KietH_Regression Project

April 26, 2024

You may use this notebook for your project or you may develop your project on your own machine. Either way, be sure to submit all your code to Vocareum via this notebook or upload any code used for your project as a part of the sumbission.

If you intend to use this notebook for your report (pdf) submission; be sure to look into markdown text for any discussion you need: Jupyter Documentation

1 Partially comprehensive guide on your Future aka (Ja Oh Bee)

Dataset used:

1. Adult-UCI Machine Learning Repository
   - **'Age'**- Integer
   - **'workclass'** - Categorical
   - **'education'** - Categorical
   - **'education-num'** - Integer
   - **'marital-status'** - Categorical
   - **'sex'** - Binary
   - **'income'** - Targeting variables - Binary
2. Employee Productivity and Satisfaction HR Data
   - **'Age'** - age of employee
   - **'Gender'** - gender of employee
   - **'Projects Completed'** - no of projects completed out of 25
   - **'Satisfaction Rate'** - rate out of 100
   - **'Feedback Score'** - score out of 5
3. Employee Turnover
   - **'stag'** - Experience
   - **'event'** - Employee Turnover
   - **'profession'** - Employee Profession

2 Importing Libraries

```
In [38]: #!pip install scipy #delete and reinstall scipy
         import numpy as np
```

```
import pandas as pd
import sklearn
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LogisticRegression
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
```

3 This code was ran with local machine.

On UCI repository, the dataset was under Stata format, therefore, I performed this code, and imported the dataset in CSV to Jupyter Notebook.

```
In [ ]: #file_path = ''

         #Define column headers based on the apparent structure of the file
        #column_headers = [
            #'Age', 'Workclass', 'Fnlwgt', 'Education', 'Education-num',
            #'Marital-status', 'Occupation', 'Relationship', 'Race', 'Sex',
            #'Capital-gain', 'Capital-loss', 'Hours-per-week', 'Native-country', 'Income'
        #]

         #Load the data into a dataFrame with the define headers
        #data = pd.read_csv(file_path, header=None, names=column_headers)

         #Define the path to save the new CSV
        #new_csv_path = ''
        #data.to_csv(new_csv_path, index=False)
```

4 Transforming data with a set of dummies variables

For each category in your categorical variable, create a new binary variable. This variable takes the value 1 if the observation falls into that category and 0 otherwise.

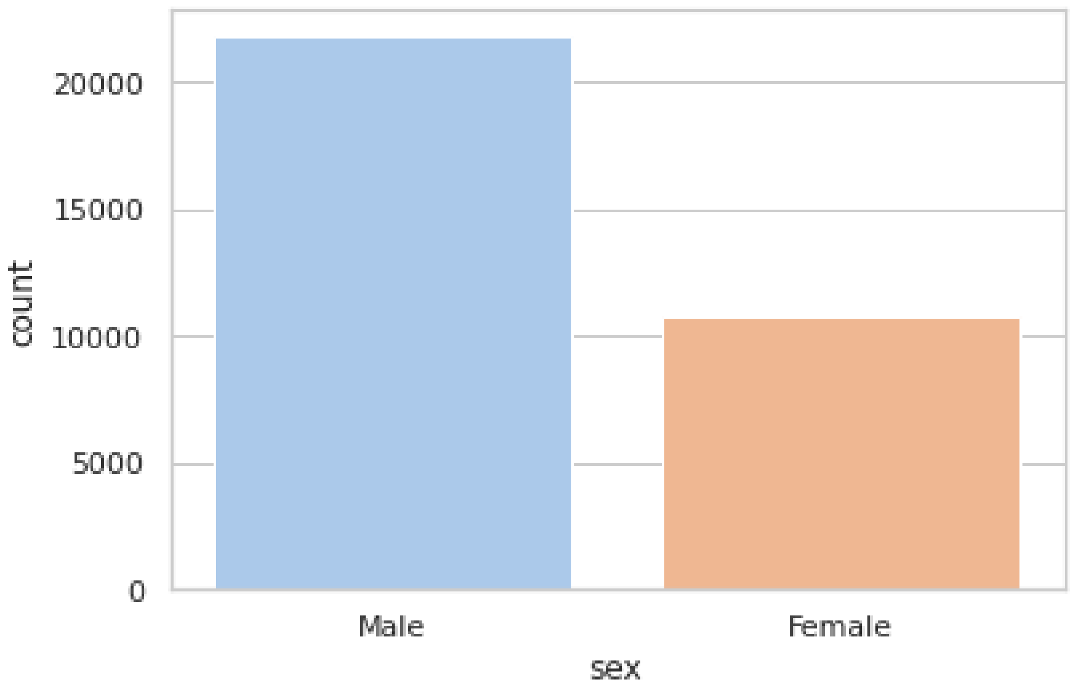
```
In [40]: adult_df = pd.read_csv('adult_dataset (1).csv')
         adult_df.head(10)

         missing_data = adult_df.isnull().sum()
         #Converting categorical variable - Using a fun way like one-hot encoding
         #defining categorical column
         categorical_columns = [col for col in adult_df.columns if adult_df[col].dtype == 'obj
         adult_df['income'].replace({'<=50K':0,'>50K':1},inplace=True)
         adult_df_encoded = pd.get_dummies(adult_df, columns=categorical_columns, drop_first=T
```

```python
#Spliting the data into two set of data, in-sample and out-of-sample data
train_data, test_data = train_test_split(adult_df_encoded, test_size=0.2, random_state
```

