ASSIGNMENT14/TASK14

SHWETA JHA

Reg.No- GO_STP_12574

**Practice KNN - We have a dataset that contains multiple user's information through the social network who are interested in buying SUV Car or not. **

```
import sklearn
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df=pd.read_csv('/content/User_Data.csv')
df.head()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

```
df.tail()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

```
df.ndim
```

```
2
```

```
df.size
```

```
2000
```

```
df.columns
```

```
Index(['User ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'], dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   User ID          400 non-null    int64
 1   Gender           400 non-null    object
 2   Age              400 non-null    int64
 3   EstimatedSalary  400 non-null    int64
 4   Purchased        400 non-null    int64
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

```
df.describe()
```

	User ID	Age	EstimatedSalary	Purchased
count	4.000000e+02	400.000000	400.000000	400.000000
mean	1.569154e+07	37.655000	69742.500000	0.357500
std	7.165832e+04	10.482877	34096.960282	0.479864
min	1.556669e+07	18.000000	15000.000000	0.000000
25%	1.562676e+07	29.750000	43000.000000	0.000000
50%	1.569434e+07	37.000000	70000.000000	0.000000
75%	1.575036e+07	46.000000	88000.000000	1.000000
max	1.581524e+07	60.000000	150000.000000	1.000000

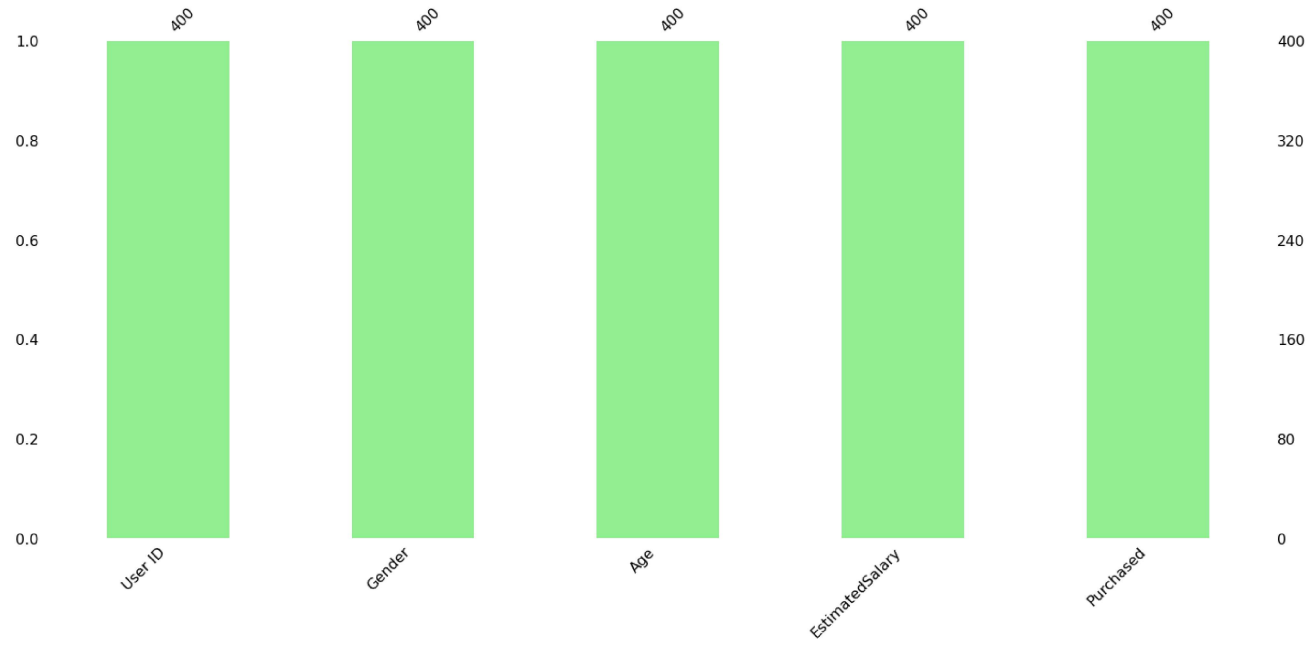
```
df.dtypes
```

```
User ID             int64
Gender             object
Age                 int64
EstimatedSalary     int64
Purchased           int64
dtype: object
```

