COMP5310

Project Stage 1

Team: Pikachus

530146887 yzha9685

530147530 yyan6135

Contents

yzha9685

1. Description

1.1 Topic Description

The topic or question I chose is how do people find the right house in Sydney? This question stems from my experience of renting an apartment, where I experienced a series of problems such as information silos, rental scams, and false information. Although I was lucky enough to eventually find the right place to live, the experience was a bad one. It became a complex issue how people evaluate the rental information in a certain area, including their transportation situation, environment, price, security, and other factors.

1.2 Who can get help from this project?

Anyone in need of rental housing can be assisted in this program. The audience for this project includes, but is not limited to, people who are long-term residents of the Sydney area and international students. They can use the project's research to gain a more detailed understanding of the rental situation in various areas of Sydney, as well as to more easily assess the attributes associated with the property they are looking for.

1.3 The relationship between Topic and dataset

This dataset shows the price of housing, the population, transportation, environment rating, recommended people to live in, and other related information in each area of Sydney. By analyzing this dataset, we can provide a reasonable choice of rental areas for different rental groups and solve the problem of how to rent a suitable apartment in Sydney.

2. Dataset

2.1 Data Source

This dataset is sourced from the Kaggle data platform. The link to this dataset is https://www.kaggle.com/datasets/karltse/sydney-suburbs-reviews?resource=download.Data Using License and Restriction. And the direct source of the data is https://sydneysuburbreviews.com/suburb-rankings/, the Kaggle author moved the data from the platform directly to Kaggle.

2.2 Data Usage License and Restrictions

The Kaggle owner of the data does not specify the permissions and restrictions on the use of the data. So, I contacted the original producer of the data and obtained permission to use the original data.

2.3 Changes of Dataset

The "Review Link" has been removed because it is not important and directly related to this project.

2.4 Data Dictionary

Feature Name	Description	Data Type	Data Format
Name	Area Name	Nominal	String
Region	Region of the area	Nominal	String
Population (rounded)*	Number of the population	Interval	Int
Postcode	Number of the postcode	Nominal	Int
Ethnic Breakdown 2016	Proportion of different ethnic groups in the region	Ratio	Structural Data
Median House Price (2020)	Median house price of 2020	Interval	String
Median House Price (2021)	Median house price of 2021	Interval	String
% Change	Price growth rate from 2020 to 2021	Ratio	Double
Median House Rent (per week)	Median house rent of each week	Interval	String
Median Apartment Price (2020)	Median apartment price of 2020	Interval	String
Median Apartment Rent (per week)	Median apartment rent price of each week	Interval	String
Public Housing %	The rate of public housing	Ratio	Double
Avg. Years Held	Mean years hold of each house	Interval	Double
Time to CBD (Public Transport) [Town Hall St]	Time to CBD by public transport	Ratio	Int
Time to CBD (Driving) [Town Hall St]	Time to CBD by car	Ratio	Int
Nearest Train Station	Nearest Train Station	Nominal	String
Highlights/Attractions	Nearest Highlights and Atrractions	Nominal	String
Ideal for	Ideal for groups of people to live	Nominal	String
Traffic	Rate of traffic	Ordinal	Int
Public Transport	Rate of public transport	Ordinal	Int
Affordability (Rental)	Rate of rent affordability	Ordinal	Int
Affordability (Buying)	Rate of buy affordability	Ordinal	Int
Nature	Rate of nature	Ordinal	Int
Noise	Rate of noise	Ordinal	Int
Things to See/Do	Rate of thing to do	Ordinal	Int
Family-Friendliness	Rate of family friendliness	Ordinal	Int
Pet Friendliness	Rate of pet friendliness	Ordinal	Int
Safety	Rate of safety	Ordinal	Int
Overall Rating	Rate of summary	Ordinal	Int
Review Link	The link of each place	Text	String

3. Clean the data

3.1 Checking and handling missing data problems.

```
dataframe.isnull().sum().sort_values()
dataframe=dataframe.dropna(axis=0,how='any')
```

