**Machine Learning Lab**

**22pw28**

**Prathoshini Devi M B**

**Exercise: 1**

Explore pandas library using the following the dataset

https://drive.google.com/file/d/1Yzw9jJUpbB6ea24RQEoRzceWOgPQHidc/view?usp=drive\_link

* Read and load it onto a DataFrame.

**import pandas as pd**

**df=pd.read\_csv("/home/cslinux/Downloads/IPL IMB381IPL2013.csv")**

**data=pd1.DataFrame(df)**

**print(data)**

* use pandas features such as selecting, filtering, aggregating, joining, slicing/dicing of data to prepare and explore the dataset.

import pandas as pd

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**# Read the CSV file**

**df = pd.read\_csv(r"C:\Users\prath\Downloads\IPL IMB381IPL2013.csv")**

**# Display the first few rows of the dataframe**

**print(df.head())**

**# Convert 'AGE' to categorical variable**

**df['AGE'] = pd.Categorical(df['AGE'])**

**# Selecting specific columns**

**df\_selected = df[['PLAYER NAME', 'AGE', 'PLAYING ROLE', 'SOLD PRICE']]**

**print(df\_selected.head())**

**# Filtering rows where AGE is 25**

**df\_age\_25 = df[df['AGE'] == 25]**

**print(df\_age\_25)**

**# Grouping by AGE and calculating the mean SOLD PRICE**

**mean\_sold\_price\_by\_age = df.groupby('AGE')['SOLD PRICE'].mean().reset\_index()**

**mean\_sold\_price\_by\_age.columns = ['AGE', 'AVG\_SOLD\_PRICE\_BY\_AGE']**

**print(mean\_sold\_price\_by\_age)**

**# Grouping by AGE and PLAYING ROLE and calculating the mean SOLD PRICE**

**mean\_sold\_price\_by\_age\_role = df.groupby(['AGE', 'PLAYING ROLE'])['SOLD PRICE'].mean().reset\_index()**

**mean\_sold\_price\_by\_age\_role.columns = ['AGE', 'PLAYING\_ROLE', 'AVG\_SOLD\_PRICE\_BY\_AGE\_ROLE']**

**print(mean\_sold\_price\_by\_age\_role)**

**# Merge the two DataFrames on 'AGE' using an inner join**

**merged\_df = pd.merge(mean\_sold\_price\_by\_age\_role, mean\_sold\_price\_by\_age, on='AGE', how='inner')**

**print(merged\_df)**

**# Calculate the change (premium) in percentage terms**

**merged\_df['PREMIUM'] = ((merged\_df['AVG\_SOLD\_PRICE\_BY\_AGE\_ROLE'] - merged\_df['AVG\_SOLD\_PRICE\_BY\_AGE']) / merged\_df['AVG\_SOLD\_PRICE\_BY\_AGE']) \* 100**

**# Rename columns for clarity**

**merged\_df.columns = ['AGE', 'PLAYING\_ROLE', 'AVG\_SOLD\_PRICE\_BY\_AGE\_ROLE', 'AVG\_SOLD\_PRICE\_BY\_AGE', 'PREMIUM']**