5 Adult.csv overview

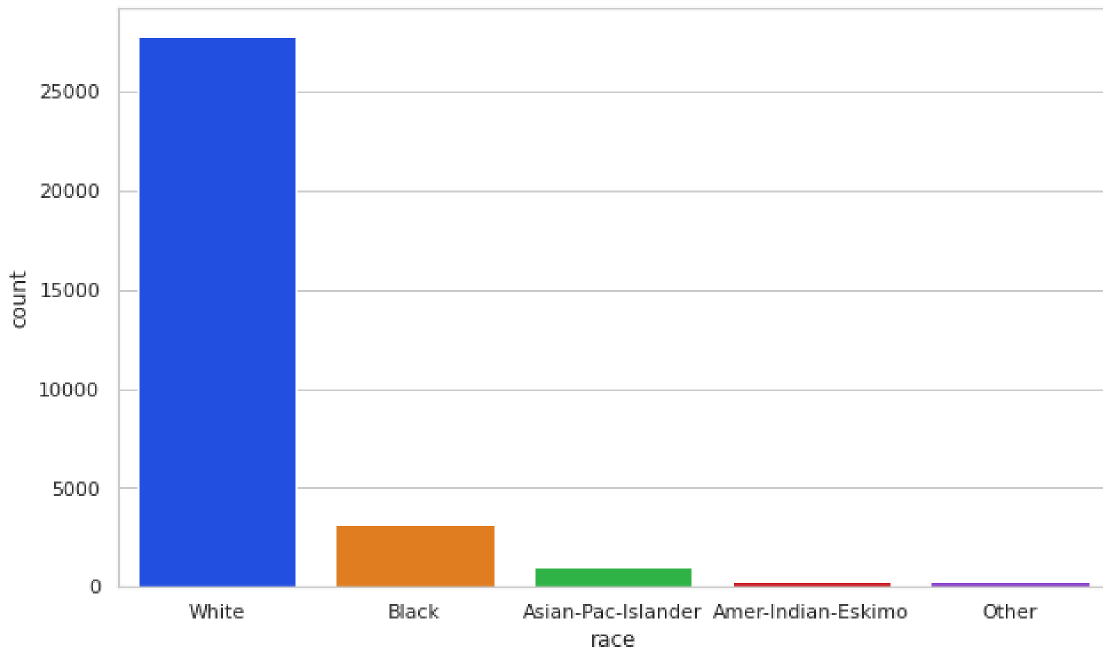
In [41]:
```python
#Distribution of gender
sns.countplot(x = 'sex',data=adult_df,palette='pastel')
```

Out[41]: <AxesSubplot:xlabel='sex', ylabel='count'>



In [42]:
```python
plt.figure(figsize=(10,6),)
sns.countplot(x = 'race',data=adult_df,palette='bright')
plt.xticks(rotation=0)
```