3.2 Checking for data duplication problems.

```
dataframe.duplicated().any()
```

```
False
```

3.3 Splitting of structural data.

```
dataframe=pd.read_csv('Sydney Suburbs Reviews noNA.csv')
Eb=dataframe['Ethnic Breakdown 2016'].str.split(', ',expand=True)
Eb.columns=['TopCountry1','TopCountry2','TopCountry3','TopCountry4','TopCountry5']
```

```
print(Eb)
print(type(Eb))
Eb.shape
```

```
         TopCountry1       TopCountry2       TopCountry3      TopCountry4  \
0      Chinese 17.1%     English 16.8%  Australian 14.0%      Indian 5.9%
1      English 23.0%  Australian 21.1%      Chinese 9.8%       Irish 8.9%
2      English 19.4%  Australian 16.4%        Irish 9.5%    Scottish 6.2%
3      English 28.2%  Australian 26.3%        Irish 9.8%    Scottish 6.5%
4      English 24.9%  Australian 15.5%       Irish 11.0%     Chinese 8.4%
..               ...               ...               ...              ...
93     Chinese 33.8%     English 13.1%   Australian 8.1%       Irish 6.2%
94     English 22.4%  Australian 12.4%        Irish 9.6%    Scottish 7.0%
95     Chinese 18.1%     English 13.0%  Australian 11.7%       Irish 5.5%
96  Vietnamese 16.3%    Lebanese 12.7%      Chinese 9.5%  Australian 6.8%
97     English 22.6%  Australian 17.5%       Irish 10.8%    Scottish 6.9%

      TopCountry5
0      Irish 5.6%
1   Scottish 5.7%
2      Greek 5.2%
3    Chinese 3.0%
4   Scottish 8.1%
```

3.4 Unified Data Types.

```python
import re
data1=dataframe['Time to CBD (Driving) [Town Hall St]']
bool=data1.str.contains('mins',case=True, flags=0,regex=True)
C1data=data1[bool]
print(C1data)
data2=dataframe['Time to CBD (Public Transport) [Town Hall St]']
bool=data2.str.contains('minuntes',case=True, flags=0,regex=True)
C2data=data2[bool]
print(C2data)
```

```
4     15 mins
Name: Time to CBD (Driving) [Town Hall St], dtype: object
45    25 minuntes
Name: Time to CBD (Public Transport) [Town Hall St], dtype: object
```

```python
dataframe['Time to CBD (Driving) [Town Hall St]'].loc[4]=15
dataframe['Time to CBD (Public Transport) [Town Hall St]'].loc[45]=25
```

3.5 Data type conversion.

```python
dataframe=pd.read_csv('Sydney Suburbs Reviews noNA Splitted.csv')
dataframe['Median House Price (2020)']=(dataframe['Median House Price (2020)'].str.strip('$'))
dataframe['Median House Price (2021)']=(dataframe['Median House Price (2021)'].str.strip('$'))
dataframe['Median House Rent (per week)']=(dataframe['Median House Rent (per week)'].str.strip(
dataframe['Median Apartment Price (2020)']=(dataframe['Median Apartment Price (2020)'].str.stri
dataframe['Median Apartment Rent (per week)']=(dataframe['Median Apartment Rent (per week)'].st
```

3.6 Unifying semantic data representations.

```python
class Recommend(Enum):
    FAMILY=1
    PROFESSIONAL=2
    RETIREE=3
    OTHER=4
    NONE=0

def classfiy(value):
    if value in {'small families','families','wealthy families'}:
        return (Recommend.FAMILY)
    elif value in {'professionals','young professionals'}:
        return (Recommend.PROFESSIONAL)
    elif value in{'retirees'}:
        return (Recommend.RETIREE)
    elif value==None:
        return (Recommend.NONE)
    else:
        return (Recommend.OTHER)
```

