab types of units has thirty three percent whereas twenty eight percent of units are both new and rehab types of units. Lastly, there are six percent of 2 bedroom units which are not defined in the data.