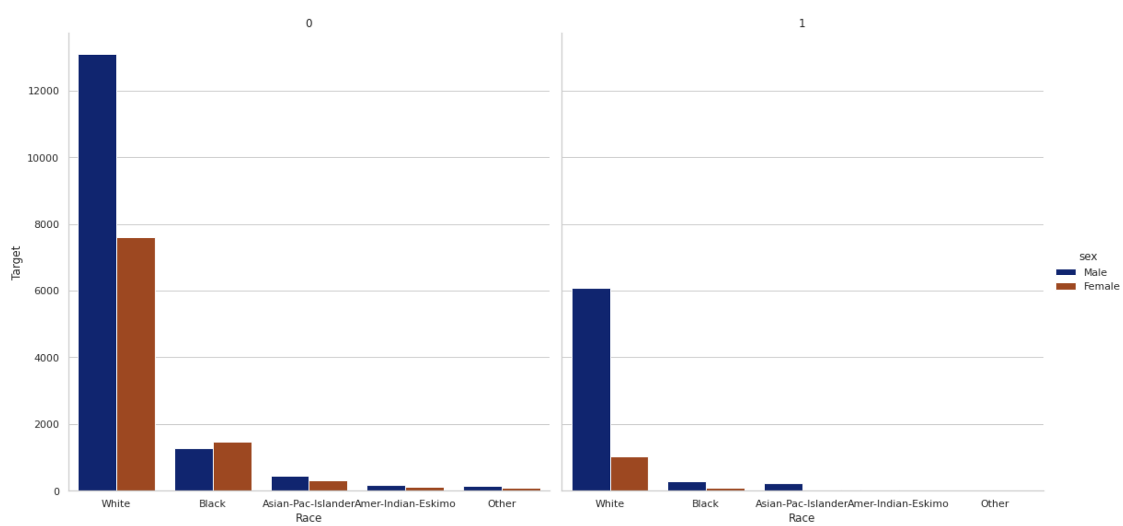
Out[42]:
```
(array([0, 1, 2, 3, 4]),
 [Text(0, 0, 'White'),
  Text(1, 0, 'Black'),
  Text(2, 0, 'Asian-Pac-Islander'),
  Text(3, 0, 'Amer-Indian-Eskimo'),
  Text(4, 0, 'Other')])
```



In [43]:
```
#Comprehensive distribution of individuals on whether their income exceeded 50K
plt.figure(figsize=(20,10))

g = sns.catplot(x='race',hue='sex', col = 'income', data=adult_df,
                kind='count', palette='dark',height=8)
g.set_axis_labels("Race", "Target")
g.set_titles("{col_name}")
```

Out[43]: <seaborn.axisgrid.FacetGrid at 0x7f468a08a7c0>

<Figure size 1440x720 with 0 Axes>



6 Regression Analysis

- Correlation between **explanatory variables** and the **predicted outcome**. This type of correlation helps to understand the strength of the relationship between two variables.

$P(y = 1|x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+...+\beta_n x_n)}}$

```
In [57]: a=['age','capital-loss','capital-gain','hours-per-week','fnlwgt']
         for i in a:
             #print(a)
             print(i,':',stats.pointbiserialr(adult_df['income'],adult_df[i])[0])

age : 0.23403710264885763
capital-loss : 0.15052631177035364
capital-gain : 0.2233288181953827
hours-per-week : 0.22968906567081054
fnlwgt : -0.009462557247529214
```

```
In [45]: X_train = train_data.drop('income_1', axis=1)
         y_train = train_data['income_1']
         X_test = test_data.drop('income_1', axis=1)
         y_test = test_data['income_1']

         logistic_regression = LogisticRegression(max_iter=1000, solver='liblinear')
         logistic_regression.fit(X_train, y_train)

Out[45]: LogisticRegression(max_iter=1000, solver='liblinear')
```

```
In [58]: coefficients = logistic_regression.coef_[0]
         feature_names = X_train.columns
         intercept = logistic_regression.intercept_[0]

         print(coefficients)
         print(intercept)

[-3.84988044e-03 -3.83762908e-06 -2.45718583e-03  3.31314855e-04
  7.58921383e-04 -1.00398951e-02  1.75349546e-04  7.94574875e-05
 -3.50256879e-06 -1.95734624e-03  3.96381828e-04 -3.47179793e-05
 -1.80866042e-05 -5.26831697e-06 -4.16304389e-04 -1.22787640e-04
 -5.14408103e-05 -9.86973917e-05 -1.91650533e-04 -1.47174809e-04
 -3.65131187e-05 -2.07381827e-05  1.02923485e-03  2.71715637e-04
 -1.78675585e-03  6.41643729e-04 -2.09194858e-05  3.01364257e-04
 -9.39156591e-04  4.06992844e-06  3.92506680e-03 -1.16942818e-04
 -3.90989803e-03 -3.15802676e-04 -2.92609120e-04 -8.09180852e-04
```

```
-2.94578654e-06 -2.35121199e-04  1.19668950e-03 -2.73640267e-04
-4.15742007e-04 -4.27043026e-04 -1.17188953e-03 -6.04241589e-05
 8.95118017e-04  8.17551353e-05 -5.18918190e-05  6.67404799e-05
-1.56654769e-04 -2.26947130e-03 -3.52531043e-04 -2.13300102e-03
-1.09214541e-03  4.46506029e-04 -7.52991026e-05 -5.96629509e-04
-7.46010376e-05 -1.02872660e-03  1.02385345e-03  2.00130947e-06
 5.48291471e-06 -6.36659546e-06 -2.05890452e-05  9.43566986e-06
-2.68826278e-05 -8.88853879e-06 -2.48787147e-05  8.97441079e-06
 1.29730346e-05  7.02616793e-06 -4.75814928e-06 -1.67130355e-05
-9.60501441e-06 -1.59472902e-06 -3.42656214e-06  5.05156732e-06
-3.59447518e-06  1.11970321e-05  3.58799827e-06 -1.74539155e-06
 1.43969633e-05 -1.25136236e-05  8.43928865e-06 -4.53219615e-06
-1.45555116e-04 -7.21119907e-06 -6.24429488e-06 -9.32857994e-06
 6.25961893e-07 -5.72292159e-06 -8.65869290e-06 -3.40087310e-05
 1.04150465e-06 -1.45191548e-05  9.12515986e-06  4.53267536e-07
-4.62637567e-06 -1.54841401e-03 -2.88857706e-05  6.17555544e-06]
-0.001876633302712089
```

