Goals and a business objective

To develop an application to predict the income of an individual, based on the factors used in developing marketing profiles on people.Data provided by the United States Census Bureau was used. The focus was kept on $50,000 as the key number for salary.

Assumptions

Assumptions were made about conversion of original data, final weights, similar demographic characteristics should have similar weights. It was assumed that the salary was binary and categorical. Salary greater than $50,000 was replaced with '1' and salary less than$**50,000 was replaced with '0' for analysis.**

Import libraries, Load data and clean data

In [140…

```python
# Import dependencies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn import metrics
import os
import sys
%matplotlib inline
```

In [2]:

```python
# Load data
adult_df=pd.read_csv('adult.data',header=None)
adult_df.columns=['age','workclass','fnlwgt','education','education-num','marital-statu
adult_df.head()
```

Out[2]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	c
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	

In [3]:

```python
# Clean the dataset by dropping rows with ?
```

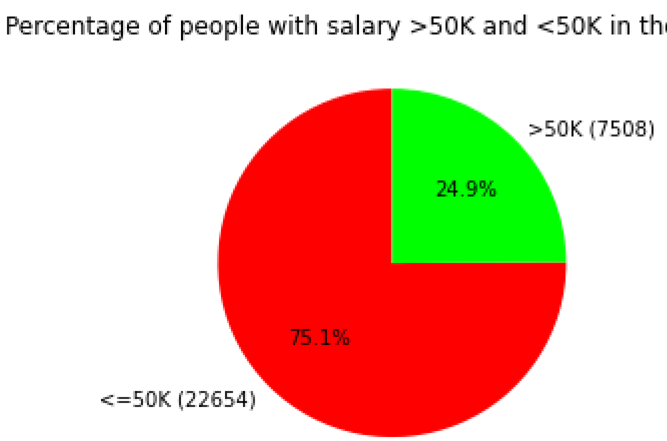
```
clean_df = adult_df.replace(' ?', np.NaN).dropna()
clean_df.shape
```

Out[3]:
```
(30162, 15)
```

In [4]:
```
# Find the number of people with salary >50K and <50K in the dataset
salary_count = clean_df.groupby('salary').size()
salary_count
```

Out[4]:
```
salary
 <=50K    22654
 >50K      7508
dtype: int64
```

In [5]:
```
# Pie chart of the number of people with salary >50K and <50K
plt.pie(salary_count, labels=['<=50K (22654)', '>50K (7508)'], autopct='%1.1f%%', start
plt.title('Percentage of people with salary >50K and <50K in the dataset')
plt.show()
```



Analyze the data to see which attributes or factors contribute to a higher individual salary, and help create various marketing profiles using this analysis.

In [6]:
```
# Assume that the salary is binary and categorical variable and convert it to numeric b
log_df = clean_df.copy()
log_df.loc[clean_df.salary.str.strip() == "<=50K", 'salary'] = 0
log_df.loc[clean_df.salary.str.strip() == ">50K", 'salary'] = 1
log_df.tail()
```

Out[6]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	s
32556	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Fema
32557	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Ma

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	s
32558	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Fema
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Ma
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Fema

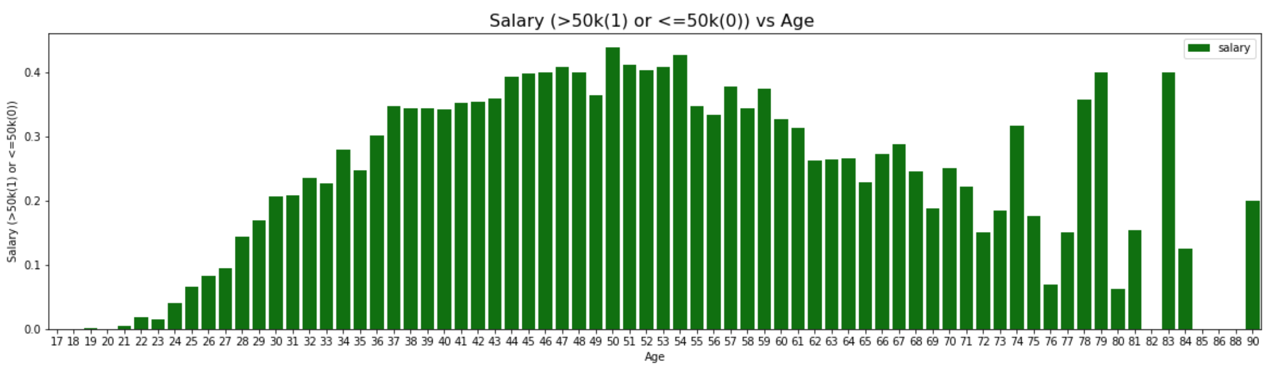
Salary vs Age analysis

In [109…

```python
# Plot showing relation between salary and age
fig = plt.figure(figsize=(20,5))
ax_age = sns.barplot(x='age', y='salary', data=log_df, color='green', label='salary', c
ax_age.set_title("Salary (>50k(1) or <=50k(0)) vs Age ", loc='center', fontsize=16)
ax_age.set_xlabel("Age")
ax_age.set_ylabel("Salary (>50k(1) or <=50k(0))")
ax_age.legend(loc="upper right")
```

Out[109…

```
<matplotlib.legend.Legend at 0x2925c0926a0>
```



Conclusion: There is a significant correlation or influence of age on salary earned.