```
x=df.iloc[:,[2,3]].values
y=df.iloc[:,4].values
```

Missing Data

```
import missingno as msno
msno.bar(df,color="lightgreen")
plt.show()
```



KNN(K-Nearest Neighbour)

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=42)
```

```
from sklearn.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier(n_neighbors=3)
```

```
knn.fit(X_train,y_train)

    KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
```

```
                       metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                       weights='uniform')
```

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

y_pred=knn.predict(X_test)
y_pred
```

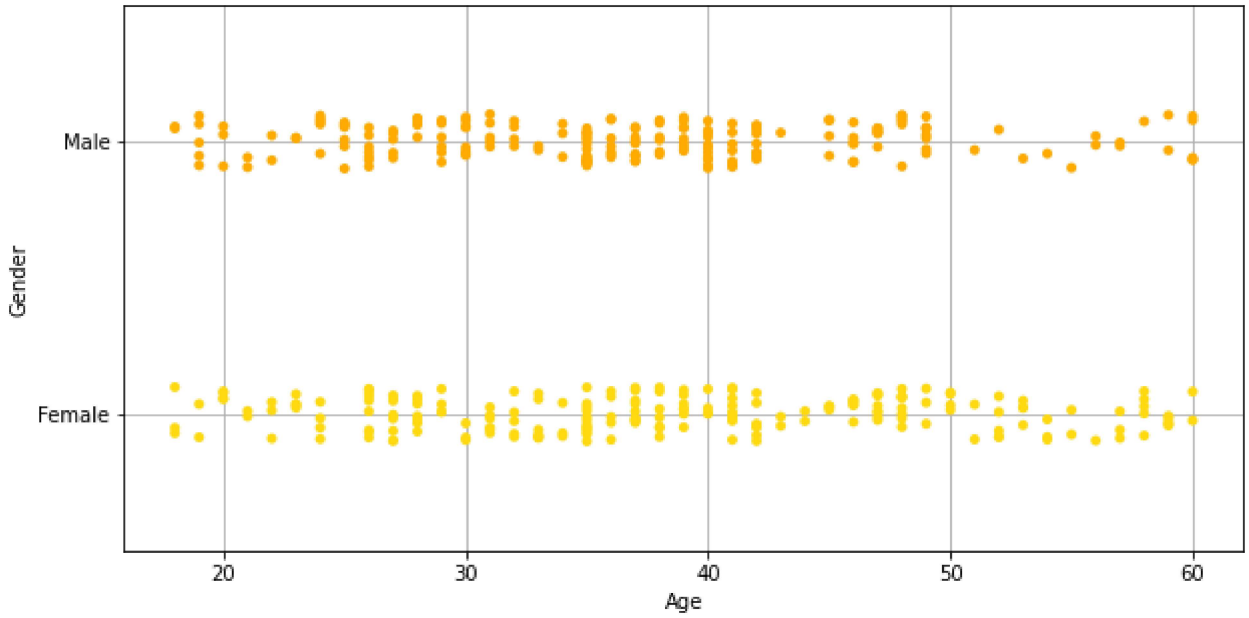
```
array([1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0])
```