In [47]:
```
y_pred = logistic_regression.predict(X_test)

accuracy = logistic_regression.score(X_test, y_test)
mse = mean_squared_error(y_test, y_pred)

accuracy, mse
```

Out[47]:
```
(0.7994779671426377, 0.2005220328573622)
```

7 Key Takeaways

- *Accuracy*: 79.95%
- *Mean Squared Error (MSE)*: 0.2005
  - For this dataset, Xgboost, Random Forest Classification will yield higher accuracy than Logistic Regression. However, these methods require OOP which is out of my capability.

8 Turnover.csv dataset

In [48]:
```
turnover_df = pd.read_csv('turnover.csv',encoding='latin1')
turnover_df.head(5)
#turnover_df['profession'].unique()

turnover_df.groupby('industry')
turnover_df
```

Out[48]:
```
               stag  event gender   age         industry  profession  \
0          7.030801      1      m  35.0            Banks          HR
```

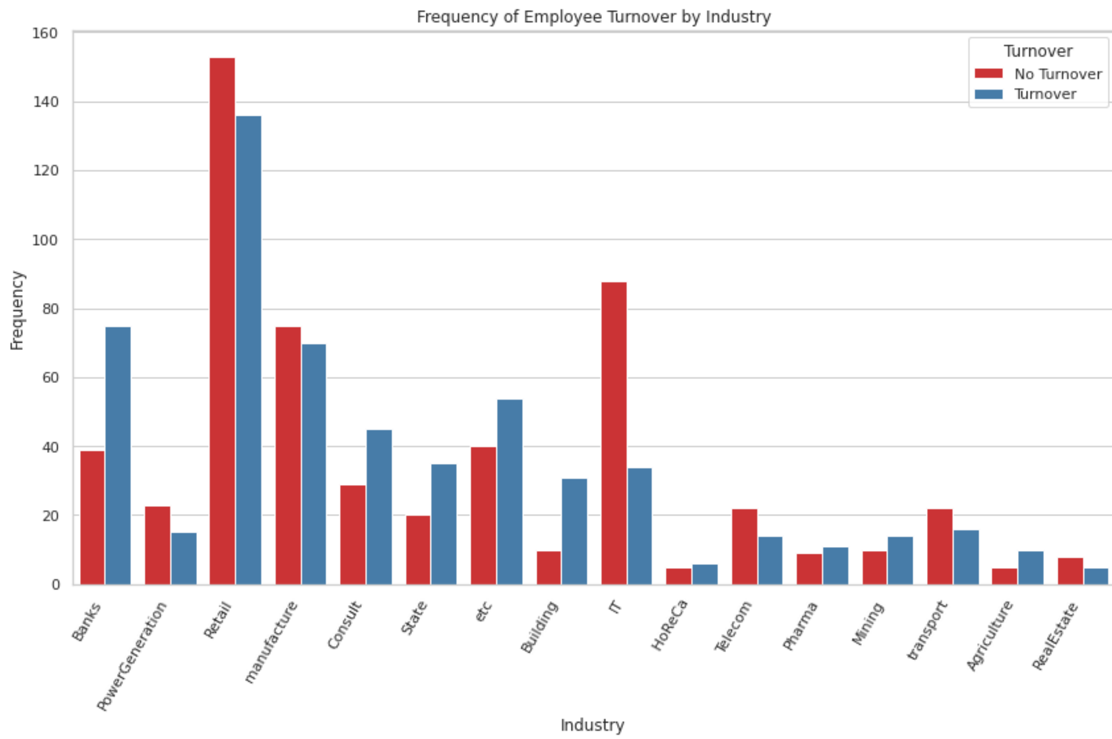
```
1      22.965092      1      m  33.0             Banks          HR
2      15.934292      1      f  35.0  PowerGeneration          HR
3      15.934292      1      f  35.0  PowerGeneration          HR
4       8.410678      1      m  32.0            Retail  Commercial
...          ...    ...    ...   ...               ...         ...
1124   10.611910      0      f  41.0             Banks          HR
1125   10.611910      0      f  41.0             Banks          HR
1126  118.800821      0      f  34.0           Telecom  Accounting
1127   49.412731      0      f  51.0           Consult          HR
1128   24.837782      0      f  29.0            Retail          HR

          traffic    coach head_gender greywage  way  extraversion  independ  \
0     rabrecNErab       no           f    white  bus           6.2       4.1
1           empjs       no           m    white  bus           6.2       4.1
2     rabrecNErab       no           m    white  bus           6.2       6.2
3     rabrecNErab       no           m    white  bus           5.4       7.6
4           youjs      yes           f    white  bus           3.0       4.1
...           ...      ...         ...      ...  ...           ...       ...
1124  rabrecNErab  my head           m    white  bus           8.6       3.4
1125  rabrecNErab  my head           m    white  bus           8.6       3.4
1126           KA       no           f    white  bus           4.6       5.5
1127        empjs       no           m     grey  bus           3.8       7.6
1128        youjs       no           f    white  car           9.4       1.2

      selfcontrol  anxiety  novator
0             5.7      7.1      8.3
1             5.7      7.1      8.3
2             2.6      4.8      8.3
3             4.9      2.5      6.7
4             8.0      7.1      3.7
...           ...      ...      ...
1124          2.6      4.8      8.3
1125          2.6      4.8      8.3
1126          7.2      6.3      3.7
1127          5.7      6.3      5.2
1128          4.1      5.6      6.7

[1129 rows x 16 columns]
```

9 Turnover.csv overview

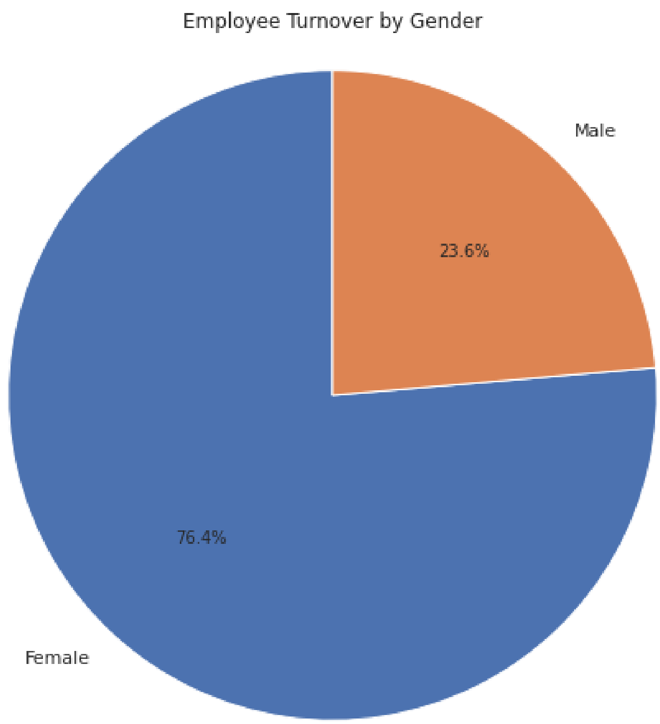
In [49]:

```python
# Create a countplot to show turnover frequency by industry
plt.figure(figsize=(12, 8))
sns.countplot(data= turnover_df, x='industry', hue='event', palette='Set1')
plt.title('Frequency of Employee Turnover by Industry')
plt.xlabel('Industry')
plt.ylabel('Frequency')
```

