



# AIR BNB Price Prediction

# Problem Statement

- ◆ The purpose of this project is to explore the data using pyspark and predict the price of an Airbnb listing from features extracted from the listings using SparkML.

# Data Source

- <https://www.kaggle.com/stevezhenghp/airbnb-price-prediction>
- Number of columns : 29
- Number of Rows : 74112
- Dataset Size: 99MB

# Data Source details

```
df.printSchema()

root
|-- id: string (nullable = true)
|-- log_price: double (nullable = true)
|-- property_type: string (nullable = true)
|-- room_type: string (nullable = true)
|-- amenities: string (nullable = true)
|-- accommodates: integer (nullable = true)
|-- bathrooms: double (nullable = true)
|-- bed_type: string (nullable = true)
|-- cancellation_policy: string (nullable = true)
|-- cleaning_fee: boolean (nullable = true)
|-- city: string (nullable = true)
|-- description: string (nullable = true)
|-- first_review: string (nullable = true)
|-- host_has_profile_pic: string (nullable = true)
|-- host_identity_verified: string (nullable = true)
|-- host_response_rate: string (nullable = true)
|-- host_since: string (nullable = true)
|-- instant_bookable: string (nullable = true)
|-- last_review: string (nullable = true)
|-- latitude: double (nullable = true)
|-- longitude: double (nullable = true)
|-- name: string (nullable = true)
|-- neighbourhood: string (nullable = true)
|-- number_of_reviews: integer (nullable = true)
|-- review_scores_rating: double (nullable = true)
|-- thumbnail_url: string (nullable = true)
|-- zipcode: string (nullable = true)
|-- bedrooms: double (nullable = true)
|-- beds: double (nullable = true)
```

id	- AIRBNB unique ID
log_price	- Price for the AIRBNB
property type	- Apartment/house/villa
room type	- Entire house/sharing/single room
amenities	- TV/Wifi/Hot water
accommodates	- Accomodates how many people
bathroom	- How many bathrooms
bed_type	- Real bed/couch/airbed
Cancellation policy	- Strict/Flexible
cleaning fees	- True/False
description	- Description
first_review	- first review date
host_has_profile pic	- True/False
host_identity_verified	- True/False
host_response_rate	- Percentage
host_since	- Date
instant_bookable	- True/False
last_review	- Date
latitude	- Latitude position
longitude	-Longitude position
name	- Name of the AIRBNB
neighbourhood	- Neighborhood places
number_of_reviews	- Total number of reviews
review_scores_rating	- Average rating for the AIRBNB
thumbnail_url	- URL for the AIRBNB website
zipcode	- Zipcode of the location
bedrooms	- Number of bedrooms
beds	- Number of beds

# Displaying the first row of the dataframe

```
df.show(n=1,truncate=False,vertical=True)
```

```
-RECORD 0-----
id                | 6304928
log_price         | 5.1298987149230735
property_type     | Apartment
room_type         | Entire home/apt
amenities          | {"Wireless Internet","Air conditioning",Kitchen,Heating,"Family/kid friendly",Washer,Dryer,"Smoke detector","Fire
accommodates      | 7
bathrooms         | 1.0
bed_type          | Real Bed
cancellation_policy | strict
cleaning_fee      | true
city              | NYC
description        | Enjoy travelling during your stay in Manhattan. My place is centrally located near Times Square and Central Park
first_review       | 2017-08-05
host_has_profile_pic | t
host_identity_verified | f
host_response_rate | 100%
host_since        | 2017-06-19
instant_bookable  | t
last_review       | 2017-09-23
latitude          | 40.766115415949685
longitude         | -73.98903992265213
name              | Superb 3BR Apt Located Near Times Square
neighbourhood     | Hell's Kitchen
number_of_reviews | 6
review_scores_rating | 93.0
thumbnail_url     | https://a0.muscache.com/im/pictures/348a55fe-4b65-452a-b48a-bfecb3b58a66.jpg?aki\_policy=small
zipcode          | 10019
bedrooms          | 3.0
beds              | 3.0
```

```
only showing top 1 row
```



# Total count of each property type

```
SELECT property_type,count(id) as Total_count
FROM price
group by property_type
order by Total_count desc
```

```
####
```

```
print("Total count based on property type")
spark.sql(query).show()
```