**print(merged\_df)**

**# Slicing the first 10 rows and specific columns**

**sliced\_df = df.loc[:9, ['PLAYER NAME', 'AGE', 'SOLD PRICE']]**

**print(sliced\_df)**

**# Set the plot style**

**sns.set(style="whitegrid")**

**# Bar Chart: Average sold price by age**

**plt.figure(figsize=(10, 6))**

**sns.barplot(x='AGE', y='AVG\_SOLD\_PRICE\_BY\_AGE', data=mean\_sold\_price\_by\_age)**

**plt.title('Average Sold Price by Age')**

**plt.xlabel('Age')**

**plt.ylabel('Average Sold Price')**

**plt.show()**

**# Scatter Plot: Age vs. Sold Price**

**plt.figure(figsize=(10, 6))**

**sns.scatterplot(x='AGE', y='SOLD PRICE', data=df)**

**plt.title('Age vs. Sold Price')**

**plt.xlabel('Age')**

**plt.ylabel('Sold Price')**

**plt.show()**

**# Histogram: Distribution of Sold Prices**

**plt.figure(figsize=(10, 6))**

**sns.histplot(df['SOLD PRICE'], bins=30, kde=True)**

**plt.title('Distribution of Sold Prices')**

**plt.xlabel('Sold Price')**

**plt.ylabel('Frequency')**

**plt.show()**

**# Correlation Matrix and Heatmap**

**plt.figure(figsize=(10, 6))**

**correlation\_matrix = df.corr()**

**sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)**

**plt.title('Correlation Matrix Heatmap')**

**plt.show()**

* Displaying First Few Records of the DataFrame

**import pandas as pd**

**df=pd.read\_csv("/home/cslinux/Downloads/IPL IMB381IPL2013.csv")**

**#print(df.to\_string())**

**data=pd.DataFrame(df)**

**print(data)**

**first\_few\_rows=df.head(10)**

**print(first\_few\_rows)**

* Finding Summary of the DataFrame

**import pandas as pd**

**df=pd.read\_csv("/home/cslinux/Downloads/IPL IMB381IPL2013.csv")**

**#print(df.to\_string())**

**data=pd.DataFrame(df)**

**print(data)**

**print(df.describe())**

* Slicing and Indexing of DataFrame

**import pandas as pd**

**df=pd.read\_csv("/home/cslinux/Downloads/IPL IMB381IPL2013.csv")**

**#print(df.to\_string())**

**data=pd.DataFrame(df)**

**print(data)**

**#slicing of first 4 rows**

**df1=df.iloc[0:4]**

**print(df1)**

**#sicing of 2 columns of all rows**

**df2=df.iloc[:,0:2]**

**print(df2)**

**#INdexing based on player name**

**df=pd.read\_csv("/home/cslinux/Downloads/IPL IMB381IPL2013.csv",index\_col="PLAYER NAME")**

**data=pd.DataFrame(df)**

**print(data)**

* Find the occurrences of each unique value in a column.

**import pandas as pd**

**df=pd.read\_csv("Z:\SEM 5\MACHINE LEARNING LAB\Worksheet 1\IPL IMB381IPL2013.csv")**

**#print(df.to\_string())**

**data=pd.DataFrame(df)**

**print(data)**

**print('UNIQUE VALUE AND THEIR COUNT OF PLAYER NAME')**

**print(df['PLAYER NAME'].value\_counts())**

**print('UNIQUE VALUE AND THEIR COUNT OF AGE')**

**print(df['AGE'].value\_counts())**

**print('UNIQUE VALUE AND THEIR COUNT OF AUCTION YEAR')**

**print(df['AUCTION YEAR'].value\_counts())**

**print('UNIQUE VALUE AND THEIR COUNT OF BASE PRICE')**

**print(df['BASE PRICE'].value\_counts())**

**print('UNIQUE VALUE AND THEIR COUNT OF SOLD PRICE')**

**print(df['SOLD PRICE'].value\_counts())**

* Perform cross tabulation between PLAYING ROLE and AGE will give number of players in each age category for each playing role

**import pandas as pd**

**df=pd.read\_csv("Z:\SEM 5\MACHINE LEARNING LAB\Worksheet 1\IPL IMB381IPL2013.csv")**

**ct=pd.crosstab(df['PLAYING ROLE'],df['AGE'])**

**print(ct)**

* Sorting Dataframe by Column Values

**import pandas as pd**

**df=pd.read\_csv("Z:\SEM 5\MACHINE LEARNING LAB\Worksheet 1\IPL IMB381IPL2013.csv")**

**sorted\_data=df.sort\_values(by="AGE",ascending=True)**

**print(sorted\_data[['AGE','PLAYER NAME']])**

* Find average SOLD PRICE for each age category, group all records by AGE and then apply mean() on SOLD PRICE column

**import pandas as pd**

**# Read the CSV file**

**df = pd.read\_csv(r"C:\Users\prath\Downloads\IPL IMB381IPL2013.csv")**

**# Group the data by 'AGE'**

**gb = df.groupby("AGE")**

**# Calculate the mean of the 'SOLD PRICE' for each 'AGE' group**

**mean\_sold\_price = gb["SOLD PRICE"].mean()**

**# Print the 'AGE' and mean 'SOLD PRICE'**

**print(mean\_sold\_price)**

* Perform joining of data frame- compare the average SOLD PRICE for different AGE categories with the different age and PLAYING ROLE categories, we need to merge the DataFrames soldprice\_by\_age and soldprice\_by\_ age\_role. The common column is AGE and this needs outer join. Age is considered as categorical varaibale(as 1,2,3)

import pandas as pd

# Read the CSV file

df = pd.read\_csv(r"C:\Users\prath\Downloads\IPL IMB381IPL2013.csv")

# Convert 'AGE' to categorical variable

df['AGE'] = pd.Categorical(df['AGE'])