4. Simple data exploration

4.1 Simple statistics on house prices.

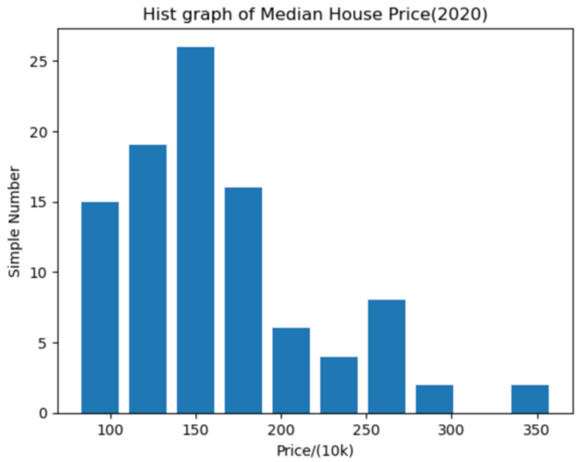
```python
data1=data['Median House Price (2020)'].str.replace(",","")
data=(data1.astype('float'))/10000
data.describe()
```

```
count     98.000000
mean     161.484694
std       57.815537
min       80.000000
25%      120.000000
50%      150.000000
75%      188.750000
max      360.000000
Name: Median House Price (2020), dtype: float64
```

4.2 Hist diagram of house prices.

```
plt.hist(data,rwidth=0.8)
plt.xlabel('Price/(10k)')
plt.ylabel('Simple Number')
plt.title('Hist graph of Median House Price(2020)')
plt.show
```

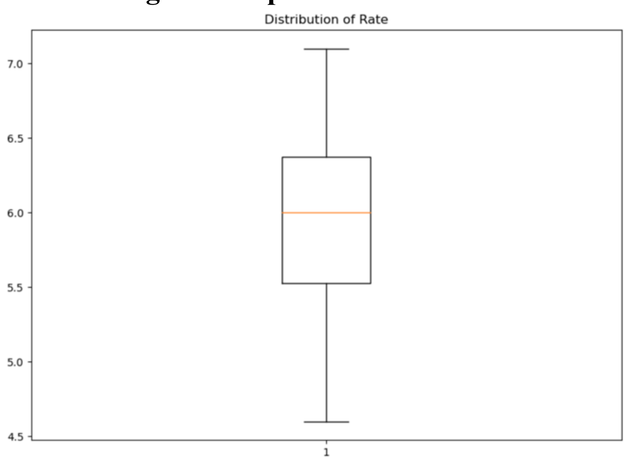
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



4.3 Plotting the scatter plot of rate over prices.

```
data=pd.read_csv('Sydney Suburbs Reviews clean.csv')
data=data.drop(columns=['Unnamed: 0'])
data2=data[['Median House Price (2021)','Overall Rating']]
data3= data2['Median House Price (2021)'].str.replace(",","")
plt.scatter(data3.astype('float'), data2['Overall Rating'])
plt.title('House price 2021 vs Rate')
plt.xlabel('Price/1000000')
plt.ylabel('Rate')
plt.show()
```

4.4 Plotting the box plot of rate.



Reference

Sydney Suburbs Reviews. (n.d.). Retrieved March 15, 2023, from www.kaggle.com website: https://www.kaggle.com/datasets/karltse/sydney-suburbs-reviews?select=Sydney+Suburbs+Reviews.csv

yyan6135

1. Identifying the topic

The consensus is employees are one of the most significant intangible assets of the enterprise based on their unmeasurable supportive and productive role. Therefore, understanding why their valuable employees are attrition is the beginning for executives to capture their human resources and the fundamentals to arrange and assign them effectively and efficiently.

What leads to the employees' attrition can be quantitively analysed, which will assist **executives** to realize the survival and prosperity of their enterprise. **The department of human resources** will possess more disposable steady intangible resources and the other departments, such as **R&D**, will be influenced less by attrition and save costs for training newcomers and employee fluctuation.