Total count based on property type

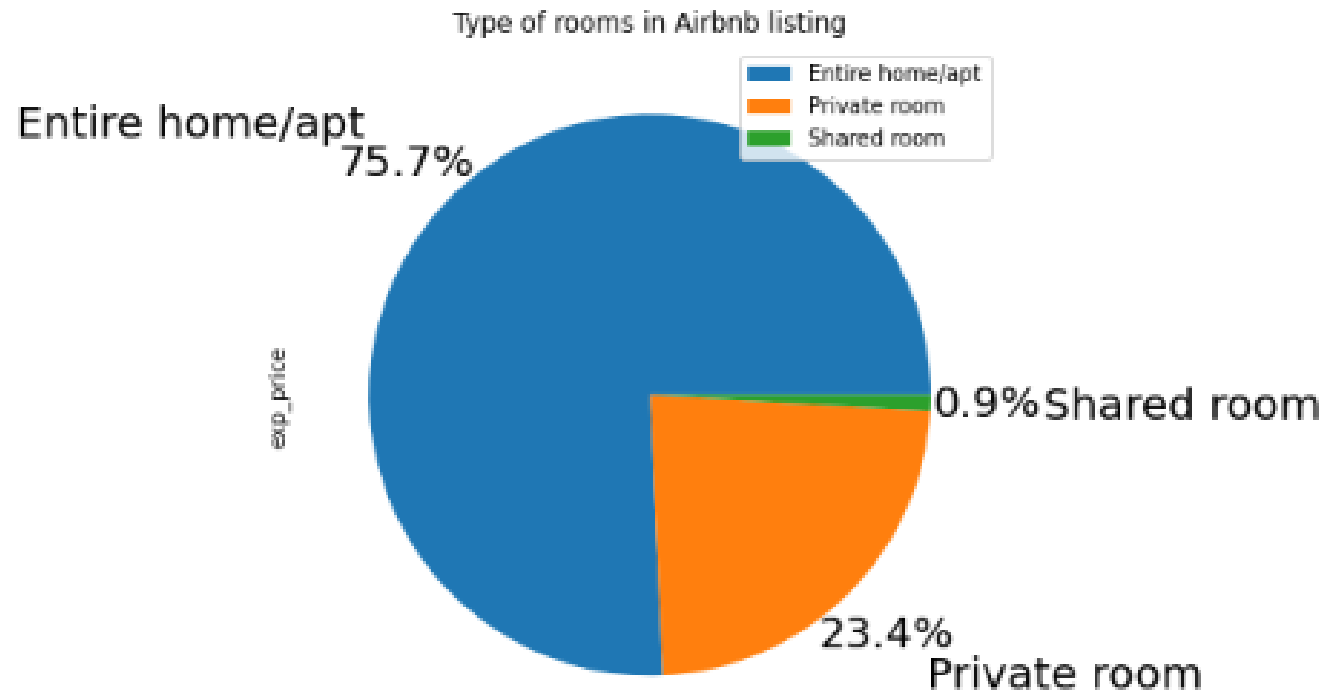
property_type	Total_count
Apartment	24750
House	8903
Condominium	1421
Townhouse	940
Loft	719
Guesthouse	335
Other	331
Bed & Breakfast	286
Bungalow	216
Guest suite	97
Dorm	90
villa	68
In-law	61
Hostel	46
Cabin	45
Camper/RV	35
Boat	35
Boutique hotel	32
Timeshare	30
Serviced apartment	12

