

In [101...]

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(color_codes=True)
```

In [102...]

```
df=pd.read_csv('ds_salaries.csv')
df.head()
```

Out[102]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	er
0	2023	SE	FT	Principal Data Scientist	80000	EUR	85847	
1	2023	MI	CT	ML Engineer	30000	USD	30000	
2	2023	MI	CT	ML Engineer	25500	USD	25500	
3	2023	SE	FT	Data Scientist	175000	USD	175000	
4	2023	SE	FT	Data Scientist	120000	USD	120000	

◀ ▶

## Data Processing Part 1

In [103...]

```
# drop salary columns since salary_in_usd can be used
# drop salary_current to use universal currency i.e usd

df.drop(columns=['salary','salary_currency'], inplace=True)
df.head()
```

Out[103]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remot
0	2023	SE	FT	Principal Data Scientist	85847	ES	
1	2023	MI	CT	ML Engineer	30000	US	
2	2023	MI	CT	ML Engineer	25500	US	
3	2023	SE	FT	Data Scientist	175000	CA	
4	2023	SE	FT	Data Scientist	120000	CA	

◀ ▶

In [104...]

```
#Check for missing values
check_missing=df.isnull().sum()*100/df.shape[0]
```

```
check_missing[check_missing>0].sort_values(ascending=False)
```

```
Out[104]: Series([], dtype: float64)
```

```
In [105... #check no of unique values on object datatype  
df.select_dtypes(include='object').nunique()
```

```
Out[105]: experience_level      4  
employment_type        4  
job_title             93  
employee_residence    78  
company_location       72  
company_size           3  
dtype: int64
```

Catagorise the job title

```
In [106... df.job_title.unique()
```

```
Out[106]: array(['Principal Data Scientist', 'ML Engineer', 'Data Scientist',
   'Applied Scientist', 'Data Analyst', 'Data Modeler',
   'Research Engineer', 'Analytics Engineer',
   'Business Intelligence Engineer', 'Machine Learning Engineer',
   'Data Strategist', 'Data Engineer', 'Computer Vision Engineer',
   'Data Quality Analyst', 'Compliance Data Analyst',
   'Data Architect', 'Applied Machine Learning Engineer',
   'AI Developer', 'Research Scientist', 'Data Analytics Manager',
   'Business Data Analyst', 'Applied Data Scientist',
   'Staff Data Analyst', 'ETL Engineer', 'Data DevOps Engineer',
   'Head of Data', 'Data Science Manager', 'Data Manager',
   'Machine Learning Researcher', 'Big Data Engineer',
   'Data Specialist', 'Lead Data Analyst', 'BI Data Engineer',
   'Director of Data Science', 'Machine Learning Scientist',
   'MLOps Engineer', 'AI Scientist', 'Autonomous Vehicle Technician',
   'Applied Machine Learning Scientist', 'Lead Data Scientist',
   'Cloud Database Engineer', 'Financial Data Analyst',
   'Data Infrastructure Engineer', 'Software Data Engineer',
   'AI Programmer', 'Data Operations Engineer', 'BI Developer',
   'Data Science Lead', 'Deep Learning Researcher', 'BI Analyst',
   'Data Science Consultant', 'Data Analytics Specialist',
   'Machine Learning Infrastructure Engineer', 'BI Data Analyst',
   'Head of Data Science', 'Insight Analyst',
   'Deep Learning Engineer', 'Machine Learning Software Engineer',
   'Big Data Architect', 'Product Data Analyst',
   'Computer Vision Software Engineer', 'Azure Data Engineer',
   'Marketing Data Engineer', 'Data Analytics Lead', 'Data Lead',
   'Data Science Engineer', 'Machine Learning Research Engineer',
   'NLP Engineer', 'Manager Data Management',
   'Machine Learning Developer', '3D Computer Vision Researcher',
   'Principal Machine Learning Engineer', 'Data Analytics Engineer',
   'Data Analytics Consultant', 'Data Management Specialist',
   'Data Science Tech Lead', 'Data Scientist Lead',
   'Cloud Data Engineer', 'Data Operations Analyst',
   'Marketing Data Analyst', 'Power BI Developer',
   'Product Data Scientist', 'Principal Data Architect',
   'Machine Learning Manager', 'Lead Machine Learning Engineer',
   'ETL Developer', 'Cloud Data Architect', 'Lead Data Engineer',
   'Head of Machine Learning', 'Principal Data Analyst',
   'Principal Data Engineer', 'Staff Data Scientist',
   'Finance Data Analyst'], dtype=object)
```

In [107...]

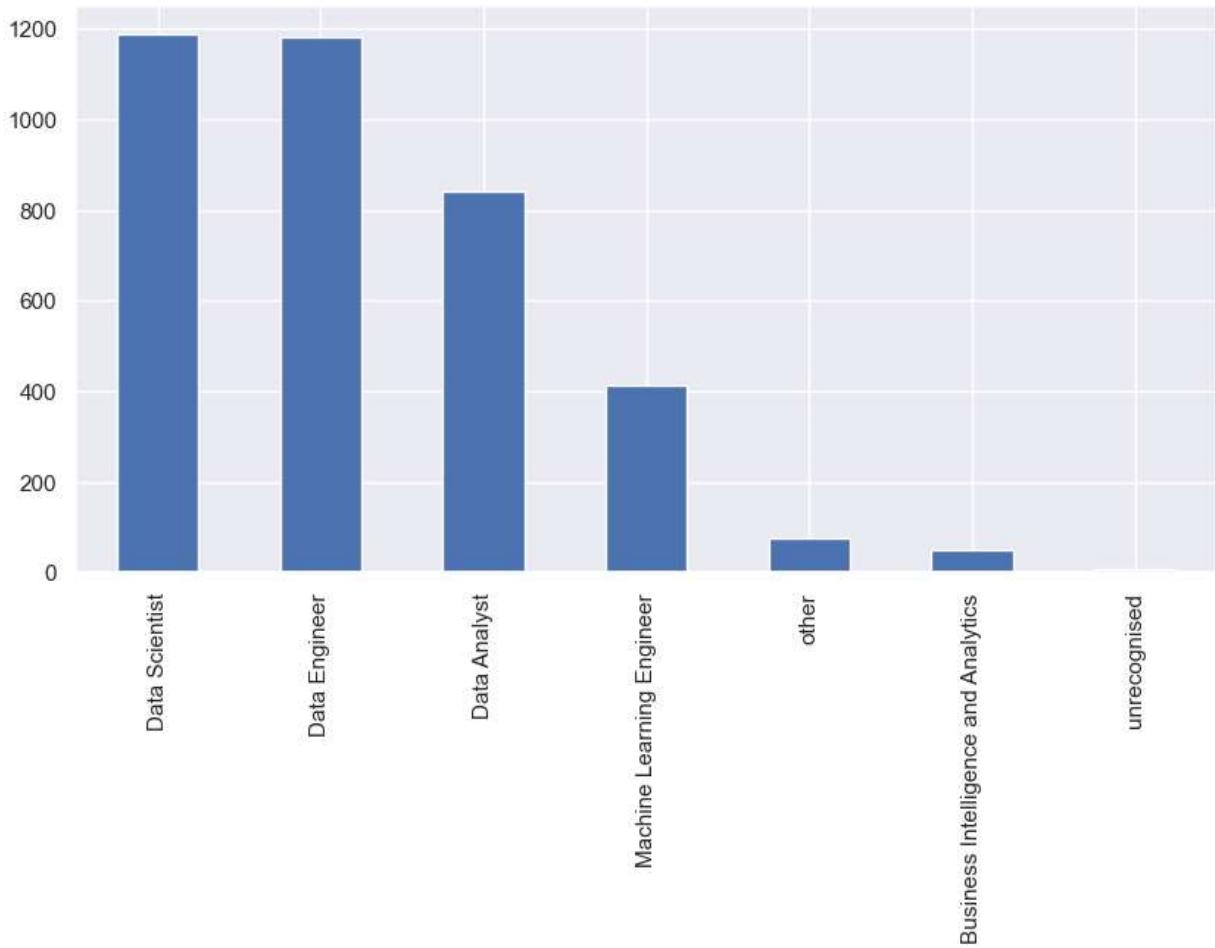
```
def seg_job_titles(job_title):
    data_scientist_titles=['Principal Data Scientist','Data Scientist','Applied Data S
                           'Head of Data', 'Data Manager', 'Data Specialist', 'Director o
                           'Head of Data Science', 'NLP Engineer', 'Data Science Tech
                           'Product Data Scientist',
                           'Staff Data Scientist', 'Deep Learning Researcher', 'Deep Lear
                           'Computer Vision Software Engineer',
                           'Lead Data Scientist', 'Research Scientist', 'Applied Scienti
    machine_learning_titles=[ 'Machine Learning Engineer', 'Applied Machine Learning Eng
                           'Machine Learning Scientist', 'MLOps Engineer', 'Applied Mac
                           'Machine Learning Infrastructure Engineer', 'Machine Learnin
                           'Machine Learning Research Engineer', 'Machine Learning Dev
                           'Lead Machine Learning Engineer', 'ML Engineer',
                           'Principal Machine Learning Engineer', 'Head of Machine Lea
    data_analyst_titles=[ 'Data Analyst', 'Data Modeler', 'Analytics Engineer', 'Data Qua
                           'Compliance Data Analyst', 'Data Analytics Manager', 'Busines
                           'Lead Data Analyst', 'Financial Data Analyst', 'Data Science C
                           'BI Data Analyst', 'Product Data Analyst', 'Data Analytics Lead
```