Salary vs fnlwgt

In [112…

```python
# Relation between salary and fnlwgt below and above 50K
below50_df = log_df[log_df.salary == 0]
above50_df = log_df[log_df.salary == 1]
sumweightbelow = below50_df['fnlwgt'].sum()
sumweightabove = above50_df['fnlwgt'].sum()
print("Percentage weight below 50k: ", (sumweightbelow/(sumweightbelow + sumweightabove
print("Percentage weight above 50k: ", (sumweightabove/(sumweightbelow + sumweightabove
```

```
Percentage weight below 50k:  75.32335207448135
Percentage weight above 50k:  24.676647925518644
```

Conclusion: As it is showing the distribution of the dataset, we can ignore this as an influencing factor.

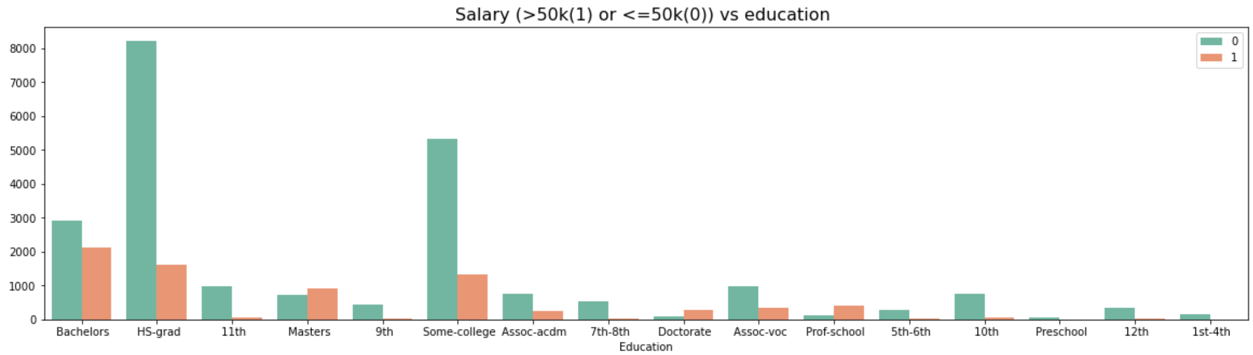
Salary vs Education

In [9]:

```python
# Plot showing relation between salary and education
```

```
fig = plt.figure(figsize=(20,5))
ax_edu = sns.countplot(data=log_df, x='education', hue='salary', palette='Set2')
ax_edu.set_title("Salary (>50k(1) or <=50k(0)) vs education ", loc='center', fontsize=1
ax_edu.set_xlabel("Education")
ax_edu.set_ylabel(" ")
ax_edu.legend(loc="upper right")
```

Out[9]: `<matplotlib.legend.Legend at 0x29242a6f100>`



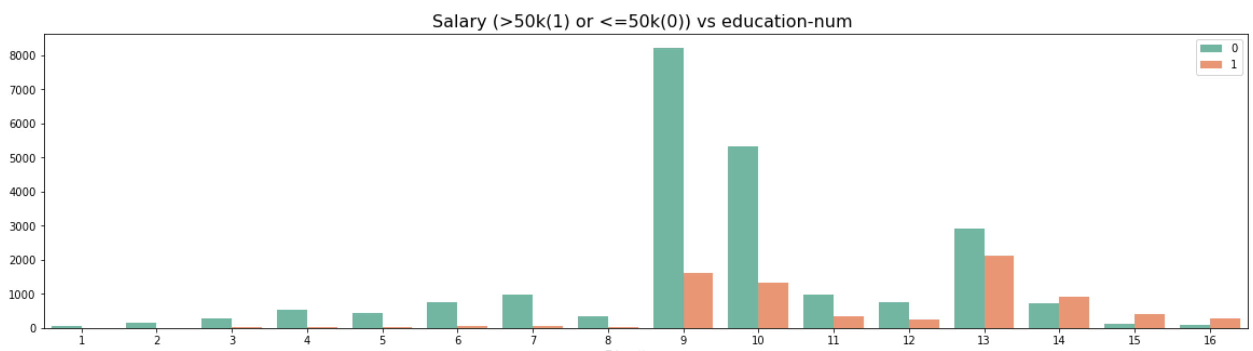
Conclusion: We see that education does influence the earning power and hence can be considered a factor.