only showing top 20 rows

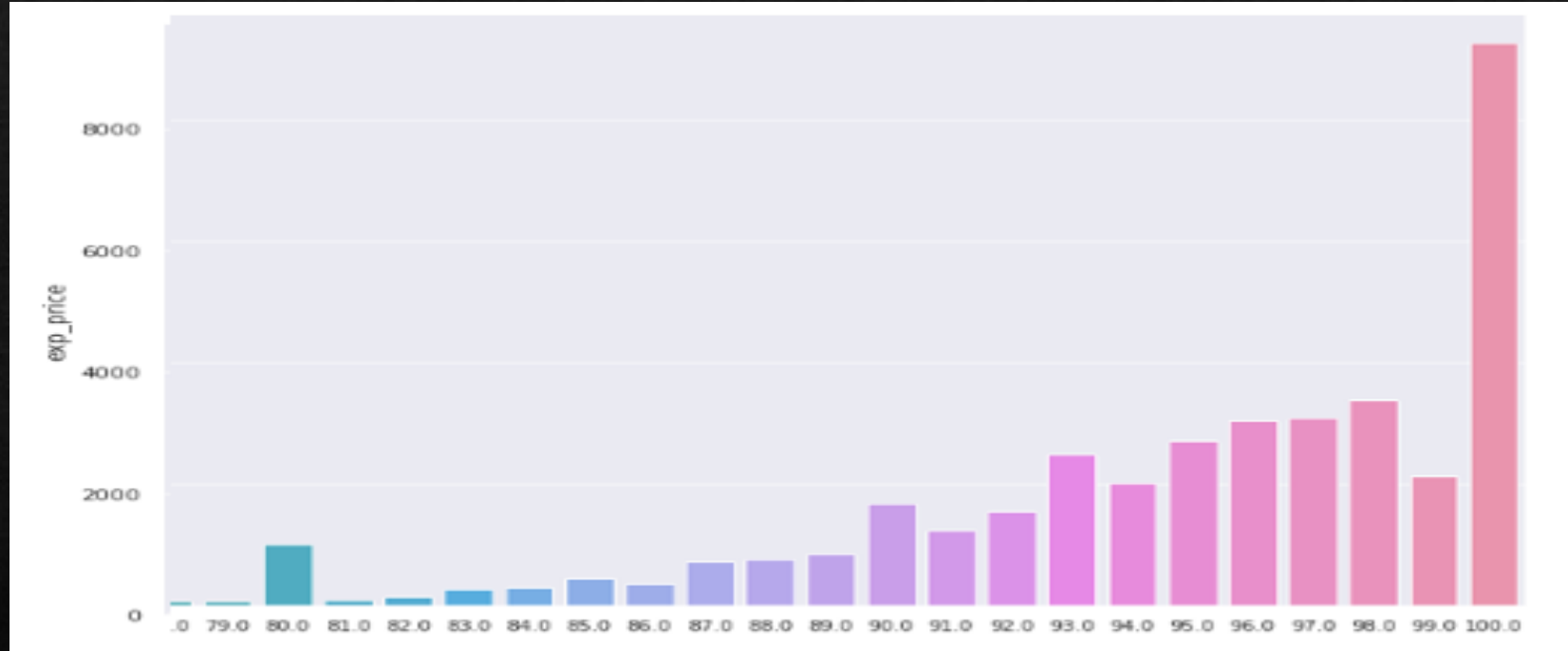
# Type of rooms available in Airbnb listings

```
df1.groupby(['room_type']).sum().plot(kind='pie',y= 'exp_price',radius = 1,title=
    "Type of rooms in Airbnb listing",autopct='%1.1f%%',fontsize=20,figsize=(6, 6),pctdistance=1.2,labeldistance=1.4)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4e813be210>



# Is there any correlation between the price and rating of the Airbnb listing?



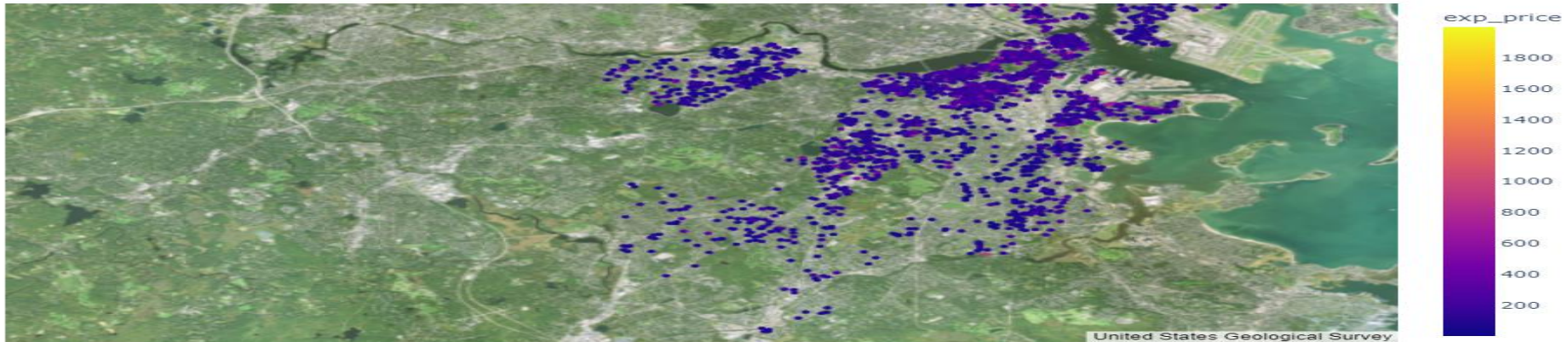
It could be inferred from the above graph that, the price of the Airbnb listing is proportional to the rating



# Location of the Airbnb listings along with the price