```
'Data Analytics Consultant', 'Marketing Data Analyst', 'Data C
    'Insight Analyst', 'Finance Data Analyst', 'BI Analyst' ]
data_engineer_titles=['Data Strategist', 'Data Engineer', 'Data Architect', 'Big Dat
        'Software Data Engineer', 'Marketing Data Engineer', 'Cloud D
        'Principal Data Engineer', 'Data Science Engineer', 'Princip
        'Data Operations Engineer'
    ]
bi_analytics_titles=['BI Developer', 'Azure Data Engineer', 'Manager Data Management
        'AI Developer', 'ETL Engineer', 'Data DevOps Engineer', 'AI Sci
other_titles=['Research Engineer',
    'Power BI Developer',
    'ETL Developer', 'Cloud Data Architect', 'Big Data Engineer', '3D Computer Vision
    'BI Data Engineer', 'Autonomous Vehicle Technician', 'Cloud Database Engineer', '
]
if job_title in data_scientist_titles:
    return 'Data Scientist'
elif job_title in machine_learing_titles:
    return 'Machine Learning Engineer'
elif job_title in data_analyst_titles:
    return 'Data Analyst'
elif job_title in data_engineer_titles:
    return 'Data Engineer'
elif job_title in bi_analytics_titles:
    return 'Business Intelligence and Analytics'
elif job_title in other_titles:
    return 'other'
else:
    return 'unrecognised'
```

In [108]: df['job\_title']=df['job\_title'].apply(seg\_job\_titles)

In [109]: plt.figure(figsize=(10,5))
df['job\_title'].value\_counts().plot(kind='bar')

Out[109]: <AxesSubplot:>



### Categorize the Employee Residence

```
In [110]: df.employee_residence.unique()
```

```
Out[110]: array(['ES', 'US', 'CA', 'DE', 'GB', 'NG', 'IN', 'HK', 'PT', 'NL', 'CH',
       'CF', 'FR', 'AU', 'FI', 'UA', 'IE', 'IL', 'GH', 'AT', 'CO', 'SG',
       'SE', 'SI', 'MX', 'UZ', 'BR', 'TH', 'HR', 'PL', 'KW', 'VN', 'CY',
       'AR', 'AM', 'BA', 'KE', 'GR', 'MK', 'LV', 'RO', 'PK', 'IT', 'MA',
       'LT', 'BE', 'AS', 'IR', 'HU', 'SK', 'CN', 'CZ', 'CR', 'TR', 'CL',
       'PR', 'DK', 'BO', 'PH', 'DO', 'EG', 'ID', 'AE', 'MY', 'JP', 'EE',
       'HN', 'TN', 'RU', 'DZ', 'IQ', 'BG', 'JE', 'RS', 'NZ', 'MD', 'LU',
       'MT'], dtype=object)
```

```
In [111]: #Define a function to categorize the unique values
```

```
def categorize_region(country):
    if country in['DE', 'GB', 'PT', 'NL', 'CH', 'CF', 'FR', 'FI', 'UA', 'IE', 'AT', 'GR', 'PL', 'HU']:
        return 'Europe'
    elif country in['US', 'CA', 'MX']:
        return 'North America'
    elif country in['BR', 'AR', 'CL', 'BO', 'CR', 'DO', 'PR', 'HN', 'UY']:
        return 'South America'
    elif country in['NG', 'GH', 'KE', 'TN', 'DZ']:
        return 'Africa'
    elif country in['HK', 'IN', 'CN', 'JP', 'KR', 'BD', 'VN', 'PH', 'MY', 'ID', 'AE', 'PK', 'SK', 'SI', 'AM', 'MA', 'BA']:
        return 'Asia'
    elif country in['AU', 'NZ']:
        return 'Oceania'
    else:
        return 'Other'
```

```

        return 'Oceania'
    else:
        return 'unknown'

```

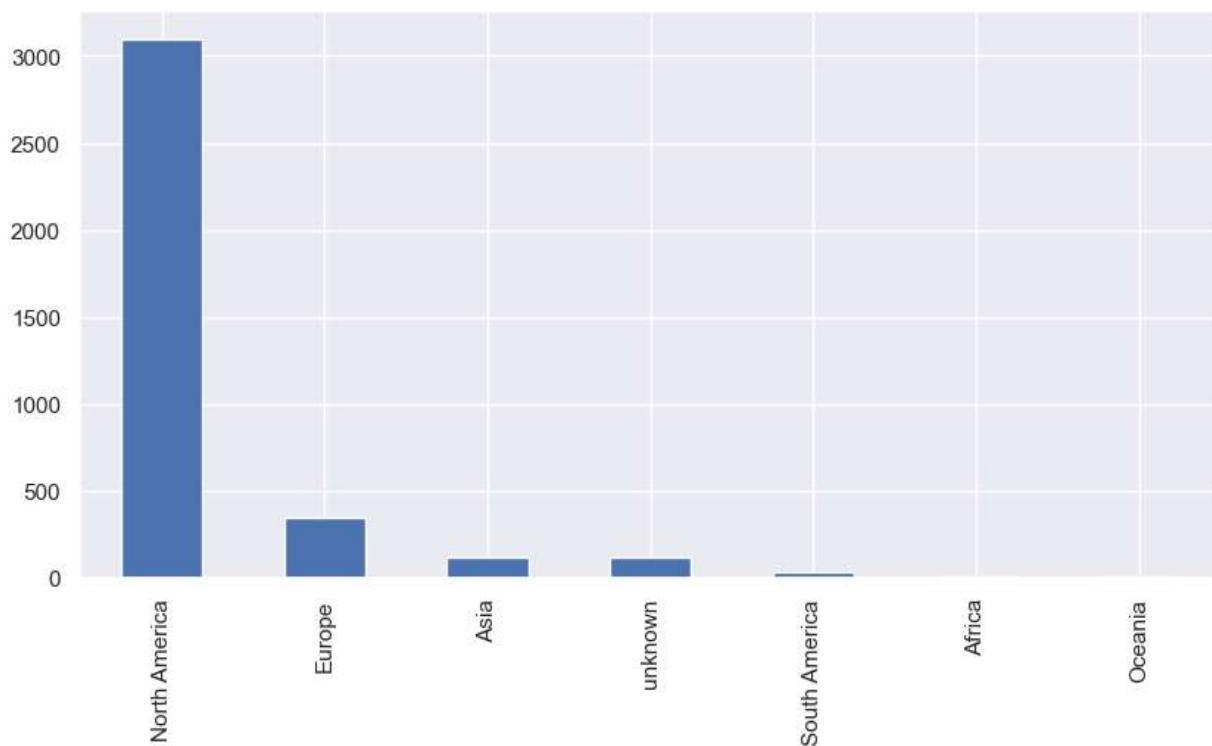
In [112]:

```
# Apply this function to the employee_residence columns to update with new Locations
df['employee_residence']=df['employee_residence'].apply(catagorize_region)
```

In [113]:

```
plt.figure(figsize=(10,5))
df['employee_residence'].value_counts().plot(kind='bar')
```

Out[113]:



In [114]:

```
df.company_location.unique()
```

Out[114]:

```
array(['ES', 'US', 'CA', 'DE', 'GB', 'NG', 'IN', 'HK', 'NL', 'CH', 'CF',
       'FR', 'FI', 'UA', 'IE', 'IL', 'GH', 'CO', 'SG', 'AU', 'SE', 'SI',
       'MX', 'BR', 'PT', 'RU', 'TH', 'HR', 'VN', 'EE', 'AM', 'BA', 'KE',
       'GR', 'MK', 'LV', 'RO', 'PK', 'IT', 'MA', 'PL', 'AL', 'AR', 'LT',
       'AS', 'CR', 'IR', 'BS', 'HU', 'AT', 'SK', 'CZ', 'TR', 'PR', 'DK',
       'BO', 'PH', 'BE', 'ID', 'EG', 'AE', 'LU', 'MY', 'HN', 'JP', 'DZ',
       'IQ', 'CN', 'NZ', 'CL', 'MD', 'MT'], dtype=object)
```

In [115]:

```
def catagorize_region(country):
    if country in['DE', 'GB', 'PT', 'NL', 'CH', 'CF', 'FR', 'FI', 'UA', 'IE', 'AT', 'GR', 'PL', 'HU']:
        return 'Europe'
    elif country in['US', 'CA', 'MX']:
        return 'North America'
    elif country in['BR', 'AR', 'CL', 'BO', 'CR', 'DO', 'PR', 'HN', 'UY']:
        return 'South America'
    elif country in['NG', 'GH', 'KE', 'TN', 'DZ']:
        return 'Oceania'
    else:
        return 'unknown'
```

```

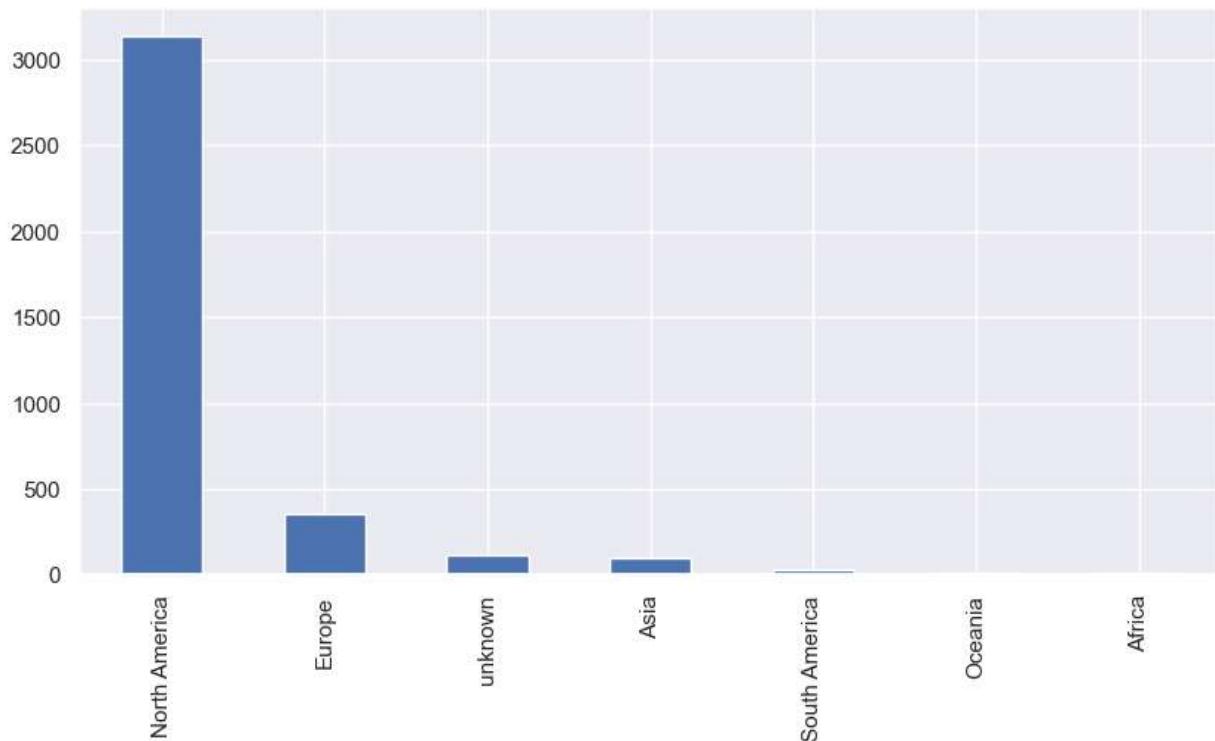
        return 'Africa'
    elif country in['HK', 'IN', 'CN', 'JP', 'KR', 'BD', 'VN', 'PH', 'MY', 'ID', 'AE', 'PK', 'SK', 'SI', 'AM', 'MA', 'BA']:
        return 'Asia'
    elif country in['AU', 'NZ']:
        return 'Oceania'
    else:
        return 'unknown'

```

In [116]: # Apply this function to the company\_location columns to update with new locations  
df['company\_location']=df['company\_location'].apply(catagorize\_region)

In [117]: plt.figure(figsize=(10,5))  
df['company\_location'].value\_counts().plot(kind='bar')

Out[117]: <AxesSubplot:>



In [118]: #Check no of unique values on object datatype  
df.select\_dtypes(include='object').nunique()

Out[118]:

experience_level	4
employment_type	4
job_title	7
employee_residence	7
company_location	7
company_size	3

dtype: int64

## EXPLORATORY DATA ANALYSIS

In [119]: df.remote\_ratio.unique()

```
Out[119]: array([100,    0,   50], dtype=int64)
```

In [120...]