The dataset is shown factors that are regarded as possible factors leading to attrition and can be used to predict potential attritions. By comprehensively considering the multidimensional information of employees, we can predict the possible attrition for the enterprises and the departments more rationally and feasibly.

2. Dataset and metadata

The **dataset** is generated by IBM data scientists and published on Kaggle. I choose it for further study on **11th March 2023**. The **link** is enclosed here: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset/discussion/86957.

The **license** mentioned on Kaggle is named ‘Database Contents License (DbCL)’. It emphasises all granted rights only restricted by copyright, so it is open and free for academic usage.

Based on the unclear definition and presence of features with alternative functions ‘MonthlyIncome’, the columns ‘DailyRate’, ‘HourlyRate’, and ‘MonthlyRate’ have been removed. In terms of the blurred definition and consistent contents, the columns ‘EmployCount’ and ‘StandardHour’ have been deleted. All the employees in that dataset are over 18, so removing that column will benefit further research.

Attribute	Explanation	Original Data_type	Data_format
Attrition	Whether the employee leave the company (0=no, 1=yes)	Norminal	bool
Age	The age of the employee	Interval	int
BusinessTravel	How often the employee take the business teavel (1=Non-Travel, 2=Travel_Frequently, 3=Tavel_Rarely)	Ordinal	string
Department	Which department the employee work at (1=Human Resources, 2=Research & Development, 3=Sales)	Norminal	string
DistanceFromHome	How far the employee's home from the workplace	Ratio	int
Education	What the final degree of the employee (1=below college, 2=college, 3=bachelor, 4=master, 5=doctor)	Ordinal	int
EducationField	What field the employee studied during the education (1=Human Resources, 2=Life Sciences, 3=Marketing, 4=Medical, 5=Other, 6= Technical Degree)	Norminal	string
EmployeeNumber	Employee ID	Ordinal	int
EnvironmentSatisfaction	How satisfied the employee with the environment (1=low, 2=medium, 3=high, 4=very high)	Ordinal	int
Gender	Gender (1=FEMALE, 2=MALE)	Norminal	string
JobInvolvement	How involved the employee is in the position (1=low, 2=medium, 3=high, 4=very high)	Ordinal	int
JobLevel	How high the employees' position is	Ordinal	int
JobRole	What the role of the employee in the work (1= Healthcare Representative, 2= Human Resources, 3= Laboratory Technician, 4=Manager, 5= Manufacturing Director, 6= Research Director, 7= Research Scientist, 8= Sales Executive, 9= Sales Representative)	Norminal	string
JobSatisfaction	How satisfied the employee with the job (1=low, 2=medium, 3=high, 4=very high)	Ordinal	int
MaritalStatus	What the marital status of the employee (1=Divorced, 2=Married, 3=Single)	Norminal	string
MonthlyIncome	How much the employee earn per month	Ratio	int
NumCompaniesWorked	How many companies the employee has worked for	Ratio	int
OverTime	Whether the employee work overtime (0=no, 1=yes)	Norminal	bool
PercentSalaryHike	How much the salary increases	Ratio	int
PerformanceRating	How the performance of the employee has been rated	Ordinal	int
RelationshipSatisfaction	How satisfied the employee with the relations (1=low, 2=medium, 3=high, 4=very high)	Ordinal	int
StockOptionLevel	How much company stocks the employee owns from this company	Ratio	int
TotalWorkingYears	How many years the employee works	Ratio	int
TrainingTimesLastYear	How many hours the employee spent on training last year	Ratio	int
WorkLifeBalance	How the employee balance between work and outside (1=bad, 2=good, 3=better, 4=best)	Ordinal	int
YearsAtCompany	How many years the employee works at this company	Ratio	int
YearsInCurrentRole	How many years the employee works at this position	Ratio	int
YearsSinceLastPromotion	How many years takes for the employee since the last promotion	Ratio	int
YearsWithCurrManager	How many years the employee works with the same manager	Ratio	int