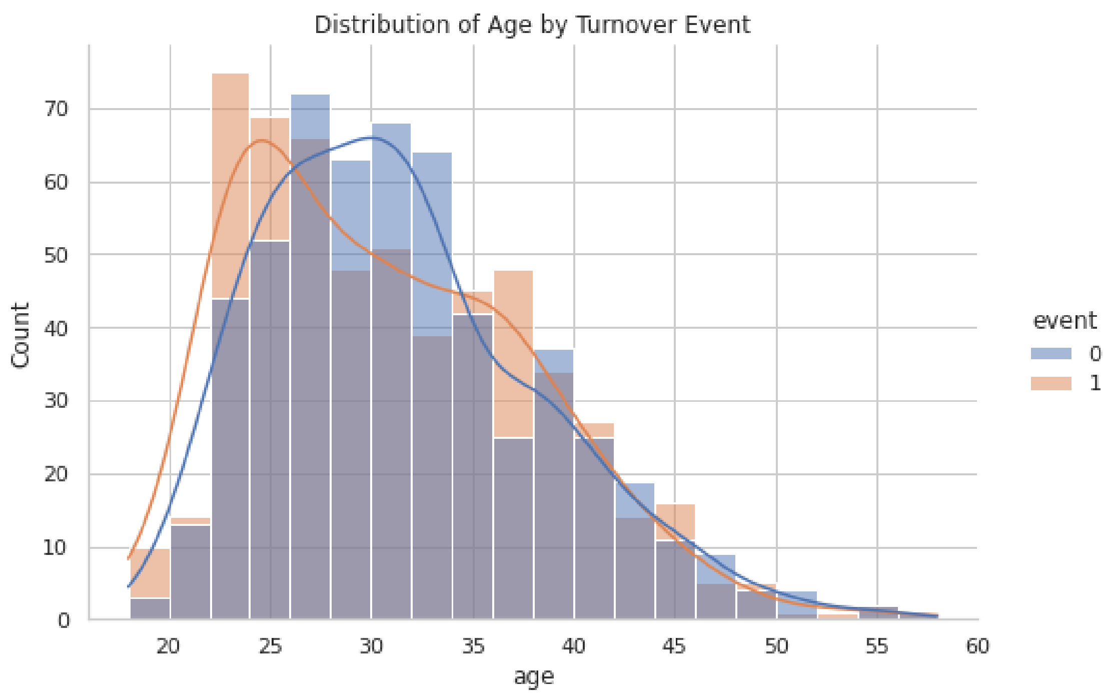
```
plt.xticks(rotation=60, ha='right')
plt.legend(title='Turnover', labels=['No Turnover', 'Turnover'])
plt.tight_layout()

# Show the plot
plt.show()
```



In [50]:
```
#Data Visualization
%matplotlib inline
import matplotlib.pyplot as plt

turnover_events = turnover_df[turnover_df['event'] == 1]
counts = turnover_events['gender'].value_counts()

plt.figure(figsize=(12, 8))
plt.pie(counts, labels=['Female','Male'], autopct='%1.1f%%', startangle=90)
plt.title('Employee Turnover by Gender')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



In [51]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Set the aesthetic style of the plots
sns.set(style="whitegrid")

# Create a distribution plot of 'age' with a hue based on 'event'
plot = sns.displot(data=turnover_df, x='age', hue='event', kind='hist', bins=20, kde=
plot.set(title='Distribution of Age by Turnover Event')
plt.show()
```



10 Key Takeaways

- People have the tendency to job hopping during their mid 20s until early 30s. Genders are also significant, with over 3/4 of Female ever left their jobs.
- One of a very interesting Behavioral Economics Experiment was Niederle & Vesterlund (2007). The study examine that " Do women shy away from competition? Do men compete too much?
  - Ability Difference? not really. Its actually because significant gender gap in decision to take risk
    - 35% of women vs 73% of men choose to take risk.

In [52]:
```
hr_df = pd.read_csv('hr_dashboard_data.csv')
hr_df.head(5)
```

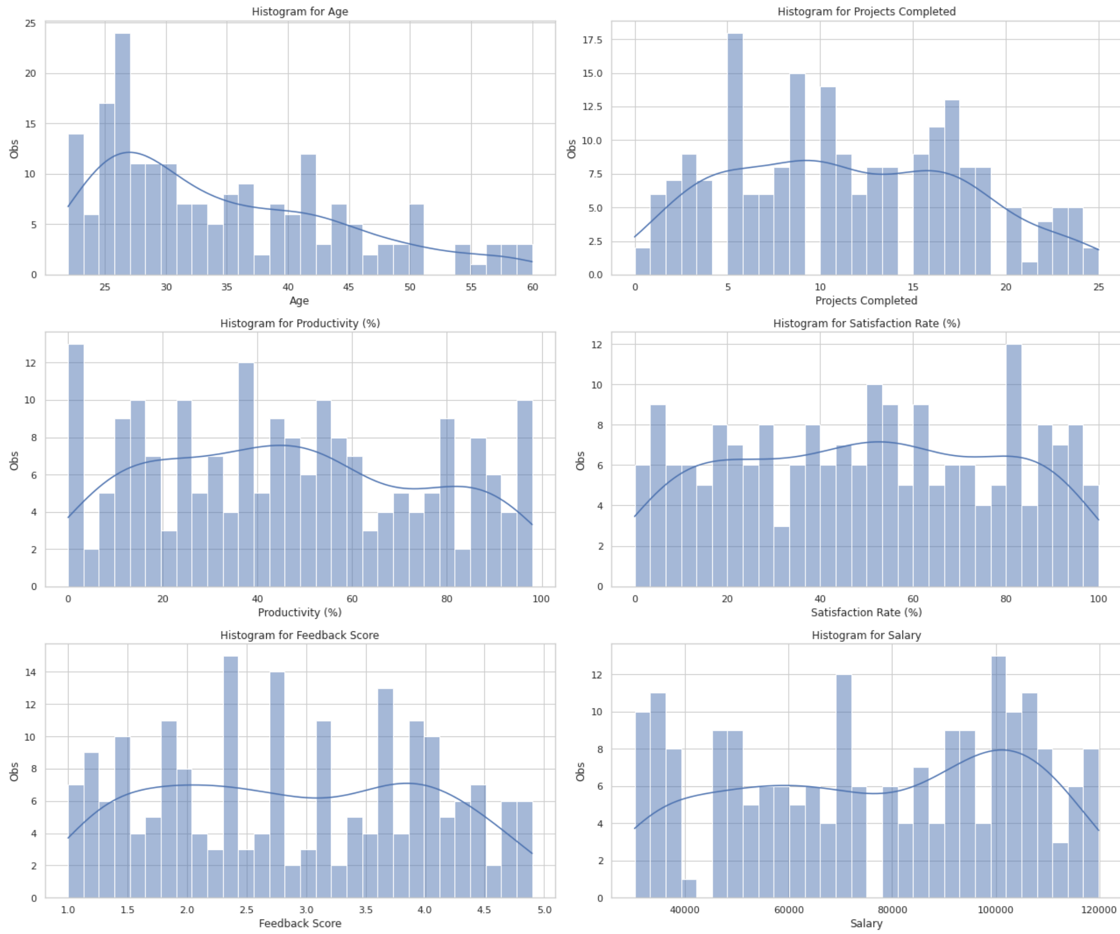
Out[52]:
```
                Name  Age  Gender  Projects Completed  Productivity (%)  \
0    Douglas Lindsey   25    Male                  11                57
1  Anthony Roberson   59  Female                  19                55
2      Thomas Miller   30    Male                   8                87
3       Joshua Lewis   26  Female                   1                53
4  Stephanie Bailey   43    Male                  14                 3

   Satisfaction Rate (%)  Feedback Score Department  Position Joining Date  \
0                     25             4.7  Marketing   Analyst       Jan-20
```

```
1                      76            2.8          IT    Manager        Jan-99
2                      10            2.4          IT    Analyst        Jan-17
3                       4            1.4   Marketing     Intern        Jan-22
4                       9            4.5          IT  Team Lead        Jan-05

   Salary
0   63596
1  112540
2   66292
3   38303
4  101133
```

11 Hr_dashboard_data.csv overview

```
In [53]: num_column =hr_df.select_dtypes(include = ['int','float'])
```

```
In [54]: num_bins = 30

         # Create subplots for each numerical column
         fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(18, 15))

         # Flatten the axes array for easy indexing
         axes = axes.flatten()

         # Iterate through numerical columns
         for i, col in enumerate(num_column.columns):
             # Create histogram with specified bins
             sns.histplot(data=hr_df, x=col, bins=num_bins, palette='dark', kde=True, ax=axes[i
             axes[i].set_title(f'Histogram for {col}')
             axes[i].set_xlabel(col)
             axes[i].set_ylabel('Obs')


         plt.tight_layout()
         plt.show()
```