```
#List of categorical variables to plot
cat_vars=['experience_level','employment_type','job_title','company_location','company'
#create figure with subplots

fig,axs=plt.subplots(nrows=2,ncols=4,figsize=(20,10))
axs=axs.flatten()

#create barplot for each categorical variable

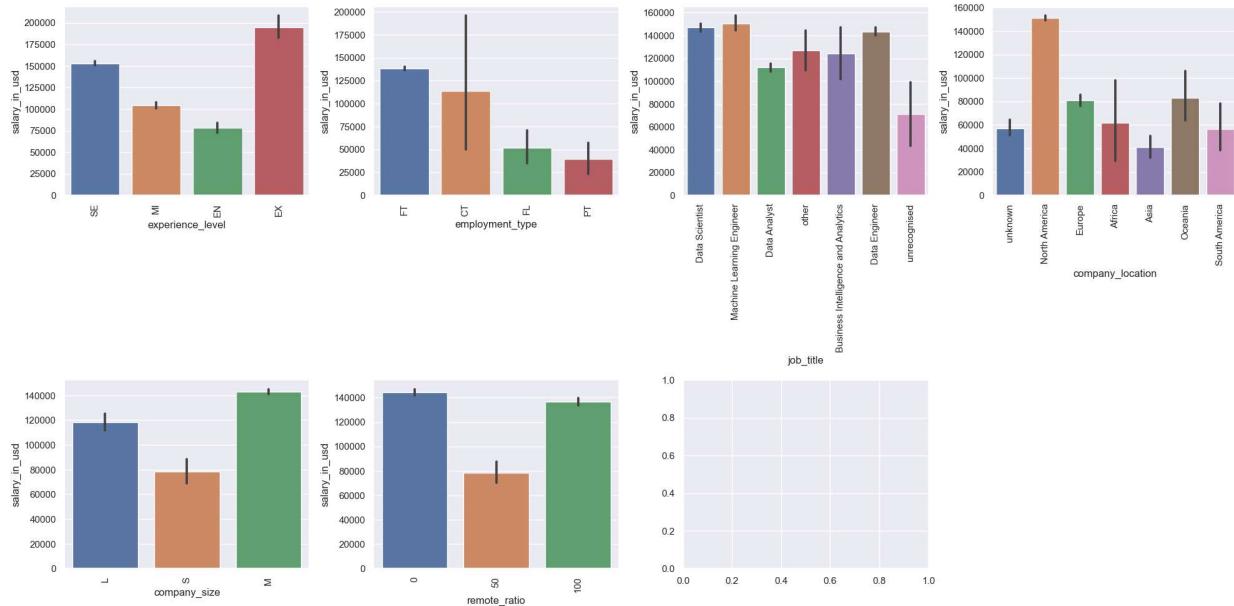
for i, var in enumerate(cat_vars):
    sns.barplot(x=var,y='salary_in_usd',data=df,ax=axs[i],estimator=np.mean)
    axs[i].set_xticklabels(axs[i].get_xticklabels(),rotation=90)

#remove 8th subplot

fig.delaxes(axs[7])
#adjust spacing between subplots

fig.tight_layout()

#show plot
plt.show()
```



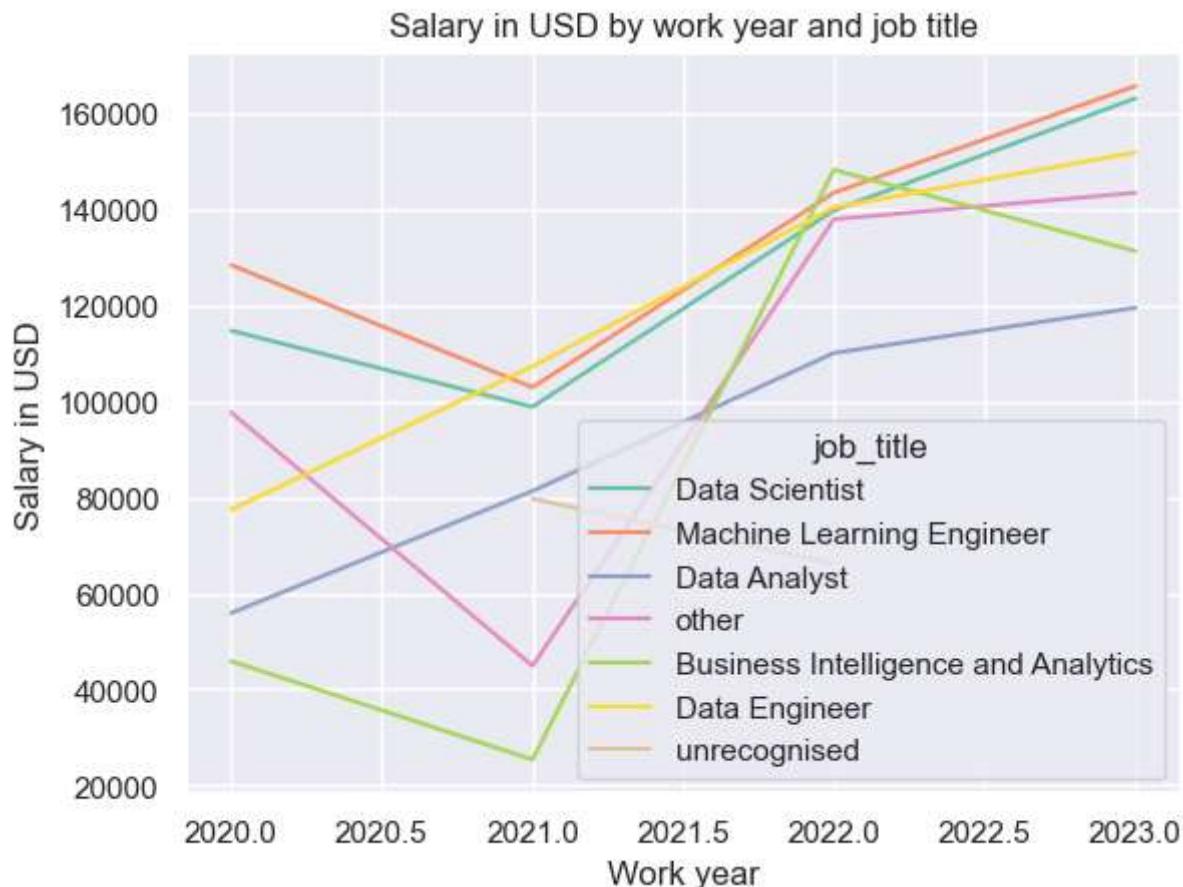
In [121...]

```
sns.set_style('darkgrid')
sns.set_palette('Set2')

sns.lineplot(x='work_year',y='salary_in_usd',hue='job_title',data=df,ci=None)

plt.title('Salary in USD by work year and job title')
plt.xlabel('Work year')
plt.ylabel('Salary in USD')

plt.show()
```



## DATA PROCESSING PART 2

### LABEL ENCODING FOR OBJECT DATATYPE

```
In [122...]: # Loop over each column in the DataFrame where dtype is 'object'
for col in df.select_dtypes(include=['object']).columns:
    #print the column name and unique values
    print(f'{col}: {df[col].unique()}')


experience_level: ['SE' 'MI' 'EN' 'EX']
employment_type: ['FT' 'CT' 'FL' 'PT']
job_title: ['Data Scientist' 'Machine Learning Engineer' 'Data Analyst' 'other'
 'Business Intelligence and Analytics' 'Data Engineer' 'unrecognised']
employee_residence: ['unknown' 'North America' 'Europe' 'Africa' 'Asia' 'Oceania'
 'South America']
company_location: ['unknown' 'North America' 'Europe' 'Africa' 'Asia' 'Oceania'
 'South America']
company_size: ['L' 'S' 'M']
```

In [ ]:

In [ ]:

```
In [123...]: from sklearn import preprocessing
#Loop over each coulmn in DataFrame where dtype is object
for col in df.select_dtypes(include=['object']).columns:
    #Initialize the label encoder
    label_encoder=preprocessing.LabelEncoder()
```

```
#Fit the encoder to the unique value in the column  
label_encoder.fit(df[col].unique())  
  
#Transform the column using encoder  
df[col]=label_encoder.transform(df[col])  
  