```
cm=confusion_matrix(y_test,y_pred)
print(cm)
print(" correct predicition",accuracy_score(y_test,y_pred))
print(" worng predicition",(1-accuracy_score(y_test,y_pred)))
```

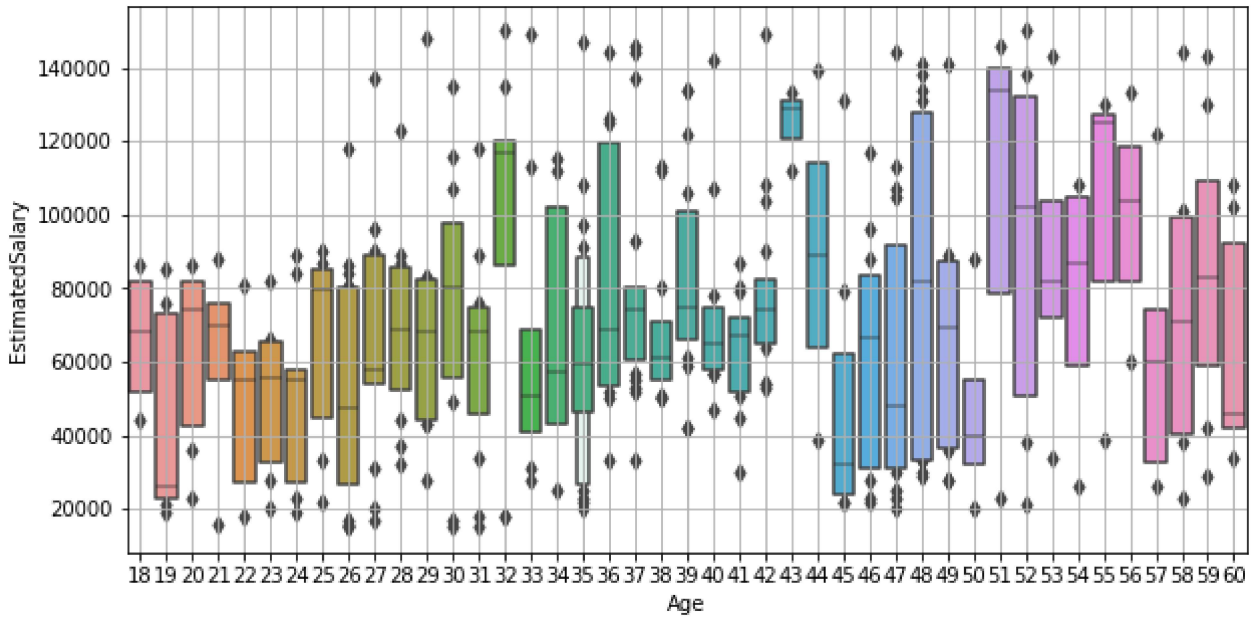
```
[[70 10]
 [21 31]]
 correct predicition 0.7651515151515151
 worng predicition 0.23484848484848486
```

Data Visualizations

```
plt.figure(figsize=(10,5))
sns.stripplot(data=df,y='Gender',x='Age',palette='Wistia_r')
plt.grid()
plt.show()
```

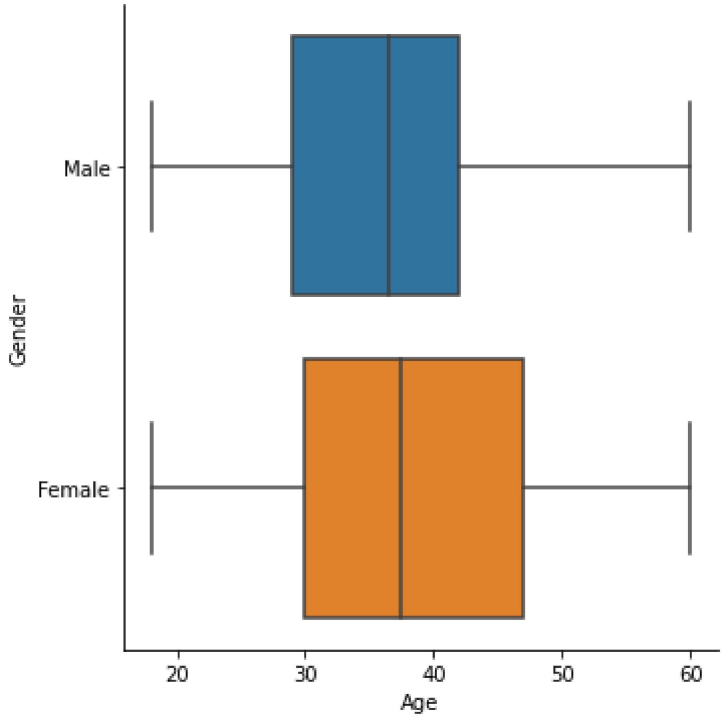


```
plt.figure(figsize=(10,5))
sns.boxenplot(x='Age',y='EstimatedSalary',data=df)
plt.grid()
plt.show()
```

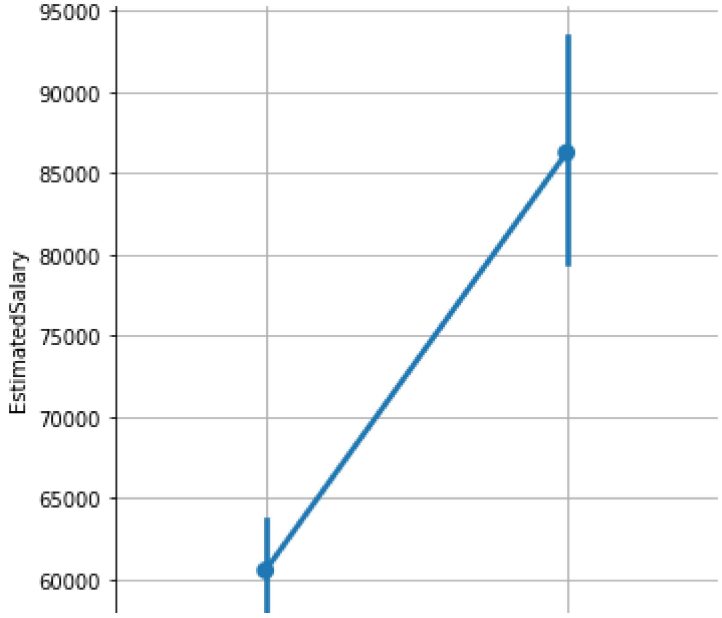



```
sns.catplot(data=df,y='Gender',x='Age', kind="box")
```

```
<seaborn.axisgrid.FacetGrid at 0x7ff53409c2d0>
```

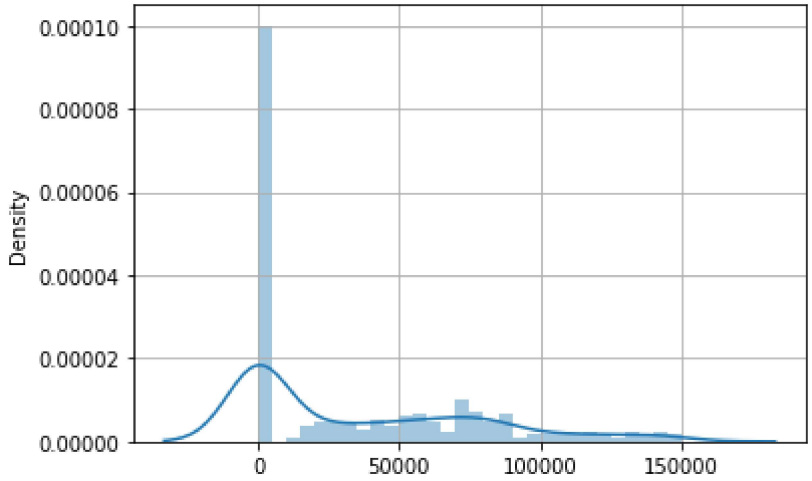



```
sns.catplot(data=df,y='EstimatedSalary',x='Purchased', kind="point")
plt.grid()
plt.show()
```



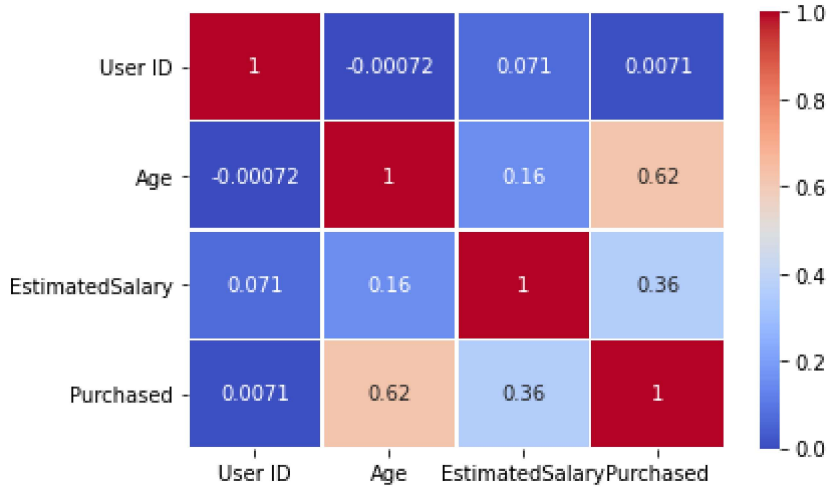

```
sns.distplot(x, kde=True, rug=False, hist=True, bins=30)
plt.grid()
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning:
  warnings.warn(msg, FutureWarning)
```




```
sns.heatmap(df.corr(), cmap='coolwarm', annot=True, linewidths=0.30)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff53188e710>
```



✓ 1s completed at 2:03 AM