Salary vs Education-num

In [10]:

```
# Plot showing correlation between salary and education-num
fig = plt.figure(figsize=(20,5))
ax_edu_num = sns.countplot(data=log_df, x='education-num', hue='salary', palette='Set2'
ax_edu_num.set_title("Salary (>50k(1) or <=50k(0)) vs education-num ", loc='center', fo
ax_edu_num.set_xlabel("Education-num")
ax_edu_num.set_ylabel(" ")
ax_edu_num.legend(loc="upper right")
```

Out[10]: `<matplotlib.legend.Legend at 0x29243c2b700>`



Conclusion: This is same as education and can be considered for the model.

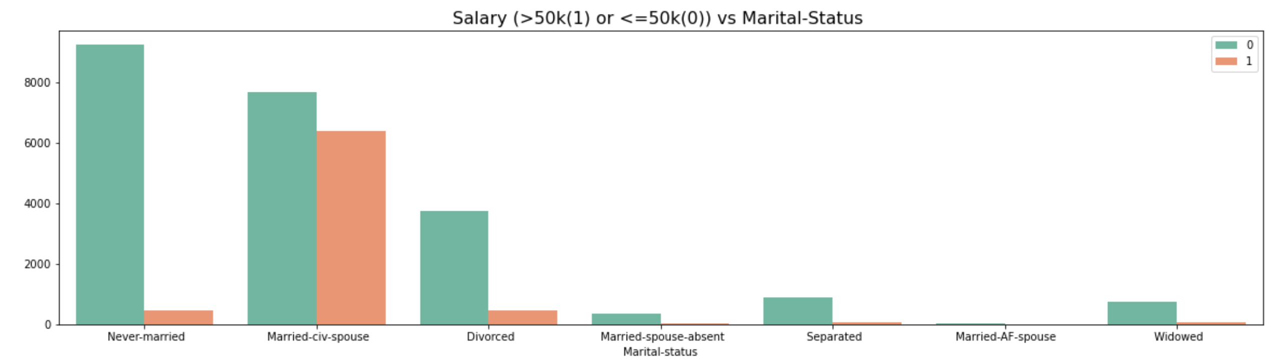
Salary vs Marital status

In [11]:

```
# Plot showing relation between salary and matrial-status matplotlib
fig = plt.figure(figsize=(20,5))
ax_mat = sns.countplot(data=log_df, x='marital-status', hue='salary', palette='Set2')
ax_mat.set_title("Salary (>50k(1) or <=50k(0)) vs Marital-Status ", loc='center', fonts
ax_mat.set_xlabel("Marital-status")
ax_mat.set_ylabel(" ")
ax_mat.legend(loc="upper right")
```

`<matplotlib.legend.Legend at 0x29242a6f160>`

Out[11]:



Conclusion: Again we see a strong influence on salary.

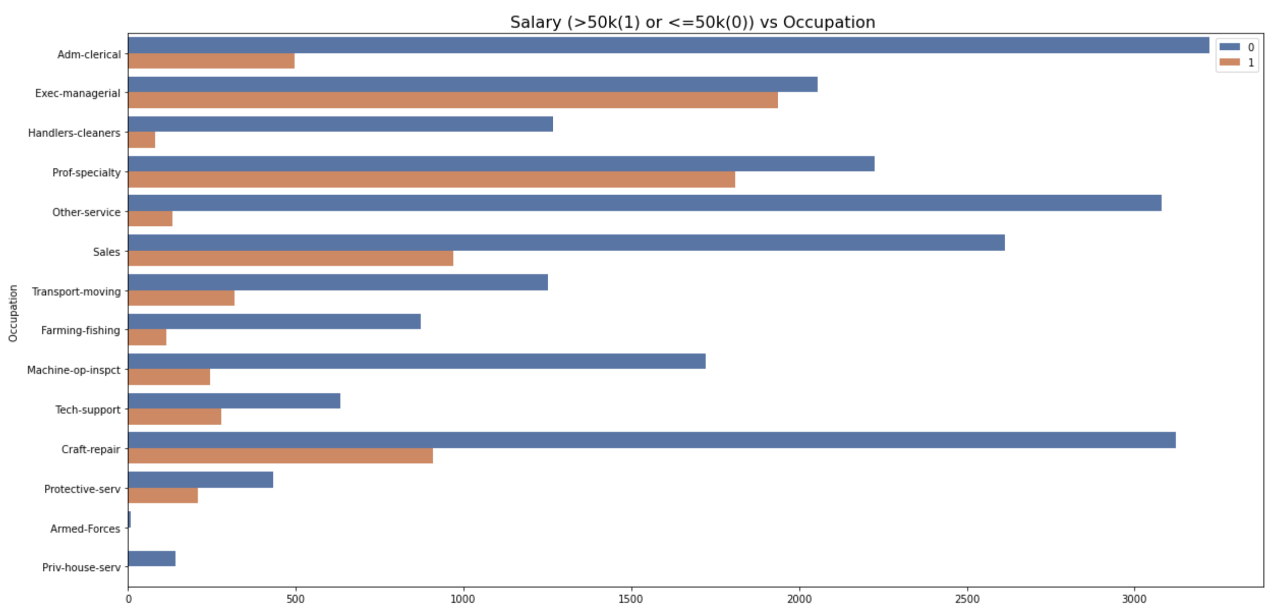
Salary vs Occupation

In [12]:

```
# Plot showing relation between salary and occupation
fig = plt.figure(figsize=(20,10))
ax_occ = sns.countplot(data=log_df, y='occupation', hue='salary', palette='deep',)
ax_occ.set_title("Salary (>50k(1) or <=50k(0)) vs Occupation ", loc='center', fontsize=
ax_occ.set_xlabel("")
ax_occ.set_ylabel("Occupation ")
ax_occ.legend(loc="upper right")
```

Out[12]:

```
<matplotlib.legend.Legend at 0x2924417b1c0>
```



Conclusion: We can conclude that occupations strongly influence salary earned.