#Print the column name and unique encoded values  
print(f"{col}:{df[col].unique()}")
```

experience\_level:[3 2 0 1]  
employment\_type:[2 0 1 3]  
job\_title:[3 4 1 5 0 2 6]  
employee\_residence:[6 3 2 0 1 4 5]  
company\_location:[6 3 2 0 1 4 5]  
company\_size:[0 2 1]

In [124]: df.dtypes

Out[124]:

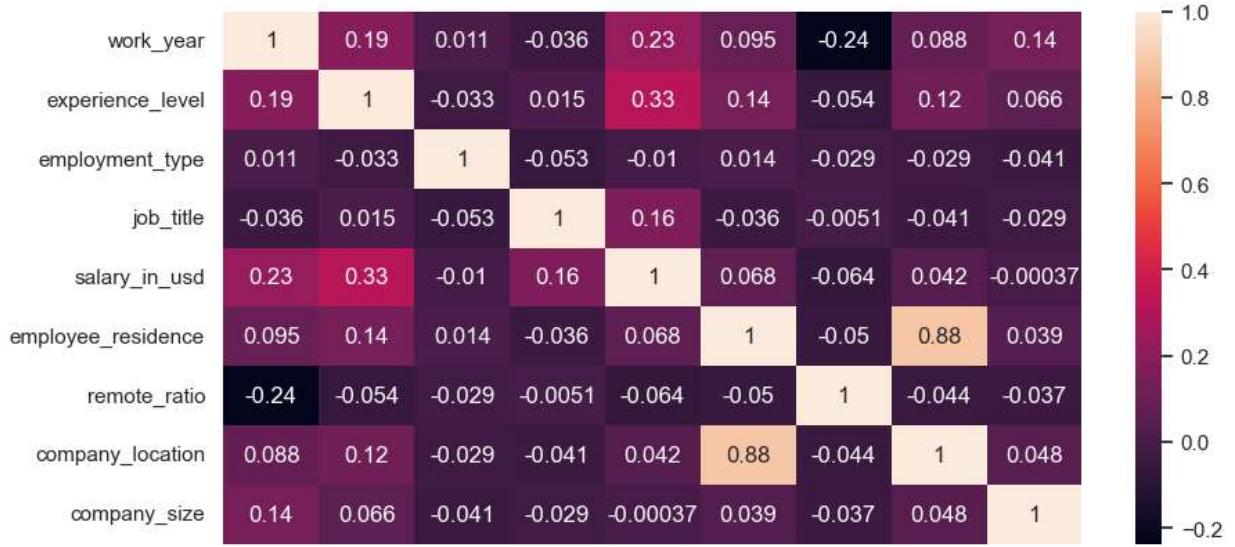
work_year	int64
experience_level	int32
employment_type	int32
job_title	int32
salary_in_usd	int64
employee_residence	int32
remote_ratio	int64
company_location	int32
company_size	int32
dtype:	object

ALL THE DATA IS CATEGORIAL SO THAT MEANS THERE ARE NO OUTLIERS

In [125]:

```
#CORELATION HEATMAP  
plt.figure(figsize=(10,5))  
sns.heatmap(df.corr(), fmt=' .2g ', annot=True)
```

Out[125]: <AxesSubplot:>



## TRAIN TEST SPLITS

```
In [126...]: X=df.drop('salary_in_usd', axis=1)
y=df['salary_in_usd']
```

```
In [127...]: #test size 20% and train size 80%
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

## DECISION TREE REGRESSOR

```
In [133...]: pip install -U scikit-learn
```

Requirement already satisfied: scikit-learn in c:\users\kiran\anaconda3\lib\site-packages (1.2.2)  
Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: joblib>=1.1.1 in c:\users\kiran\anaconda3\lib\site-packages (from scikit-learn) (1.2.0)  
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\kiran\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)  
Requirement already satisfied: numpy>=1.17.3 in c:\users\kiran\anaconda3\lib\site-packages (from scikit-learn) (1.21.5)  
Requirement already satisfied: scipy>=1.3.2 in c:\users\kiran\anaconda3\lib\site-packages (from scikit-learn) (1.9.1)

```
In [143...]: from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import GridSearchCV
data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22, header=None)
data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
target = raw_df.values[1::2, 2]
```

```
# Create a Decision Tree Regressor Object

dtree=DecisionTreeRegressor()

#Define the hyperparamaters to tune and thier values

param_grid={

    'max_depth':[2,4,6,8],
    'min_samples_split':[2,4,6,8],
    'min_samples_leaf':[1,2,3,4],
    'max_features':['auto','sqrt','log2']

}

#Create a Grid Search CV object

grid_search=GridSearchCV(dtree,param_grid,cv=5,scoring='neg_mean_squared_error')

#Fit the GridSearchCV object to the data
grid_search.fit(X_train,y_train)

#print the best hyperparameters
print(grid_search.best_params_)
```



```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
```



```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
```

```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
```

```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
```



```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
```



```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
```



```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
```



```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
```

```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
```











```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
```

```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_depth': 8, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 8}
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
```

In [145...]

```
from sklearn.tree import DecisionTreeRegressor
dtree=DecisionTreeRegressor(random_state=0,max_depth=8,max_features='auto',min_samples_leaf=2,min_samples_split=8)
dtree.fit(X_train,y_train)
```

```
C:\Users\KIRAN\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0``.
    warnings.warn(
```

Out[145]:

```
▼ DecisionTreeRegressor
DecisionTreeRegressor(max_depth=8, max_features='auto', min_samples_leaf=2,
                      min_samples_split=8, random_state=0)
```

In [146...]

```
from sklearn import metrics
from sklearn.metrics import mean_absolute_percentage_error
import math
y_pred=dtree.predict(X_test)
mae=metrics.mean_absolute_error(y_test,y_pred)
mape=mean_absolute_percentage_error(y_test,y_pred)
mse=metrics.mean_squared_error(y_test,y_pred)
r2=metrics.r2_score(y_test,y_pred)
rmse=math.sqrt(mse)

print('MAE is {}'.format(mae))
print('MAPE is {}'.format(mape))
print('MSE is {}'.format(mse))
print('R2 is {}'.format(r2))
print('RMSE is {}'.format(rmse))
```