```
fig = px.scatter_mapbox(df1,lat="latitude",lon="longitude",hover_data=['name'],color='exp_price',zoom=10)
fig.update_layout(
    title = f'Airbnb prices in ',geo_scope='usa',width=1000, height=600,mapbox_style="white-bg",
    mapbox_layers=[{
        "below": 'traces',"sourcetype": "raster","sourceattribution": "United States Geological Survey",
        "source": ["https://basemap.nationalmap.gov/arcgis/rest/services/USGSImageryOnly/MapServer/tile/{z}/{y}/{x}"]
    }]
)
fig.show()
```

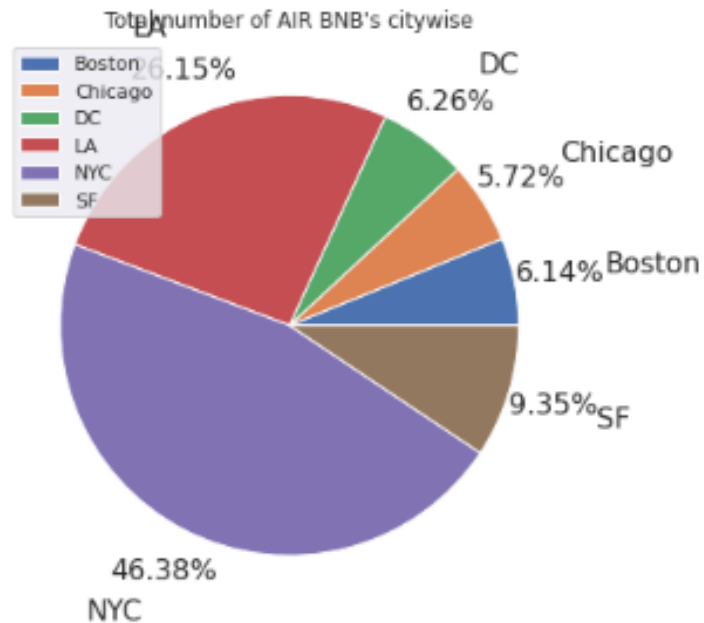
Airbnb prices in



# Which city has the highest number of Airbnbs?

```
df1.groupby(['city']).count().plot(kind='pie',y= 'id',radius = 1,title="Total number of AIR BNB's citywise",autopct='%1.2f%%',fontsize=16,figsize=(6, 6),pctdistance=1.2,labeldistance=1.4)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4e80eab510>



# Does the rating of Airbnb listing depend on the cancellation policy?

```
df.createOrReplaceTempView('cancellation')

query = """
SELECT cancellation_policy,count(cancellation_policy),sum(review_scores_rating)/count(review_scores_rating) as avg_rating
FROM cancellation
group by cancellation_policy
order by avg_rating desc

"""

print("Cancellation policy")
spark.sql(query).show()
```

```
Cancellation policy
+-----+-----+-----+
|cancellation_policy|count(cancellation_policy)|      avg_rating|
+-----+-----+-----+
|      moderate|      11386|95.08598278587739|
|     flexible|       7342|94.55107600108963|
|       strict|     19733|93.82673693812396|
|super_strict_30|        33|89.33333333333333|
|super_strict_60|         6|85.66666666666667|
+-----+-----+-----+
```

From the above figures, it could be inferred that the rating doesn't depend on the cancellation policy

# Correlation of the feature variables with the log\_price

```
import six
for i in df4.columns:
    if not( isinstance(df4.select(i).take(1)[0][0], six.string_types)):
        print( "Correlation to log_price for ", i, df4.stat.corr('log_price',i))
```

```
Correlation to log_price for log_price 1.0
Correlation to log_price for accommodates 0.5821216346150896
Correlation to log_price for bathrooms 0.3066494495230136
Correlation to log_price for latitude 0.000853903764855051
Correlation to log_price for longitude -0.05823807457798872
Correlation to log_price for number_of_reviews -0.012968757197144054
Correlation to log_price for review_scores_rating 0.07742280233346707
Correlation to log_price for bedrooms 0.4808712214477055
Correlation to log_price for beds 0.4471387234238176
```



# Linear Regression to predict the Log\_price based on the features of the listings

```
(trainingData, testData) = df4.randomSplit([0.8, 0.2])
```

```
lr = LinearRegression(featuresCol='features', labelCol='log_price', maxIter=10, regParam=0.3, elasticNetParam=0.8)
```

```
lr_model = lr.fit(trainingData)
```

```
print("Coefficients: " + str(lr_model.coefficients))
```

```
print("Intercept: " + str(lr_model.intercept))
```



THANK YOU