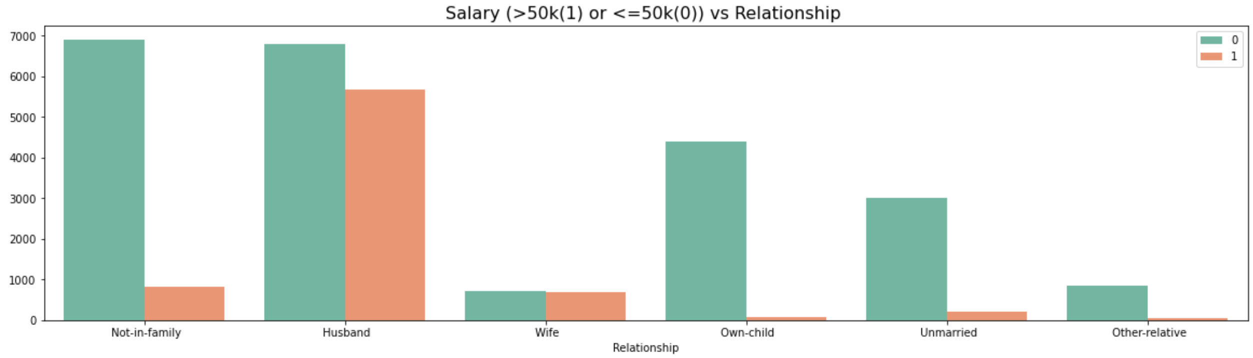
Salary vs Relationship

In [13]:

```
# Plot showing relation between salary and relationship
fig = plt.figure(figsize=(20,5))
ax_rel = sns.countplot(data=log_df, x='relationship', hue='salary', palette='Set2')
ax_rel.set_title("Salary (>50k(1) or <=50k(0)) vs Relationship ", loc='center', fontsiz
ax_rel.set_xlabel("Relationship")
ax_rel.set_ylabel(" ")
ax_rel.legend(loc="upper right")
```

```
<matplotlib.legend.Legend at 0x2924411dbe0>
```

Out[13]:



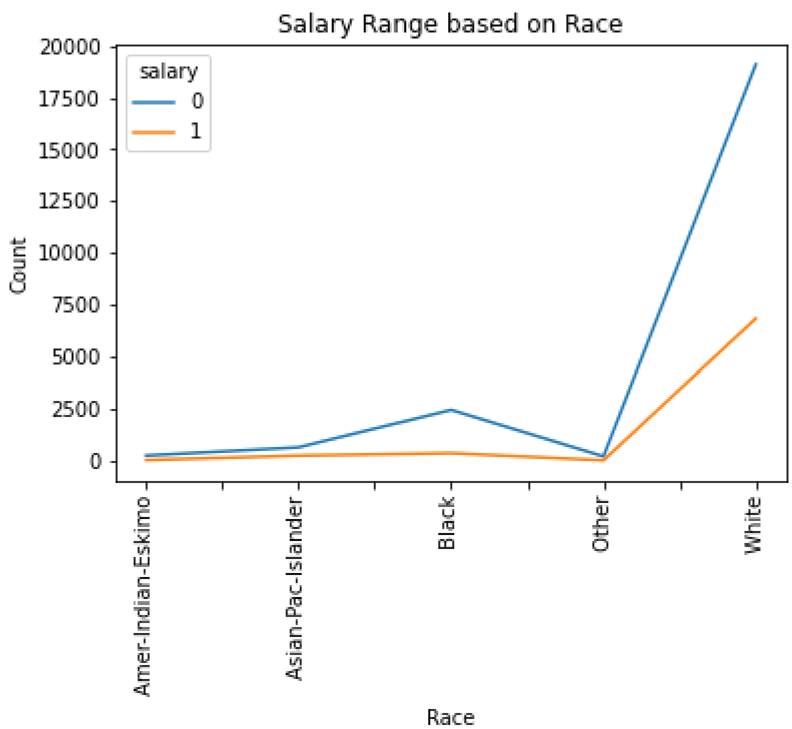
Conclusion: We can clearly see that relationships have impact on salaries.

Salary vs Race

In [26]:

```
# Plot showing relation between salary and race
#Race Analysis: This appears to have greater impact on the salary
pd.crosstab(log_df['race'], log_df['salary']).plot(kind='line', rot=90)
plt.xlabel('Race')
plt.ylabel('Count')
plt.title('Salary Range based on Race')
```

Out[26]: Text(0.5, 1.0, 'Salary Range based on Race')



Conclusion: We can see that it has an impact only if you are white. We can ignore this.

Salary vs Sex

In [15]:

```
# Plot showing correlation between salary and sex
#Compute the percentage of male and female earning below 50k.
male_df = log_df[log_df['sex'] == ' Male']
male_lessthan50k_df = log_df[(log_df['sex'] == ' Male') & (log_df['salary'] == 0)]
female_df = log_df[log_df['sex'] == ' Female']
female_lessthan50k_df = log_df[(log_df['sex'] == ' Female') & (log_df['salary'] == 0)]
```

```
print("Male & Female Percentage with salary <=50k: ", round((male_lessthan50k_df.shape[
male_df = log_df[log_df['sex'] == ' Male']
male_morethan50k_df = log_df[(log_df['sex'] == ' Male') & (log_df['salary'] == 1)]
female_df = log_df[log_df['sex'] == ' Female']
female_morethan50k_df = log_df[(log_df['sex'] == ' Female') & (log_df['salary'] == 1)]
print("Male & Female Percentage with salary >50k: ", round((male_morethan50k_df.shape[0
```

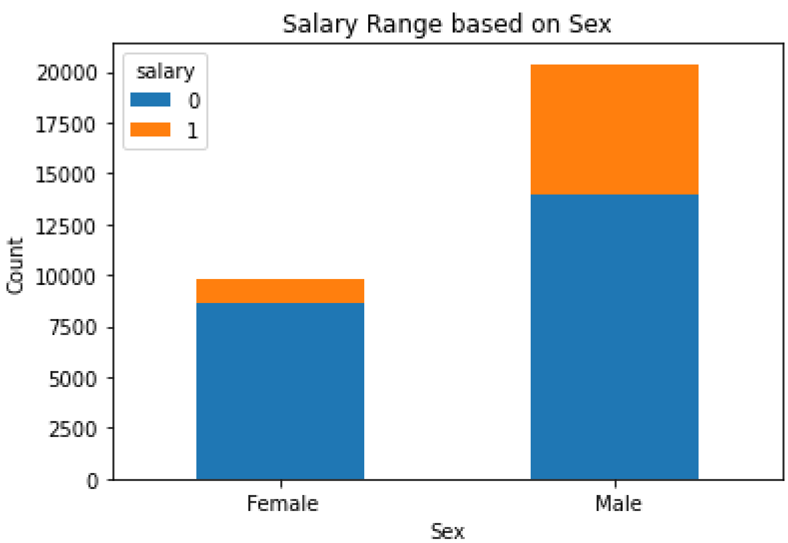
```
Male & Female Percentage with salary <=50k:  68.62 ,  88.63
Male & Female Percentage with salary >50k:  31.38 ,  11.37
```

In [23]:

```
#Sex Analysis: It can be seen that sex has an important role in determining the salary,
pd.crosstab(log_df['sex'], log_df['salary']).plot(kind='bar', rot=0, stacked=True)
plt.xlabel('Sex')
plt.ylabel('Count')
plt.title('Salary Range based on Sex')
```

Out[23]:

```
Text(0.5, 1.0, 'Salary Range based on Sex')
```



Conclusion: We can see that sex definitely defines your earning power.

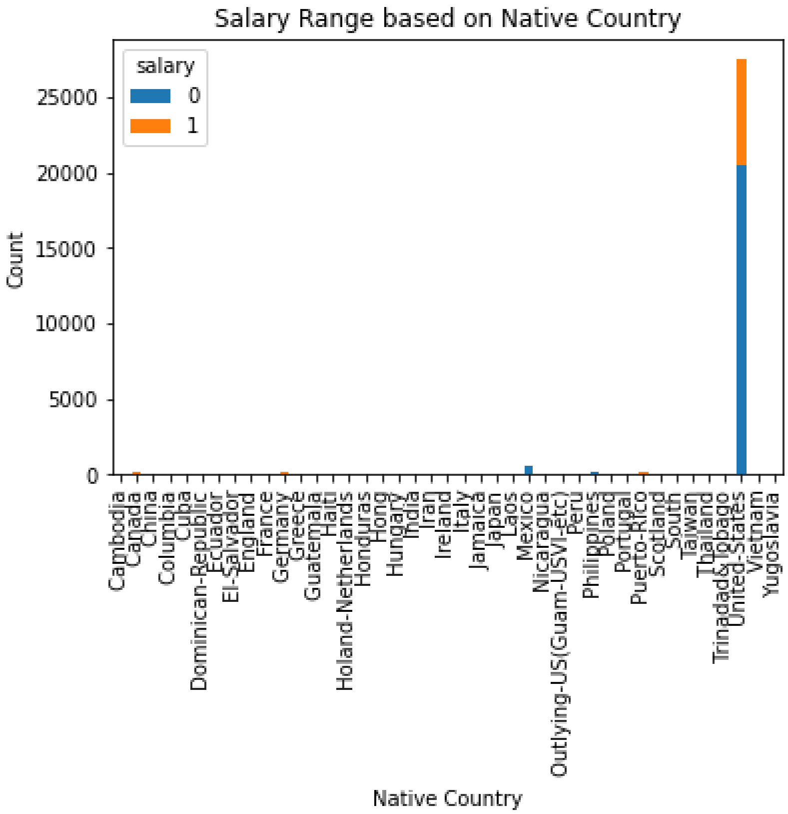
Salary vs Native country

In [24]:

```
# Plot showing rrelation between salary and native country
#Native Country Analysis: This doesn't appear to have a clear info of the impact on the
pd.crosstab(log_df['native-country'], log_df['salary']).plot(kind='bar', stacked=True)
plt.xlabel('Native Country')
plt.ylabel('Count')
plt.title('Salary Range based on Native Country')
```

Out[24]:

```
Text(0.5, 1.0, 'Salary Range based on Native Country')
```



Conclusion: The influence is limited to United states and can be ignored.

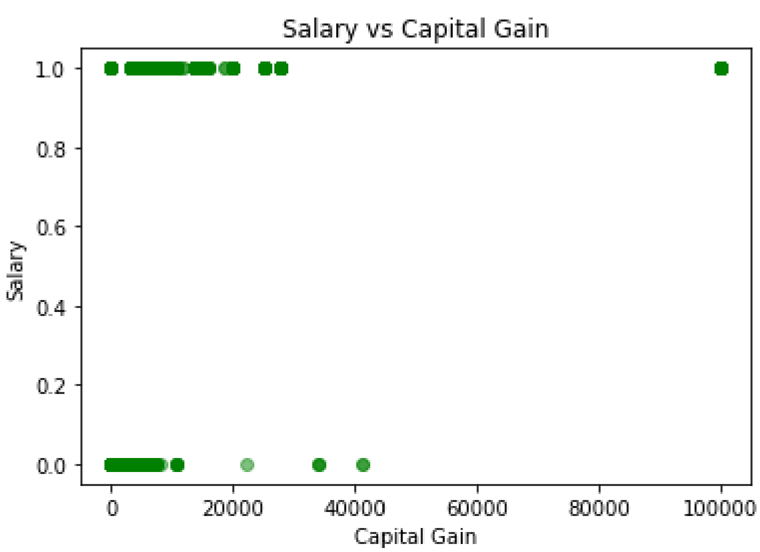
Salary vs Capital gain

In [115…

```
# Plot showing correlation between salary and capital-gain
#Capital Gain Analysis: This doesn't appear to have a clear info of the impact on the s
plt.scatter(log_df['capital-gain'], log_df['salary'], alpha=0.5, c='green')
plt.xlabel('Capital Gain')
plt.ylabel('Salary')
plt.title('Salary vs Capital Gain')
```

Out[115…

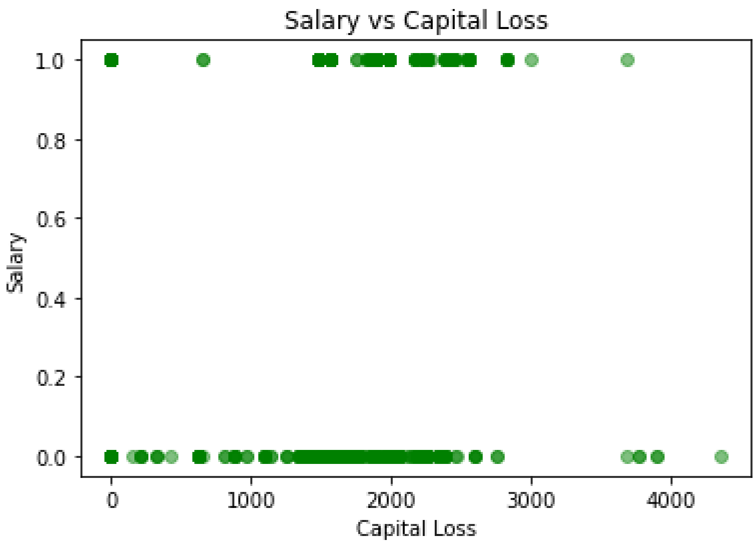
```
Text(0.5, 1.0, 'Salary vs Capital Gain')
```



Salary vs Capital loss

In [33]:
```
# Plot showing correlation between salary and capital-loss
plt.scatter(log_df['capital-loss'], log_df['salary'], alpha=0.5, c='green')
plt.xlabel('Capital Loss')
plt.ylabel('Salary')
plt.title('Salary vs Capital Loss')
```

Out[33]:
```
Text(0.5, 1.0, 'Salary vs Capital Loss')
```

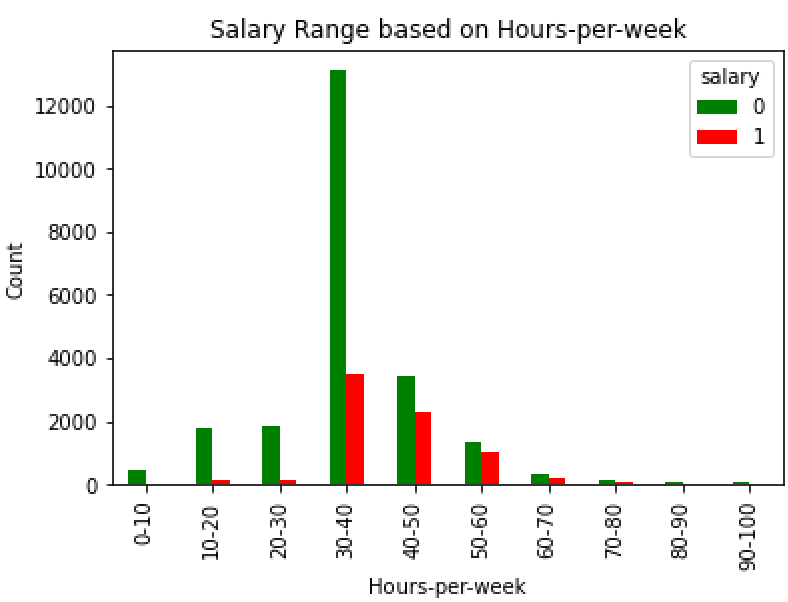


Conclusion:Both Capital gain and capital loss have a weak relationship with salary and can be ignored.

Salary vs Hours per week

In [77]:
```
# pie Plot showing correlation between salary and hours-per-week
#Hours-per-week Analysis: This doesn't appear to have a clear info of the impact on the
log_df['hours-per-week-grouped'] = pd.cut(log_df['hours-per-week'], bins=[0,10,20,30,40
pd.crosstab(log_df['hours-per-week-grouped'], log_df['salary']).plot(kind='bar',color=
plt.xlabel('Hours-per-week')
plt.ylabel('Count')
plt.title('Salary Range based on Hours-per-week')
```

Out[77]:
```
Text(0.5, 1.0, 'Salary Range based on Hours-per-week')
```



Conclusion: This can be considered a factor influencing salary.

Looking at the analysis so far, we can pick age, education, marital status, occupation, sex and hours per week as factors which influence the salary for our model.

Correlation between all the other factors.

In [17]:
```
# relation between salary and age, fnlwgt, education-num, capital-gain, capital-loss, h
fig = plt.figure(figsize=(8,8))
sns.heatmap(log_df.corr(), annot=True, cmap= 'Greens', linewidths=0.2)
```

Out[17]:
```
<AxesSubplot:>
```

In [99]:
```
fig = plt.figure(figsize=(20,5))
ax_age = sns.pairplot(log_df)
```

```
<Figure size 1440x360 with 0 Axes>
```

Conclusion: From the above heat map and pairplot, we can conclude that there is a further strong correlation between age, education, hours per week and are good factors for the model.

Now that the influencing factors are determined to build marketing profiles, a logistic regression model can be built on the training dataset, and the salary category (<=50k and >50k) can be predicted. Model accuracy can also be determined utilizing the confusion matrix.

Based on the above analysis, we can consider the following attributes/factors of the dataset are used for building the model.

In [151…

```
cols = ['age', 'education', 'marital-status', 'occupation', 'relationship', 'sex', 'hou
```

In [153…

```
X = log_df[cols]
X_dummies = pd.get_dummies(X)
y = log_df['salary']

# Split the dataset into train and test datasets, considering 70:30 ratio.
X_train, X_test, y_train, y_test = train_test_split(X_dummies, y, test_size=0.3, random
```

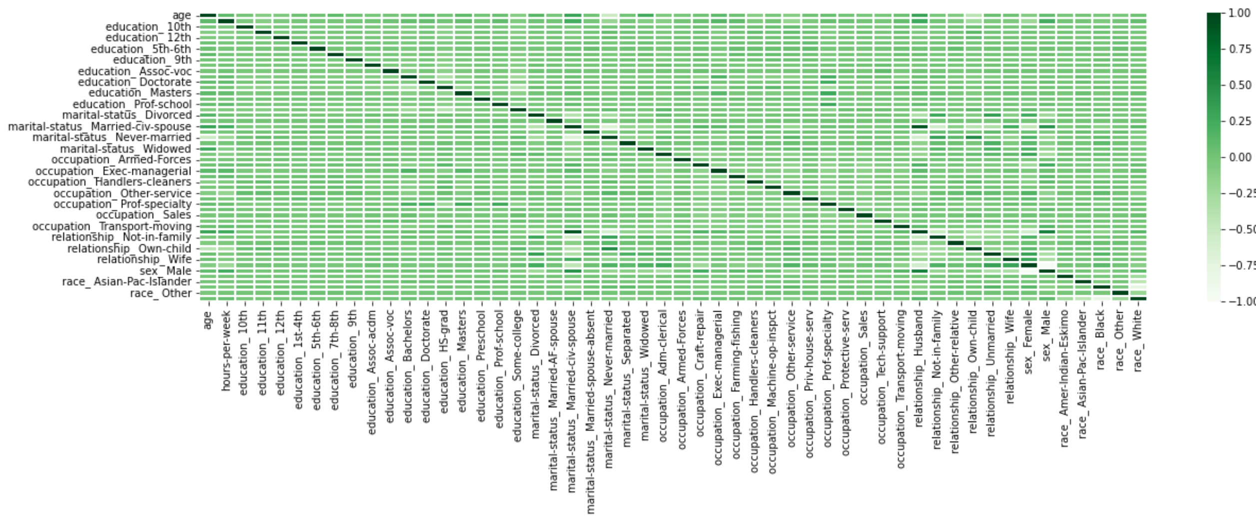
```
# Scale the train and test datasets
min_max_scaler = MinMaxScaler()
X_train_minmax = min_max_scaler.fit_transform(X_train)
X_test_minmax = min_max_scaler.fit_transform(X_test)

# Create a Logistic Regression model
logreg = LogisticRegression(max_iter=200)
y1 = y_train.astype(int)
logreg.fit(X_train_minmax, y1)

# Predicting the Test set results
y_pred=logreg.predict(X_test_minmax)
y_pred
```

Out[153…

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

In [154…

```
X_dummies = pd.get_dummies(X)
# heatmap
fig = plt.figure(figsize=(20,5))
ax_age = sns.heatmap(X_dummies.corr(), cmap= 'Greens', linewidths=0.2)
```



Salary predicted based on the selected factors

In [155…

```
pred_df = X_test.copy()
pred_df['salary'] = pd.Series(y_test, index=pred_df.index)
pred_df['predicted salary'] = pd.Series(y_pred, index=pred_df.index)
pred_df['predicted salary'] = pred_df['predicted salary'].map({0: '<=50k', 1: '>50k'})
pred_df[['salary','predicted salary']]
```

Out[155…

	salary	predicted salary
2135	0	<=50k
15639	0	<=50k
29059	0	<=50k
27523	0	<=50k
9280	0	<=50k

	salary	predicted salary
...	...	...
16826	0	<=50k
25246	0	<=50k
18980	1	<=50k
953	0	<=50k
30925	0	<=50k

9049 rows × 2 columns

Checking for the accuracy of the model

In [156…

```
y1_test = y_test.astype(int)
cnf_matrix = metrics.confusion_matrix(y1_test, y_pred)
cnf_matrix
```

Out[156…

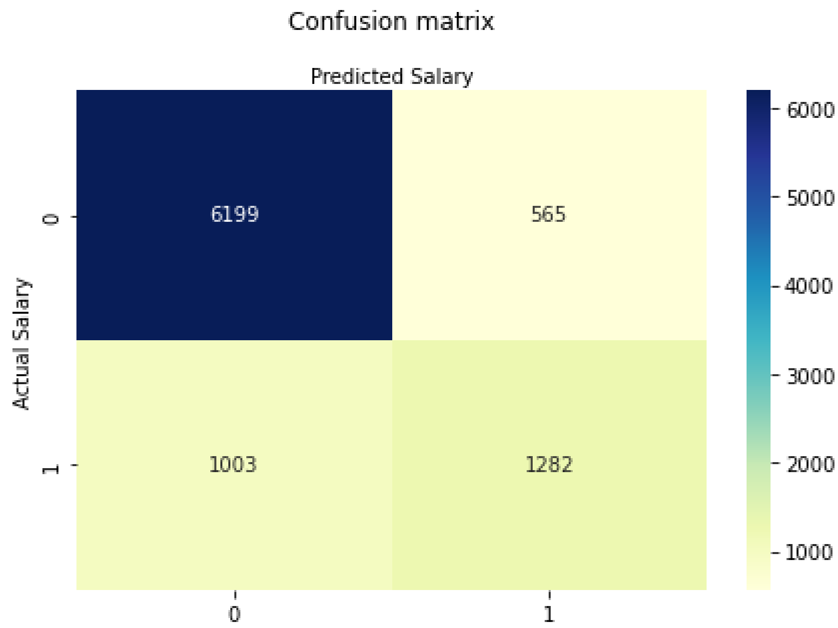
```
array([[6199,  565],
       [1003, 1282]], dtype=int64)
```

In [157…

```
class_names=[0,1] # name  of classes
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual Salary')
plt.xlabel('Predicted Salary')
```

Out[157…

```
Text(0.5, 257.44, 'Predicted Salary')
```



In [158…

```
print("Accuracy:", round(metrics.accuracy_score(y1_test, y_pred)*100, 2))
print("Precision:",round(metrics.precision_score(y1_test, y_pred)*100, 2))
```

```
Accuracy: 82.67
Precision: 69.41
```

Conclusion: We can see that based on the factors selected the model predicts the salary with an 82.67% accuracy and 69.41% precision.