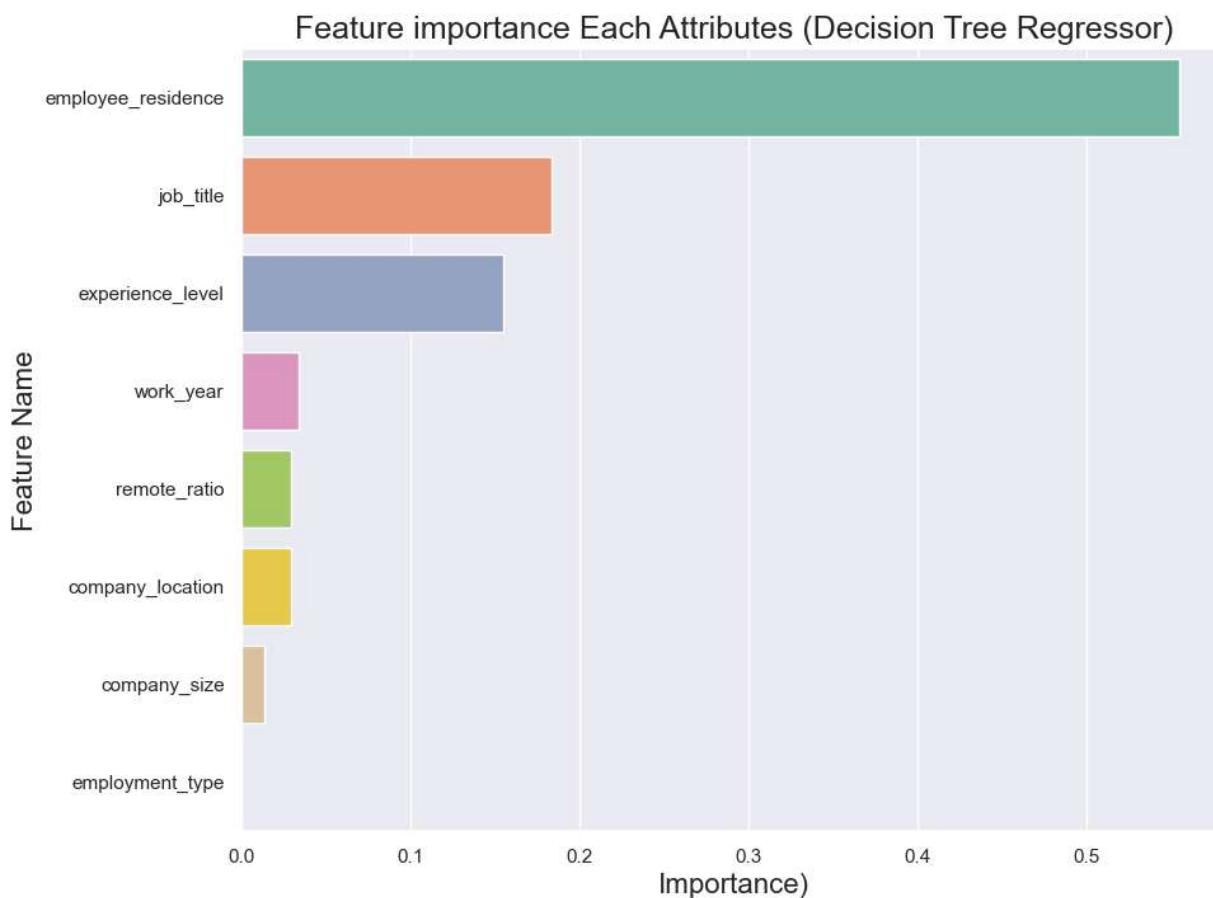
MAE is 39291.46248134226  
 MAPE is 0.35560405550569074  
 MSE is 2853754264.296422  
 R2 is 0.33113289854967665  
 RMSE is 53420.54159493726

In [149...]

```
imp_df=pd.DataFrame({
    'Feature Name':X_train.columns,
    'Importance':dtree.feature_importances_
})

fi=imp_df.sort_values(by='Importance',ascending=False)

fi2=fi.head(10)
plt.figure(figsize=(10,8))
sns.barplot(data=fi2,x='Importance',y='Feature Name')
plt.title('Feature importance Each Attributes (Decision Tree Regressor)',fontsize=18)
plt.xlabel('Importance',fontsize=16)
plt.ylabel('Feature Name',fontsize=16)
plt.show()
```



In [151...]

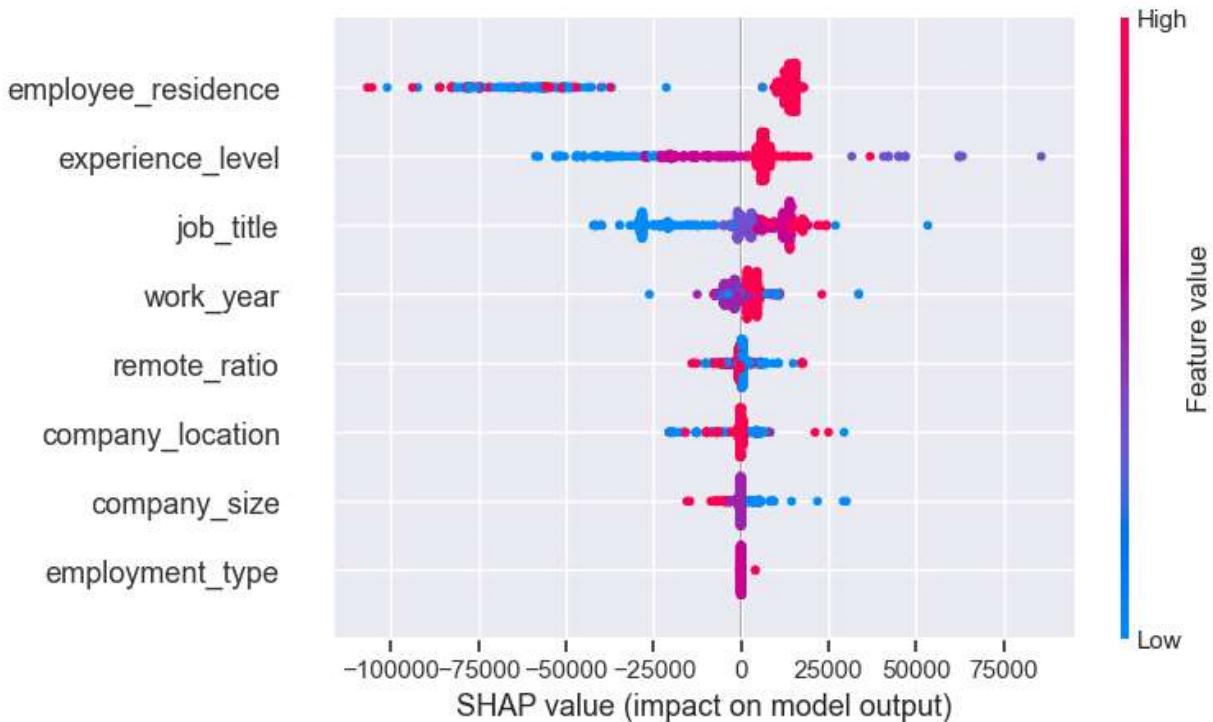
```
pip install shap
```

Collecting shapNote: you may need to restart the kernel to use updated packages.