# Calculate the average 'SOLD PRICE' by 'AGE'

soldprice\_by\_age = df.groupby('AGE')['SOLD PRICE'].mean().reset\_index()

soldprice\_by\_age.columns = ['AGE', 'AVG\_SOLD\_PRICE\_BY\_AGE']

# Calculate the average 'SOLD PRICE' by 'AGE' and 'PLAYING ROLE'

soldprice\_by\_age\_role = df.groupby(['AGE', 'PLAYING ROLE'])['SOLD PRICE'].mean().reset\_index()

soldprice\_by\_age\_role.columns = ['AGE', 'PLAYING ROLE', 'AVG\_SOLD\_PRICE\_BY\_AGE\_ROLE']

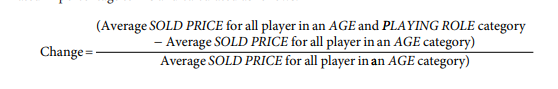
# Merge the two DataFrames on 'AGE' using an outer join

merged\_df = pd.merge(soldprice\_by\_age, soldprice\_by\_age\_role, on='AGE', how='outer')

# Print the merged DataFrame

print(merged\_df)

* To find whether players carry a premium if they belong to a specific AGE and PLAYING ROLE category. The premium (we will call it change) is calculated in percentage terms and calculated as follows



import pandas as pd

# Read the CSV file

df = pd.read\_csv(r"C:\Users\prath\Downloads\IPL IMB381IPL2013.csv")

# Convert 'AGE' to categorical variable

df['AGE'] = pd.Categorical(df['AGE'])

# Calculate the average 'SOLD PRICE' by 'AGE'

soldprice\_by\_age = df.groupby('AGE')['SOLD PRICE'].mean().reset\_index()

soldprice\_by\_age.columns = ['AGE', 'AVG\_SOLD\_PRICE\_BY\_AGE']

# Calculate the average 'SOLD PRICE' by 'AGE' and 'PLAYING ROLE'

soldprice\_by\_age\_role = df.groupby(['AGE', 'PLAYING ROLE'])['SOLD PRICE'].mean().reset\_index()

soldprice\_by\_age\_role.columns = ['AGE', 'PLAYING\_ROLE', 'AVG\_SOLD\_PRICE\_BY\_AGE\_ROLE']

# Merge the two DataFrames on 'AGE' using an outer join

merged\_df = pd.merge(soldprice\_by\_age\_role, soldprice\_by\_age, on='AGE', how='outer')

# Calculate the change (premium) in percentage terms

merged\_df['PREMIUM'] = ((merged\_df['AVG\_SOLD\_PRICE\_BY\_AGE\_ROLE'] - merged\_df['AVG\_SOLD\_PRICE\_BY\_AGE']) / merged\_df['AVG\_SOLD\_PRICE\_BY\_AGE']) \* 100

# Print the merged DataFrame with the premium

print(merged\_df)

* Exploration of data using visualization-Drawing Plots -Matplotlib -Bar Chart, scatter plot Histogram, Correlation and Heatmap.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Read the CSV file

df = pd.read\_csv(r"C:\Users\prath\Downloads\IPL IMB381IPL2013.csv")

# Convert 'AGE' to categorical variable

df['AGE'] = pd.Categorical(df['AGE'])

# Set the plot style

sns.set(style="whitegrid")

# Bar Chart: Average sold price by age

plt.figure(figsize=(10, 6))

avg\_sold\_price\_by\_age = df.groupby('AGE')['SOLD PRICE'].mean().reset\_index()

sns.barplot(x='AGE', y='SOLD PRICE', data=avg\_sold\_price\_by\_age)

plt.title('Average Sold Price by Age')

plt.xlabel('Age')

plt.ylabel('Average Sold Price')

plt.show()

# Scatter Plot: Age vs. Sold Price

plt.figure(figsize=(10, 6))

sns.scatterplot(x='AGE', y='SOLD PRICE', data=df)

plt.title('Age vs. Sold Price')

plt.xlabel('Age')

plt.ylabel('Sold Price')

plt.show()

# Histogram: Distribution of Sold Prices

plt.figure(figsize=(10, 6))

sns.histplot(df['SOLD PRICE'], bins=30, kde=True)

plt.title('Distribution of Sold Prices')

plt.xlabel('Sold Price')

plt.ylabel('Frequency')

plt.show()

# Correlation Matrix and Heatmap

plt.figure(figsize=(10, 6))

correlation\_matrix = df.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)

plt.title('Correlation Matrix Heatmap')

plt.show()