3. Data cleaning

1) **All the cleaning and checking are based on Python3.**
2) **Check the data_scale first.**

```
import pandas as pd
import numpy as np
##eliminate warnings
pd.options.mode.chained_assignment = None

df=pd.read_csv('1.csv')
df.info()
```

3) **Remove the following columns: 'HourlyRate', 'DailyRate', 'MonthlyRate', 'EmployeeCount', 'StandardHours', and 'Over18'.**

```
pd.set_option('display.max_columns', None)
data=df.drop(columns=['HourlyRate','DailyRate','MonthlyRate','EmployeeCount','StandardHours', 'Over18'])
data.head()
data.shape
data.info()
```

4) **Check whether there are unknown values in the dataset and find no NA.**

```
data.isnull().sum().sort_values()
```

5) **Check whether there are duplicated data in that frame and find there are no duplications.**

```
data.duplicated().any()
```

6) **Identify the possible results of columns in the object.**

```
data['Attrition'].unique()
```

```
array(['Yes', 'No'], dtype=object)
```

```
data['BusinessTravel'].unique()
```

```
array(['Travel_Rarely', 'Travel_Frequently', 'Non-Travel'], dtype=object)
```

```
data['Department'].unique()
```

```
array(['Sales', 'Research & Development', 'Human Resources'], dtype=object)
```

```
data['EducationField'].unique()
```

```
array(['Life Sciences', 'Other', 'Medical', 'Marketing',
       'Technical Degree', 'Human Resources'], dtype=object)
```

```
data['Gender'].unique()
```

```
array(['Female', 'Male'], dtype=object)
```

```
data['JobRole'].unique()
```

```
array(['Sales Executive', 'Research Scientist', 'Laboratory Technician',
       'Manufacturing Director', 'Healthcare Representative', 'Manager',
       'Sales Representative', 'Research Director', 'Human Resources'],
      dtype=object)
```

```
data['MaritalStatus'].unique()
```

```
array(['Single', 'Married', 'Divorced'], dtype=object)
```

```
data['OverTime'].unique()
```

```
array(['Yes', 'No'], dtype=object)
```

7) Re-order the dataset and put 'Attrition' as the first column.

```
data_Attrition=data.Attrition
data=data.drop(columns=['Attrition'])
data.insert(0,'Attrition',data_Attrition)
pd.set_option('display.max_columns', None)
data.head()
```

8) Separately change all the objects into int64 based on the dictionary.

```
data.replace({"Attrition":{'Yes':1,'No':0}},inplace=True)
data.replace({"BusinessTravel":{'Non-Travel':1,'Travel_Frequently':2,'Travel_Rarely':3}},inplace=True)
data.replace({"Department":{'Human Resources':1,'Research & Development':2,'Sales':3}},inplace=True)
data.replace({"EducationField":{'Human Resources':1,'Life Sciences':2,'Marketing':3,
                                'Medical':4,'Other':5,'Technical Degree':6}},inplace=True)
data.replace({"Gender":{'Female':1,'Male':0}},inplace=True)
data.replace({"JobRole":{'Healthcare Representative':1,'Human Resources':2,'Laboratory Technician':3,
                                'Manager':4,'Manufacturing Director':5,'Research Director':6,
                         'Research Scientist':7,'Sales Executive':8,'Sales Representative':9}},inplace=True)
data.replace({"MaritalStatus":{'Divorced':1,'Married':2,'Single':3}},inplace=True)
data.replace({"OverTime":{'Yes':1,'No':0}},inplace=True)
data.info()

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
data[:10]
```

4. Data analysis

1) Deeply understand the descriptive statistical features for one int64 and one object.

Central tendency and dispersion

```
df2['Age'].describe()
```

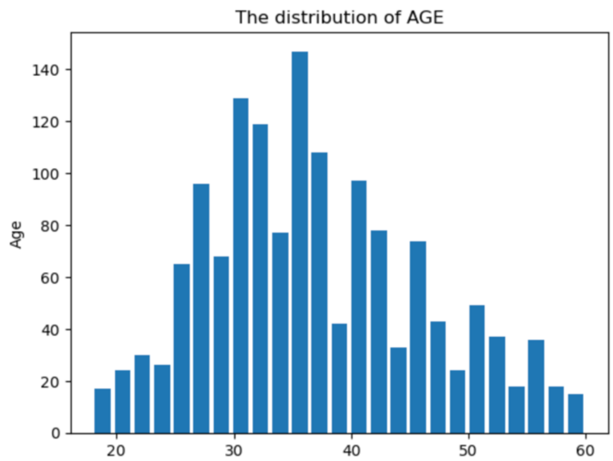
```
count    1470.000000
mean       36.923810
std         9.135373
min        18.000000
25%        30.000000
50%        36.000000
75%        43.000000
max        60.000000
Name: Age, dtype: float64
```

```
df1['Department'].describe()
```

```
count                       1470
unique                         3
top       Research & Development
freq                         961
Name: Department, dtype: object
```

2) Create the histogram for 'Age'.

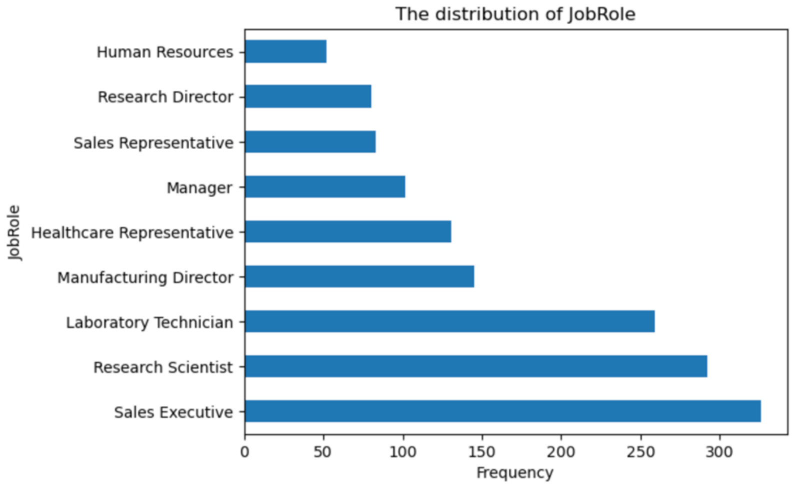
```
import matplotlib.pyplot as plt

plt.hist(df2['Age'], bins=25, rwidth=0.8)
plt.xlabel('Frequency')
plt.ylabel('Age')
plt.title('The distribution of AGE')
plt.show()
```



3) Create the bar chart for 'JobRole'.

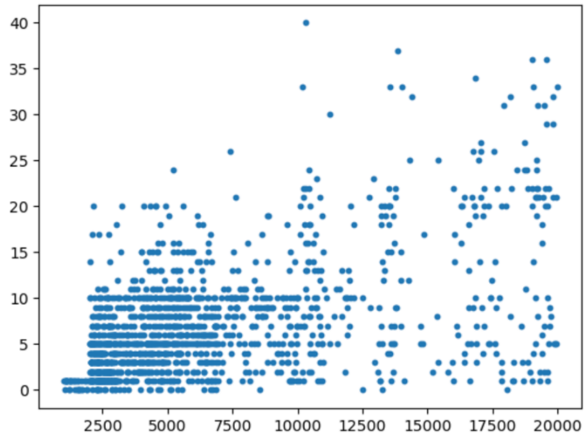
```
JobRole_freq=df1['JobRole'].value_counts()
ax = JobRole_freq.plot.barh(title='The distribution of JobRole')
ax.set_xlabel('Frequency')
ax.set_ylabel('JobRole')
```

```
Text(0, 0.5, 'JobRole')
```



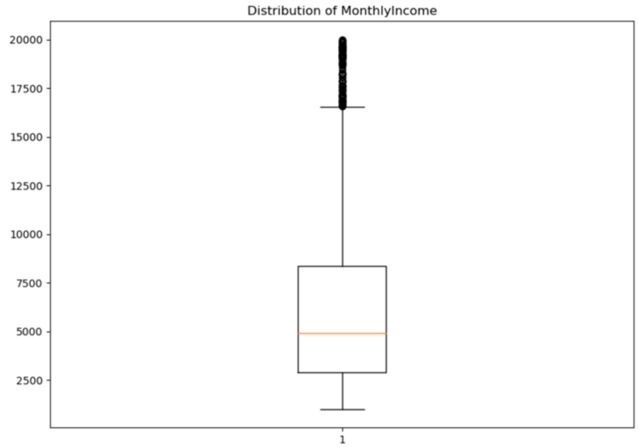
4) Create the scatter plot for 'MonthlyIncome' and 'YearsAtCompany'.

```
plt.scatter(df2['MonthlyIncome'], df2['YearsAtCompany'], s=10)
plt.show()
```



5) Create the box plot for 'MonthlyIncome'.

```
fig = plt.figure(figsize =(10, 7))
plt.boxplot(df2['MonthlyIncome'])
plt.title('Distribution of MonthlyIncome')
plt.show()
```



6) Calculate the correlations between 'MonthlyIncome' and 'YearsAtCompany'.

```
from scipy import stats

data = df2[['MonthlyIncome','YearsAtCompany']]

monthlyincome = data['MonthlyIncome']
yearsatcompany = data['YearsAtCompany']

print(stats.pearsonr(monthlyincome, yearsatcompany))
```

```
PearsonRResult(statistic=0.5142848257331963, pvalue=4.819313789734122e-100)
```

Reference

PAVANSUBHASH. (n.d.). IBM HR analytics employee attrition & performance. Retrieved March 11, 2023, from www.kaggle.com website: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

Teamwork

1. Dataset pros and cons analysis

Data set name	Advantages	Disadvantages
IBM HR Analytics Employee Attrition & Performance	● The dataset is clearly result oriented, and for employee departures, it is clearly divided into two types of departures and non-departures. This helps the accuracy of the results of the data analysis, and it is easier for companies to use the results of this analysis.	● The authenticity of this dataset is yet to be verified, and the group believes that the dataset is more likely to be simulated based on real data. Therefore, the results obtained from the analysis of this dataset may not be completely accurate for the real scenario.

	● This dataset is very uniform in terms of semantics and there is less accidental bias, so using this data for the analysis of why employees leave gives more accurate results. ● The broad coverage of the features of this dataset provides reference factors that can be used to analyse the risk of employees' possible departure in multiple dimensions, which is more reliable in practical use.	● This dataset has a more fixed definition of employees by department, and this definition makes the results derived from this dataset unsuitable for analysing more complex departments, such as operations.
Sydney Suburbs Reviews	● The data set fits well with the theme. For analysts, the dataset provides very comprehensive information on housing in the Sydney area. ● The dataset has multiple, complete quantified scores, and the elements in this dataset are diverse. Analysts can analyse through different scoring systems and are able to provide more detailed analysis models.	● The volume of data does not meet the conditional requirements for future analysis. ● The classification results of this dataset are multivariate. Therefore, the analysis results obtained do not provide accurate help to people who need the house. ● The amount of empty data in the dataset is large, and the empty data is difficult to fill with conventional methods Therefore users can only access analytics services for specific regions, they cannot analyse and query all Sydney regions.

2. Future dataset selection and reasons

After discussion, the group decided to choose the data 'IBM HR ANALYTICS EMPLOYEE ATTRITION & PERFORMANCE' as the future dataset for the following main reasons.

- The dataset strictly matches the requirements of this project for the conditions of the dataset.
- We believe studying employee attrition is more unique and innovative than predicting housing rental prices in Sydney and is a disciplinary and industry convergence for the long-term development of data science integrating management.
- This dataset is more comprehensive than the Sydney housing data in terms of final classification results and feature coverage, so the results obtained from the analysis with this dataset have more stability and validity.