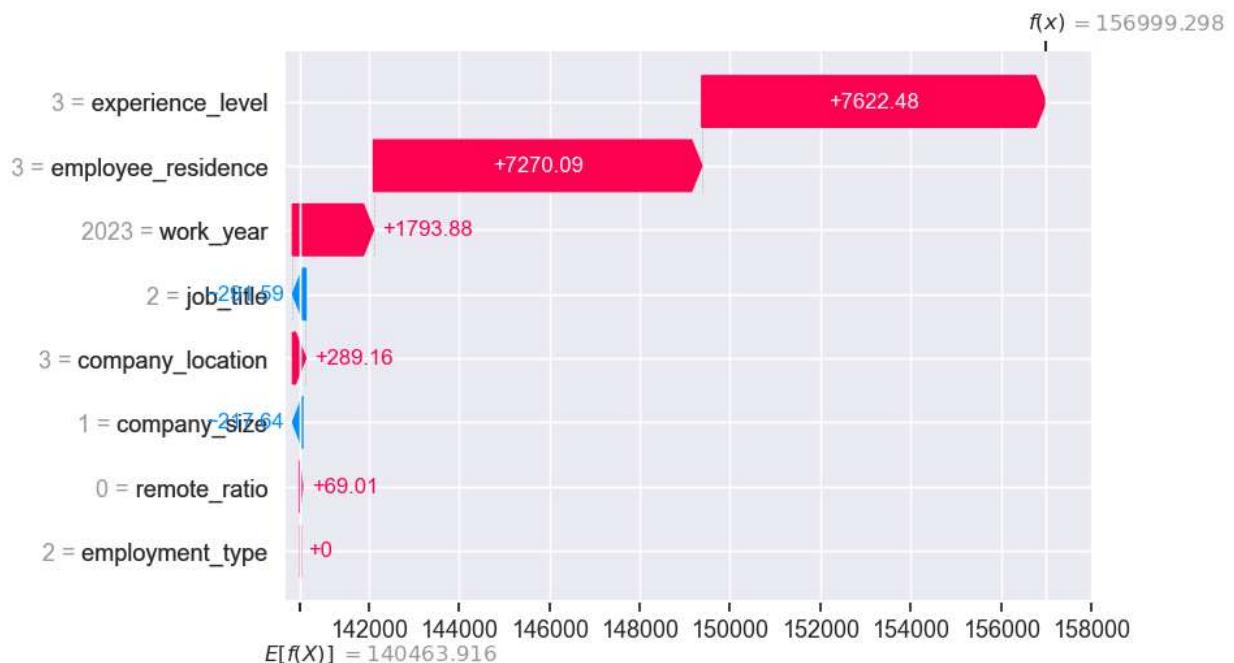
```
  Downloading shap-0.41.0-cp39-cp39-win_amd64.whl (435 kB)
  ----- 435.6/435.6 kB 663.5 kB/s eta 0:00:00
Requirement already satisfied: scikit-learn in c:\users\kiran\anaconda3\lib\site-packages (from shap) (1.2.2)
Requirement already satisfied: tqdm>4.25.0 in c:\users\kiran\anaconda3\lib\site-packages (from shap) (4.64.1)
Collecting slicer==0.0.7
  Downloading slicer-0.0.7-py3-none-any.whl (14 kB)
Requirement already satisfied: scipy in c:\users\kiran\anaconda3\lib\site-packages (from shap) (1.9.1)
Requirement already satisfied: numba in c:\users\kiran\anaconda3\lib\site-packages (from shap) (0.55.1)
Requirement already satisfied: pandas in c:\users\kiran\anaconda3\lib\site-packages (from shap) (1.4.4)
Requirement already satisfied: cloudpickle in c:\users\kiran\anaconda3\lib\site-packages (from shap) (2.0.0)
Requirement already satisfied: numpy in c:\users\kiran\anaconda3\lib\site-packages (from shap) (1.21.5)
Requirement already satisfied: packaging>20.9 in c:\users\kiran\anaconda3\lib\site-packages (from shap) (21.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\kiran\anaconda3\lib\site-packages (from packaging>20.9->shap) (3.0.9)
Requirement already satisfied: colorama in c:\users\kiran\anaconda3\lib\site-packages (from tqdm>4.25.0->shap) (0.4.5)
Requirement already satisfied: llvmlite<0.39,>=0.38.0rc1 in c:\users\kiran\anaconda3\lib\site-packages (from numba->shap) (0.38.0)
Requirement already satisfied: setuptools in c:\users\kiran\anaconda3\lib\site-packages (from numba->shap) (63.4.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\kiran\anaconda3\lib\site-packages (from pandas->shap) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\kiran\anaconda3\lib\site-packages (from pandas->shap) (2022.1)
Requirement already satisfied: joblib>=1.1.1 in c:\users\kiran\anaconda3\lib\site-packages (from scikit-learn->shap) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\kiran\anaconda3\lib\site-packages (from scikit-learn->shap) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\kiran\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas->shap) (1.16.0)
Installing collected packages: slicer, shap
Successfully installed shap-0.41.0 slicer-0.0.7
```

In [152...]

```
import shap
explainer=shap.TreeExplainer(dtree)
shap_values=explainer.shap_values(X_test)
shap.summary_plot(shap_values,X_test)
```



```
In [153]: explainer=shap.Explainer(dtree,X_test)
shap_values=explainer(X_test)
shap.plots.waterfall(shap_values[0])
```



Random Forest Regressor

```
In [159]: from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV

#Create a Random Forest Regressor object

rf=RandomForestRegressor()
```

```
#Define the hyperparameter grid
param_grid={
    'max_depth':[3,5,7,9],
    'min_samples_split':[2,5,10],
    'min_samples_leaf':[1,2,4],
    'max_features':['auto','sqrt']
}

#Create a Grid Search CV Object

grid_search=GridSearchCV(rf,param_grid,cv=5,scoring='r2')

#Fit the GridSearchCV object to the training data

grid_search.fit(X_train,y_train)

#print the best hyperparameters
print('Best Hyperparameters',grid_search.best_params_)
```



















```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestReggressors and ExtraTreesReggressors.
Best Hyperparameters {'max_depth': 7, 'max_features': 'auto', 'min_samples_leaf': 4,
'min_samples_split': 5}
```

In [160...]

```
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor(random_state=0,max_depth=7,min_samples_split=5,min_samples_lea
rf.fit(X_train,y_train)
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestReggressors and ExtraTreesReggressors.
```

Out[160]:

```
▼ RandomForestRegressor
RandomForestRegressor(max_depth=7, max_features='auto', min_samples_leaf=4,
min_samples_split=5, random_state=0)
```

In [161...]

```
from sklearn import metrics
from sklearn.metrics import mean_absolute_percentage_error
import math
y_pred=rf.predict(X_test)
mae=metrics.mean_absolute_error(y_test,y_pred)
mape=mean_absolute_percentage_error(y_test,y_pred)
mse=metrics.mean_squared_error(y_test,y_pred)
r2=metrics.r2_score(y_test,y_pred)
rmse=math.sqrt(mse)

print('MAE is {}'.format(mae))
print('MAPE is {}'.format(mape))
print('MSE is {}'.format(mse))
print('R2 is {}'.format(r2))
print('RMSE is {}'.format(rmse))
```

```
MAE is 38701.948344562785
MAPE is 0.3695190872096532
MSE is 2736934941.641224
R2 is 0.35851318237976904
RMSE is 52315.723655906964
```