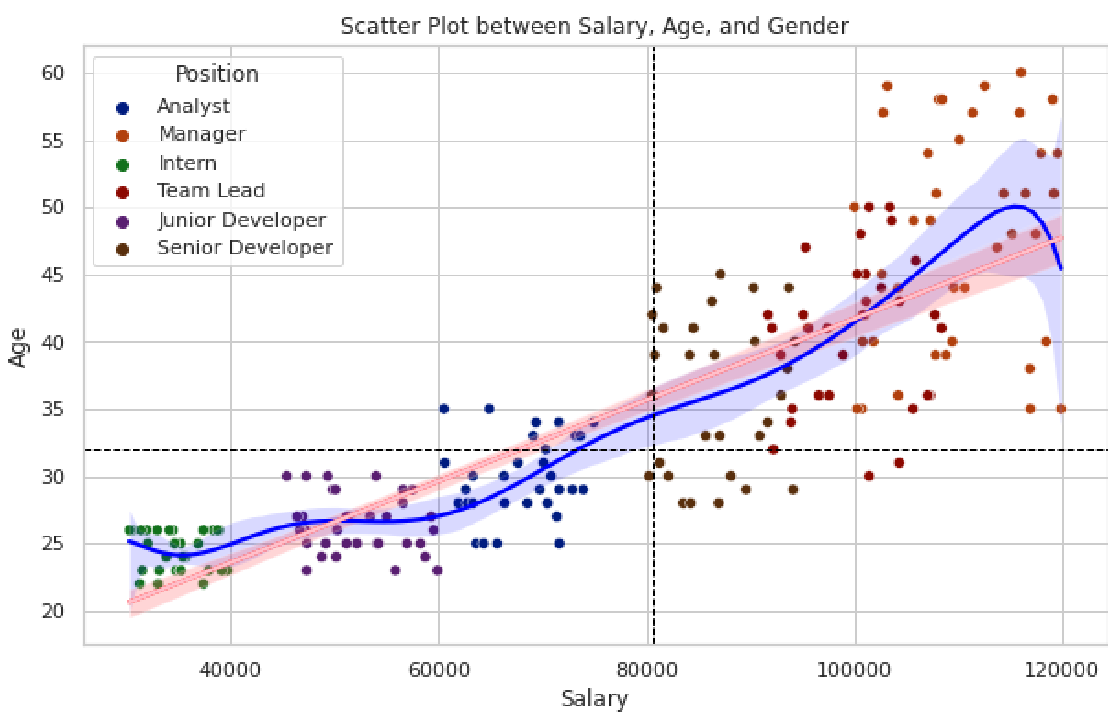
11.1 Histogram Analysis

This set of histograms illustrates various employee-related metrics:

- **Age**: The distribution of ages reveals that most employees are between 25 and 35, with a gradual decline in the number of employees in older age groups.
- **Projects Completed**: The majority of employees complete between 5 and 15 projects, with a few outliers completing 20-25 projects.
- **Productivity**: Productivity shows a relatively uniform distribution, with a slight peak around 50%.
- **Feedback Score**: Feedback scores are evenly distributed across a range from 1.0 to 5.0, with slight peaks at 2.5 and 3.0.
- **Satisfaction Rate**: This histogram displays a fairly flat distribution, with satisfaction rates spread across the entire range.
- **Salary**: Salaries tend to cluster around two main ranges: 40,000-60,000 and 80,000-100,000, with fewer employees in the upper salary brackets.

These histograms provide insights into various aspects of employee demographics and performance indicators within the workplace.

```
In [60]: plt.figure(figsize=(10, 6))
         sns.scatterplot(data=hr_df, x='Salary', y='Age', hue='Position', palette='dark')
         #Fitting the plot with higher-order polynomial regression to capture non-linear,
         sns.regplot(data=hr_df, x='Salary', y='Age', scatter=False, color='blue',order=10)
         #Fitting the plot with log-linear regression
         sns.regplot(data=hr_df, x='Salary', y='Age', scatter=False, color='red', logx=False)
         sns.regplot(data=hr_df, x='Salary', y='Age', scatter=False, color='pink',order=1)
         # Adding lines to divide the plot into four quadrants
         plt.axhline(y=hr_df['Age'].median(), color='black', linestyle='--', linewidth=1)
         plt.axvline(x=hr_df['Salary'].median(), color='black', linestyle='--', linewidth=1)


         plt.title('Scatter Plot between Salary, Age, and Gender')
         plt.xlabel('Salary')
         plt.ylabel('Age')
         plt.legend(title='Position')
         plt.show()
```



12 Key Takeaways

- The histogram above was comprehensive, but it hardly indicates any trends and high variance across observation.

- In term of Position.
  - The graph indicated that **Intern** has among the lowest salary while **Senior Developer**, managing level position will still yield more salary.
  - There is a strong positive correlation between *Age* and *Salary*, indicating that as age increase, salary tends to increase.
  - The job position categories indicate variations in salary growth with age for different genders.