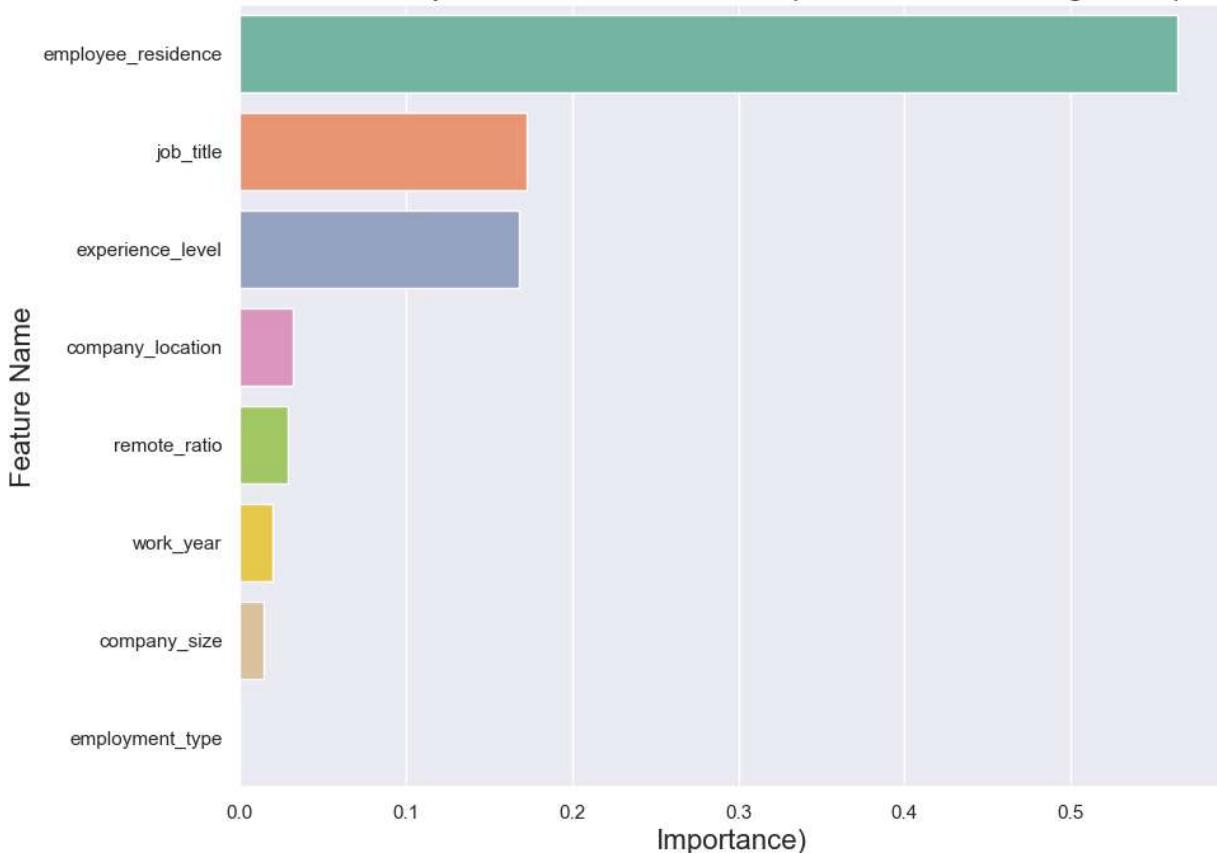
In [162...]

```
imp_df=pd.DataFrame({
    'Feature Name':X_train.columns,
    'Importance':rf.feature_importances_
})

fi=imp_df.sort_values(by='Importance',ascending=False)

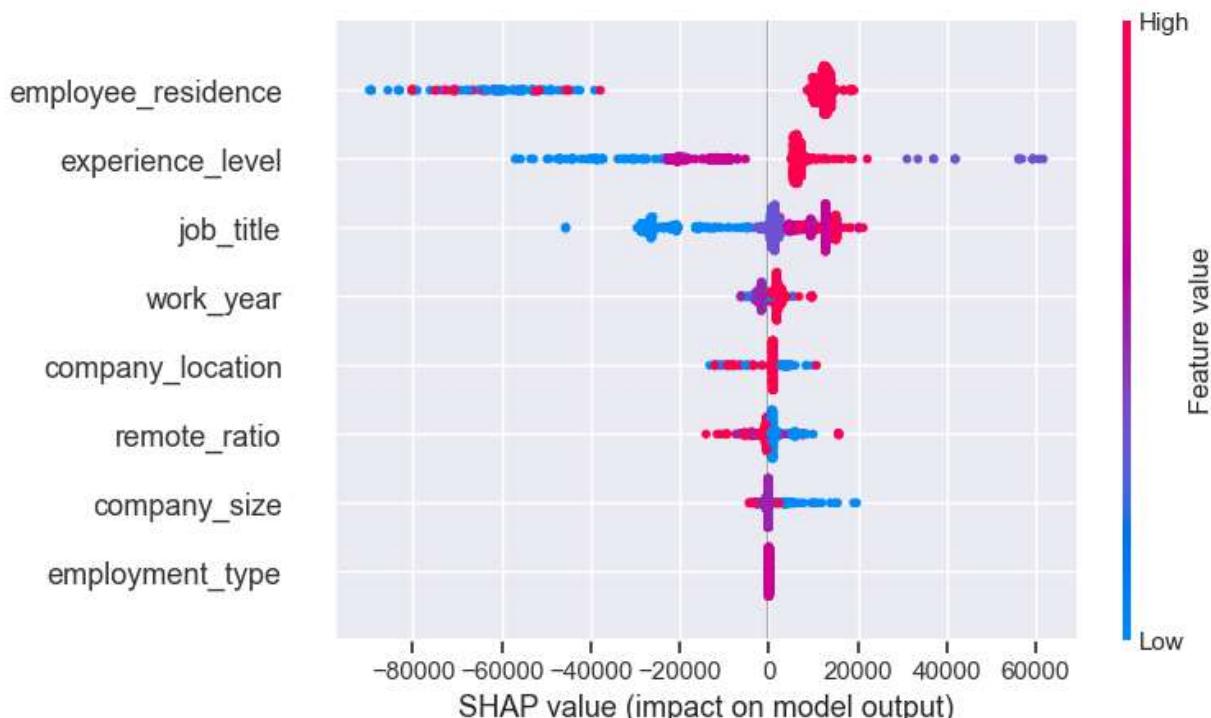
fi2=fi.head(10)
plt.figure(figsize=(10,8))
sns.barplot(data=fi2,x='Importance',y='Feature Name')
plt.title('Feature importance Each Attributes (Random Forest Regressor)',fontsize=18)
plt.xlabel('Importance',fontsize=16)
plt.ylabel('Feature Name',fontsize=16)
plt.show()
```

## Feature importance Each Attributes (Random Forest Regressor)



In [163...]

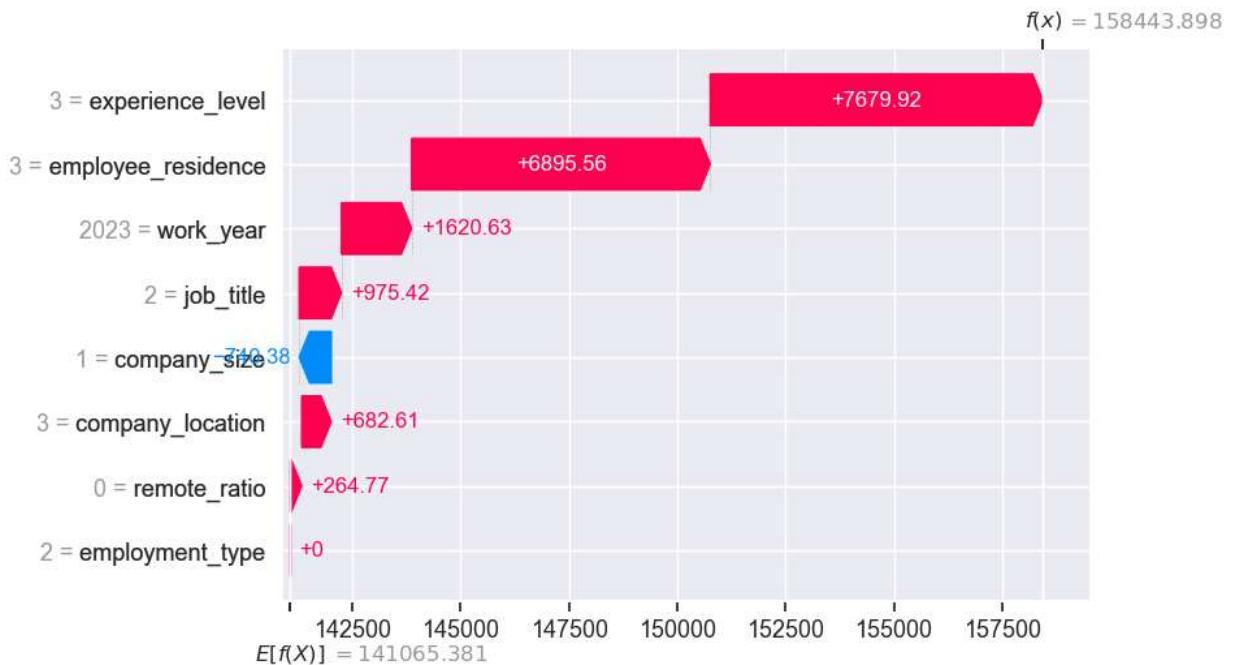
```
import shap
explainer=shap.TreeExplainer(rf)
shap_values=explainer.shap_values(X_test)
shap.summary_plot(shap_values,X_test)
```



In [164...]

```
explainer=shap.Explainer(rf,X_test)
shap_values=explainer(X_test)
```

```
shap.plots.waterfall(shap_values[0])
```



In [ ]: