MACHINE LEARNING (MATH 2319)

Project Phase 1

Predicting Credit Risk Using German Credit Risk Data

Name: Gaurav Diwan

Student IDs: s3799691

Table of Contents

1. Introduction

1.1 Objective

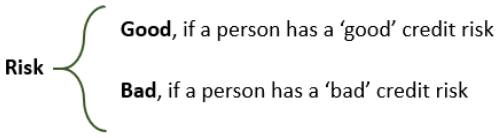
The topic that has been chosen for this project is the 'German Credit Risk'. The objective, here, is to predict whether a person has good or bad credit risks, based on a set of attributes. The idea for this data has come from UCI Machine learning repository and the data has been sourced from Kaggle at 'https://www.kaggle.com/uciml/german-credit'. The requirements for this project are branched into two phases. The first phase concentrates on data analysis and preprocessing and provides a detailed descriptive statistical analysis of the data. The second phase will present different machine learning algorithms to demonstrate the best method. This report scrutinizes the phase 1 of the project, where section 2 covers the Data preprocessing part and section 3 covers the data exploration.

1.2 Dataset

The original data present in the UCI machine learning repository had many complicated categories and symbols and hence, the author has converted it into a readable CSV file by removing few columns with uncertain/unclear descriptions, and provided it in Kaggle as 'german_credit_data.csv'. The processed dataset has 1000 observations, 9 descriptive features and 1 target feature. For this phase, the analysis has been taken out on this dataset. In phase 2, a detailed performance analysis and comparison of the algorithms would be held.

1.2.1 Target Feature

The target feature for this dataset is 'Risk' variable. Risk has two categories and therefore, can be classified as binary. It is this variable that will define a 'good' or 'bad' credit for a person. The classification can be shown as-



1.2.2 Descriptive Feature

The description of the attributes that has been chosen for descriptive features are readily comprehensible and are as follows-

Age - numeric
Sex - text: male, female
Job - numeric: -0-unskilled and non-resident, 1-unskilled and resident, 2-skilled, 3-highly skilled
Housing - text: own, rent, or free
Saving accounts - text: little, moderate, quite rich, rich
Checking account - text: little, moderate, rich
Duration - numeric: in months

Purpose - text: car, furniture/equipment, radio/TV, domestic appliances, repairs, education, business, vacation/other.

2. Data Pre-processing

2.1 Preliminaries

Firstly, all the necessary python libraries have been imported. The dataset is downloaded and moved to the home directory and is, then read into python by using respective CSV command. The dataset is named as 'CreditData' for all the further analysis.

In [1]:
```
#Importing necessary libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import seaborn as sns
import matplotlib.style as sty
from pylab import rcParams
import matplotlib
```

In [2]:
```
#Importing the dataset into Python environment

CreditData = pd.read_csv('german_credit_data.csv')
```

2.2 Data Cleaning and Transformation

The dimensions and types of all the feature attributes have been checked in order to confirm whether it matches the description as mentioned in the source documentation.

In [3]:
```
print('Dimensions of the German Credit Risk Data', CreditData.shape)  #Size of the dataset (No.of rows, No. of columns)
```

```
('Dimensions of the German Credit Risk Data', (1000, 10))
```

In [4]:
```
# Exploring the concise summary of the dataset
print(CreditData.dtypes)
```

```
Age                 int64
Sex                object
Job                 int64
Housing            object
Saving accounts    object
Checking account   object
Credit amount       int64
Duration            int64
Purpose            object
Risk               object
dtype: object
```

Unique values are calculated to outline the number of different values/objects contained by each of the features in the dataset.

In [5]:
```
#Looking for the unique values
print(CreditData.nunique())
```

```
Age                 53
Sex                  2
Job                  4
Housing              3
Saving accounts      4
Checking account     3
Credit amount      921
Duration            33
Purpose              8
Risk                 2
dtype: int64
```

The head() function gave an overview, to check whether the dataset has been correctly read into the python environment. It clearly shows some NaN values for two of the features.

In [6]:
```
#Reading the first 6 rows or observation of the dataset
CreditData.head(6)
```

Out[6]:

	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose	Risk
0	67	male	2	own	NaN	little	1169	6	radio/TV	good
1	22	female	2	own	little	moderate	5951	48	radio/TV	bad
2	49	male	1	own	little	NaN	2096	12	education	good
3	45	male	2	free	little	little	7882	42	furniture/equipment	good
4	53	male	2	free	little	little	4870	24	car	bad
5	35	male	1	free	NaN	NaN	9055	36	education	good

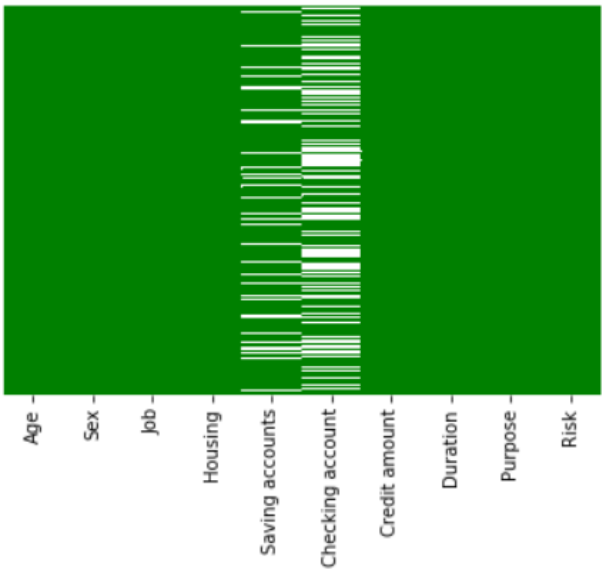
Hence, examined all those features that are having missing/NaN values, in the dataset. It is clear that only two of the features have missing/NaN values, i.e., 'Checking account' and 'Saving accounts'.

In [7]:
```
# Finding columns that have maximum missing values-

Total = CreditData.isnull().sum().sort_values(ascending=False)
Percent = (CreditData.isnull().sum()/CreditData.isnull().count()).sort_values(ascending=False)
Missing_Data = pd.concat([Total, Percent], axis=1, keys=['Total','Percent'])
print(Missing_Data)
```

```
                  Total  Percent
Checking account    394    0.394
Saving accounts     183    0.183
Risk                  0    0.000
Purpose               0    0.000
Duration              0    0.000
Credit amount         0    0.000
Housing               0    0.000
Job                   0    0.000
Sex                   0    0.000
Age                   0    0.000
```

Alternatively, the missing values, in the dataset, can also be visualized using 'Heatmap'. The whitespaces in the Heatmap represents the missing values. The previous is preferred more as Heatmap doesn't give the exact number of missing values present.

```
In [8]: sns.heatmap(CreditData.isnull(),yticklabels=False,cbar=False,cmap='ocean')
        # White spaces in the heatmap are the missing values from the data

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x55aaba8>
```



Assumption on missing values: In general practice, the missing values are eliminated from the dataset. But here, looking practically, it is not always necessary that a person would have a 'Saving account'. Similarly, is the case with the 'Checking account'. A person can have either of the accounts. So, by taking this assumption into consideration, the 'NaN' values present in these two features have not been removed or imputed.

Also, on examining the dataset, it has been found that there are 'No Whitespaces' to be stripped from any of the observations.

2.2.1 Numerical Features

The summary statistics for the numerical features are calculated. The information that can be inferred from the output is that, the feature 'Job' should not be considered as a numerical attribute. Reason being, practically 'Job' can not have a summary statistic as shown below. 'Job' should be a categorical attribute. Whereas, all the other numerical features are showing relevant statistics that are in accordance with their observations/values.

In [11]:
```
print(CreditData.describe(include='int64'))
```

```
                Age          Job  Credit amount     Duration
count  1000.000000  1000.000000    1000.000000  1000.000000
mean     35.546000     1.904000    3271.258000    20.903000
std      11.375469     0.653614    2822.736876    12.058814
min      19.000000     0.000000     250.000000     4.000000
25%      27.000000     2.000000    1365.500000    12.000000
50%      33.000000     2.000000    2319.500000    18.000000
75%      42.000000     2.000000    3972.250000    24.000000
max      75.000000     3.000000   18424.000000    72.000000
```

2.2.2 Categorical Features

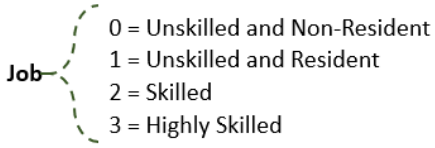
The previous code gave only the number of different values/objects present in each of the features. But for any categorical feature, it is important to know the names of the objects that define a particular attribute. So, for further references, the names of all those unique objects are shown in the output below, for each of the categorical features.

In [12]:
```
#Checking for all the unique value in the categorical data
categoricalcolumn = ['Sex','Housing','Saving accounts','Checking account','Purpose','Risk']
for col in categoricalcolumn:
    print('Unique values for ' + col)
    print(CreditData[col].unique())
    print('')
```

```
Unique values for Sex
['male' 'female']

Unique values for Housing
['own' 'free' 'rent']

Unique values for Saving accounts
[nan 'little' 'quite rich' 'rich' 'moderate']

Unique values for Checking account
['little' 'moderate' nan 'rich']

Unique values for Purpose
['radio/TV' 'education' 'furniture/equipment' 'car' 'business'
 'domestic appliances' 'repairs' 'vacation/others']

Unique values for Risk
['good' 'bad']
```

2.2.3 Converting Numerical features into Categorical features (as per the requirement)

As discussed above, the descriptive feature 'Job' should be a categorical attribute instead of numerical. Hence, converting 'Job' into categories as-



In [12]:
```
#Changing Job Variables
CreditData['Job'].replace(0,'Unskilled and Non resident',inplace=True)
CreditData['Job'].replace(1,'Unskilled and Resident',inplace=True)
CreditData['Job'].replace(2,'Skilled',inplace=True)
CreditData['Job'].replace(3,'Highly Skilled',inplace=True)

CreditData.head()
```

Out[12]:

	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose	Risk
0	67	male	Skilled	own	NaN	little	1169	6	radio/TV	good
1	22	female	Skilled	own	little	moderate	5951	48	radio/TV	bad
2	49	male	Unskilled and Resident	own	little	NaN	2096	12	education	good
3	45	male	Skilled	free	little	little	7882	42	furniture/equipment	good
4	53	male	Skilled	free	little	little	4870	24	car	bad

Also, there is one more feature that can be categorized, i.e., 'Age'. Although it is not necessary to perform this task, but grouping 'Age' would be helpful for giving better insight while doing the descriptive statistical analysis or in other words, while visualizing the data.

In [13]:
```
# Creating group on the basis of age
CreditData['Age_Group'] = np.nan

lst = [CreditData]

for col in lst:
    col.loc[(col['Age'] > 18) & (col['Age'] <= 27), 'Age_Group'] = '18-27'
    col.loc[(col['Age'] > 28) & (col['Age'] <= 37), 'Age_Group'] = '28-37'
    col.loc[(col['Age'] > 38) & (col['Age'] <= 47), 'Age_Group'] = '38-47'
    col.loc[(col['Age'] > 48) & (col['Age'] <= 57), 'Age_Group'] = '48-57'
    col.loc[col['Age'] > 58, 'Age_Group']='58+'

CreditData.head()
```

Out[13]:

	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose	Risk	Age_Group
0	67	male	Skilled	own	NaN	little	1169	6	radio/TV	good	58+
1	22	female	Skilled	own	little	moderate	5951	48	radio/TV	bad	18-27
2	49	male	Unskilled and Resident	own	little	NaN	2096	12	education	good	48-57
3	45	male	Skilled	free	little	little	7882	42	furniture/equipment	good	38-47
4	53	male	Skilled	free	little	little	4870	24	car	bad	48-57

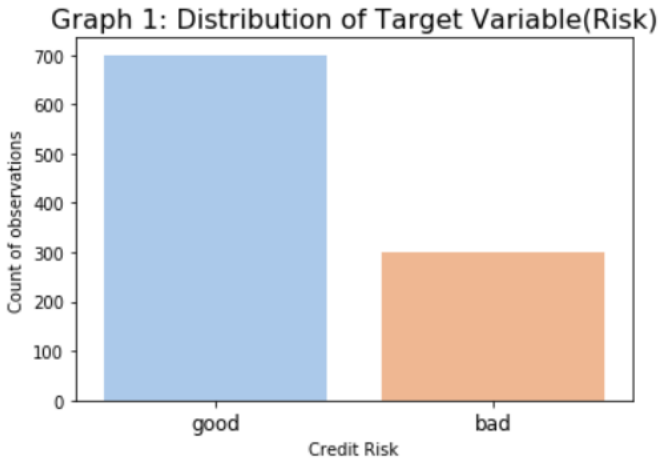
3. Data Exploration

3.1 Univariate Visualization

To apprehend the dataset statistically, countplot() is used for the categorical features and distplot() for the numerical feature, under the seaborn library. Also, for clearer visualization, pie chart is used for 'Age_Group' and 'Job' categorical features. Since the dataset contains only 1000 observations, countplot() is useful compared to barplot(), as countplot() represents the absolute count of observations for each of the categories under a categorical feature. The distplot() plots the histogram distribution for a numerical variable of the data. Pie chart is used as it gives the numerical proportion for a categorical variable.

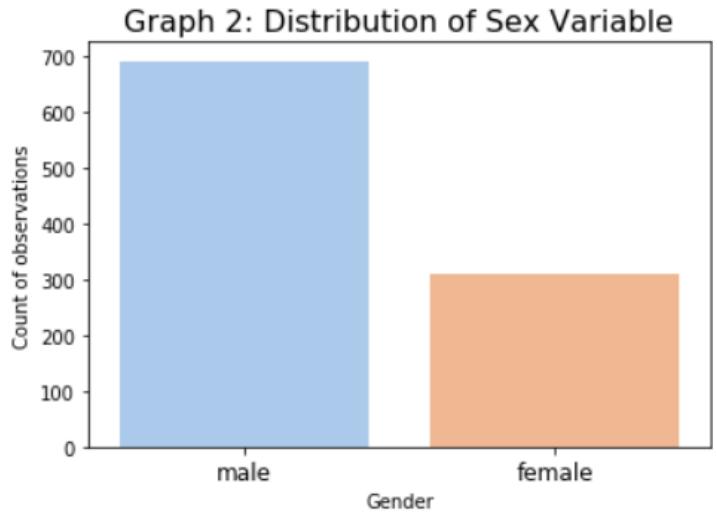
Graph 1 shows the target feature 'Risk', for the dataset. It can be seen that the people with good credit (approx. 700) are more in number as compared to that with bad credit (approx. 300).

In [14]:
```
sns.countplot('Risk',data = CreditData,palette="pastel")
plt.title('Graph 1: Distribution of Target Variable(Risk)',fontsize=16)
plt.ylabel('Count of observations')
plt.xlabel('Credit Risk')
plt.xticks(fontsize = 12)
plt.show()
```



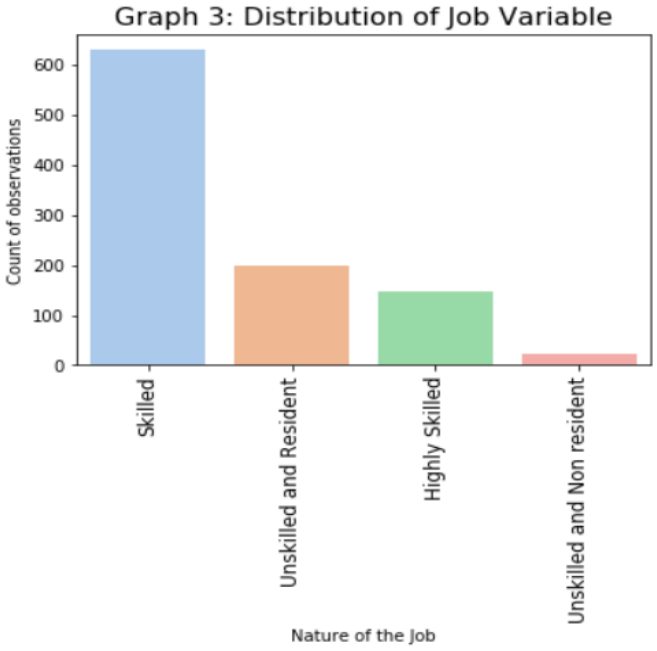
Graph 2 gives the count for the number of males and females for the dataset. With close to 700 count, the number of males are more than that of females with the count of ~300.

In [15]:
```
sns.countplot('Sex',data = CreditData,palette="pastel")
plt.title('Graph 2: Distribution of Sex Variable',fontsize=16)
plt.ylabel('Count of observations')
plt.xlabel('Gender')
plt.xticks(fontsize = 12)
plt.show()
```



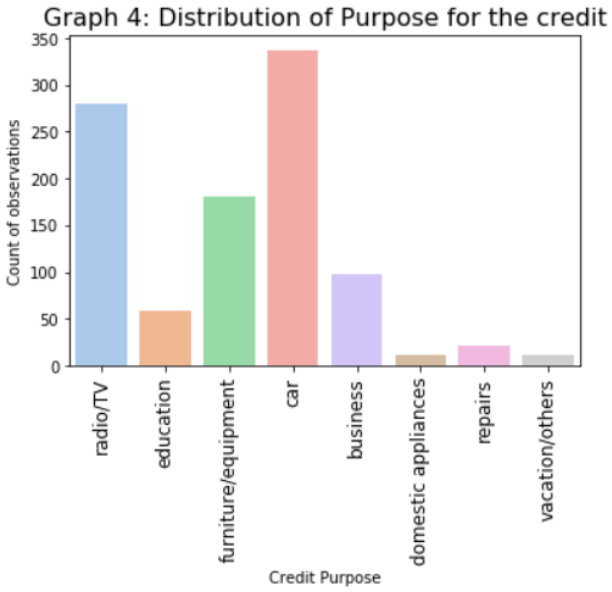
Graph 3 shows the nature of the 'Job' on which the people are classified. The highest number of people are employed under the 'Skilled' category (more than 600), whereas, the lowest are employed under the 'Unskilled and Non-resident' (~20).

In [16]:
```
sns.countplot('Job',data = CreditData,palette="pastel")
plt.title('Graph 3: Distribution of Job Variable',fontsize=16)
plt.xticks(rotation=90,fontsize = 12)
plt.ylabel('Count of observations')
plt.xlabel('Nature of the Job')
plt.show()
```



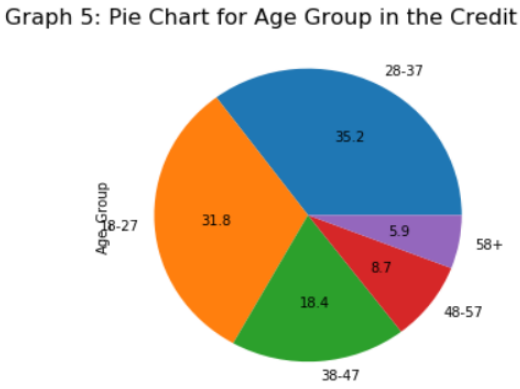
Graph 4 shows the distribution of various purposes for which the credit is taken. With near to 350 count followed by the count of ~270, it can be said that 'car' and 'radio/TV' are the most common reasons for taking the credits, respectively.

In [17]:
```
sns.countplot('Purpose',data = CreditData,palette="pastel")
plt.title('Graph 4: Distribution of Purpose for the credit',fontsize=16)
plt.ylabel('Count of observations')
plt.xlabel('Credit Purpose')
plt.xticks(rotation=90,fontsize = 12)
plt.show()
```



Graph 5, shows the proportional distribution of 'Age_Group'. The maximum number of populations in the dataset fall under the Age group of '28-37', as it has the highest percentage of 35.2%. The lowest number of people taking the credit are old-aged (58+) as it shares only 5.9% of the total.

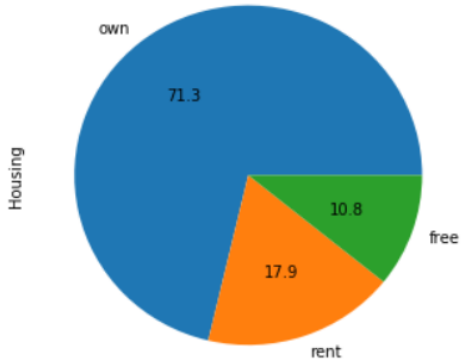
In [20]:
```
CreditData['Age_Group'].value_counts().plot(kind='pie',autopct='%.1f')
plt.title('Graph 5: Pie Chart for Age Group in the Credit Dataset',fontsize=16)
plt.show()
rcParams['figure.figsize']= 5,5
```



Here, in Graph 6, it can be seen that maximum people owns properties/houses (with 71.3% of the total). Whereas, least number of people are staying for free (with 10.8%).

In [21]:
```
CreditData['Housing'].value_counts().plot(kind='pie',autopct='%.1f')
plt.title('Graph 6: Pie Chart for the Housing variable in the Credit Dataset',fontsize=16)
plt.show()
rcParams['figure.figsize']= 5,5
```

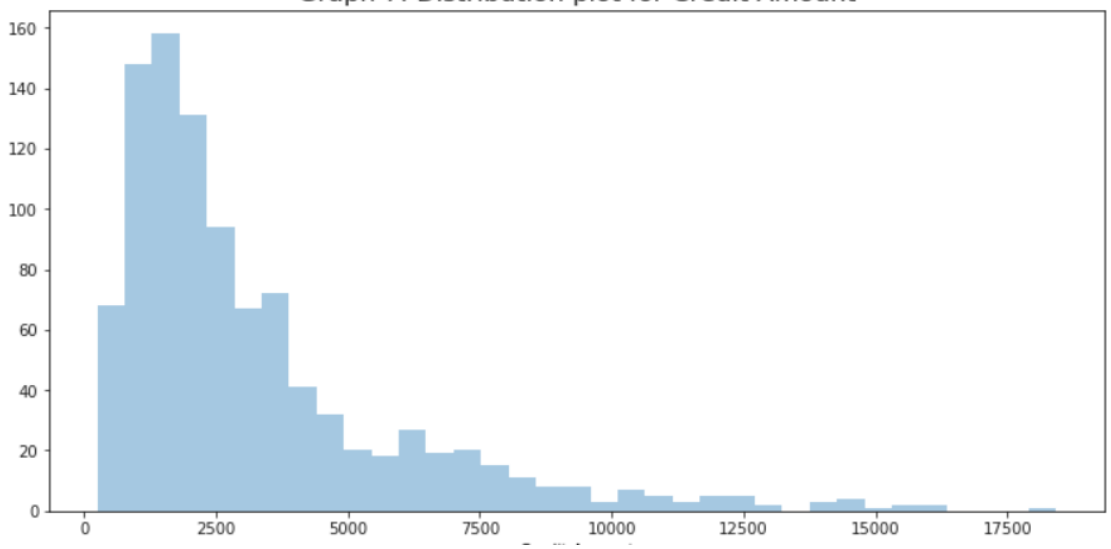
Graph 6: Pie Chart for the Housing variable in the Credit Dataset



From Graph 7, it is clear that the credit amount that is most commonly taken for any of the purposes, can be ranged between 'DM 1000 -DM 3000', approximately. It can also be said that, for the given dataset, the credit amount of more than DM 7500 are rarely been taken.

In [22]:
```
plt.figure(figsize = (12,6))
plt.title('Graph 7: Distribution plot for Credit Amount',fontsize=16)
sns.set_color_codes("colorblind")
sns.distplot(CreditData['Credit amount'],kde=False)
plt.xlabel('Credit Amount')
plt.show()
```

Graph 7: Distribution plot for Credit Amount



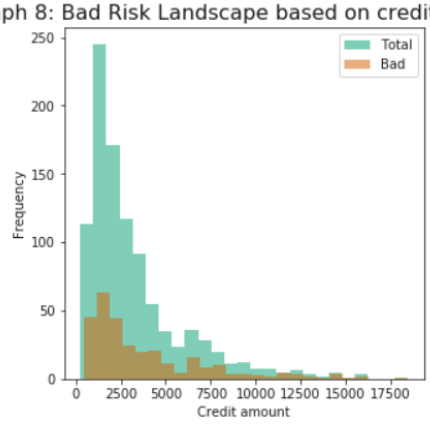
3.2 Multivariate Visualization

3.2.1 Numerical features segregated by Target feature 'Risk'

Graph 8 and Graph 9 shows that the histogram is right (positive) skewed, in other words, the mean is to the right of its median. Therefore, the bad credit will increase as the credit amount increases. Whereas, the good credit will still be categorized as good even though the credit amount increases. Additionally, the taller bars indicates that the credit amount range of DM 1000-DM 5000 is more likely for good risk than for the bad risk with smaller bars.

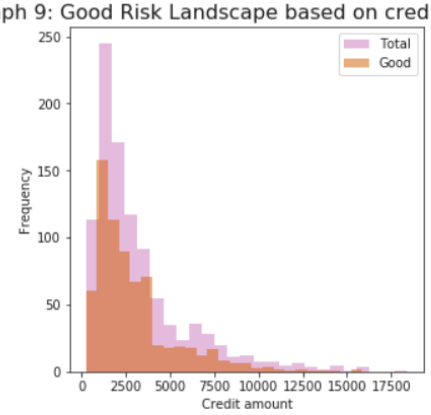
In [30]:
```
bins = 25
plt.hist(CreditData['Credit amount'],bins = bins, color='g',label = 'Total',alpha=0.5)
plt.hist(CreditData['Credit amount'][CreditData['Risk']=='bad'], bins = bins, color='r',label = 'Bad',alpha=0.5)
plt.xlabel('Credit amount'); plt.ylabel('Frequency')
plt.title('Graph 8: Bad Risk Landscape based on credit amount',fontsize=16)
plt.legend();plt.show()
```

Graph 8: Bad Risk Landscape based on credit amount



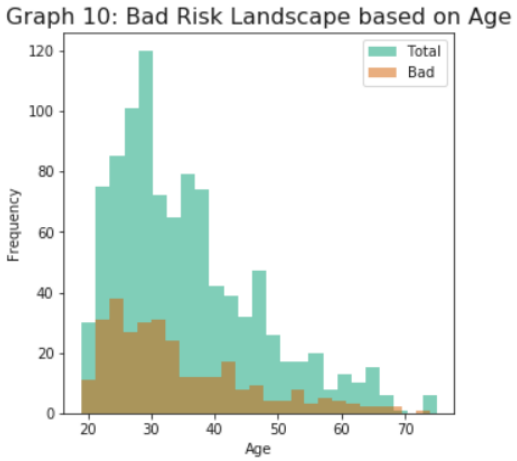
In [31]:
```
bins = 25
plt.hist(CreditData['Credit amount'],bins = bins, color='m',label = 'Total',alpha=0.5)
plt.hist(CreditData['Credit amount'][CreditData['Risk']=='good'], bins = bins, color='r',label = 'Good',alpha=0.5)
plt.xlabel('Credit amount'); plt.ylabel('Frequency')
plt.title('Graph 9: Good Risk Landscape based on credit amount',fontsize=16)
plt.legend();plt.show()
```

Graph 9: Good Risk Landscape based on credit amount

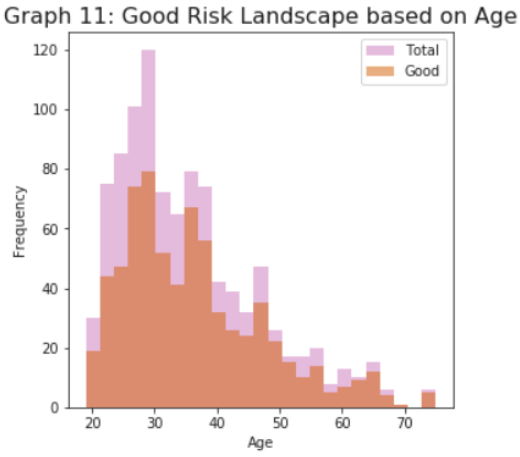


Similarly, when the 'Age' is taken under consideration, Graph 10 and Graph 11 are still positively skewed. It can be infer that the people aged under 40 are most likely to have good credit risk than bad credit risk.

In [32]:
```
bins = 25
plt.hist(CreditData['Age'],bins = bins, color='g',label = 'Total',alpha=0.5)
plt.hist(CreditData['Age'][CreditData['Risk']=='bad'], bins = bins, color='r',label = 'Bad',alpha=0.5)
plt.xlabel('Age'); plt.ylabel('Frequency')
plt.title('Graph 10: Bad Risk Landscape based on Age',fontsize=16)
plt.legend();plt.show()
```



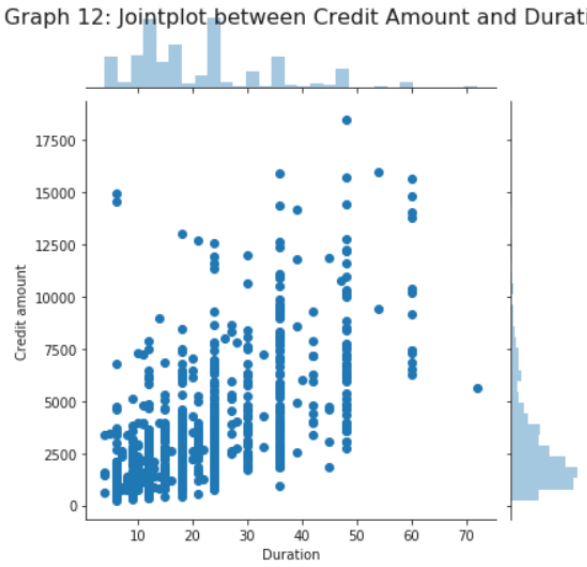
In [34]:
```
bins = 25
plt.hist(CreditData['Age'],bins = bins, color='m',label = 'Total',alpha=0.5)
plt.hist(CreditData['Age'][CreditData['Risk']=='good'], bins = bins, color='r',label = 'Good',alpha=0.5)
plt.xlabel('Age'); plt.ylabel('Frequency')
plt.title('Graph 11: Good Risk Landscape based on Age',fontsize=16)
plt.legend();plt.show()
```



3.2.2 Pairwise Joint plot between two numerical features

Graph 12 displays a Joint plot, with the scatter plot representing the correlation between the two numerical features, i.e., 'Duration' and 'Credit amount'. It can be clearly noticed from the graph that, the lesser the credit amount, the smaller is the duration (in months).

In [39]:
```
sns.jointplot(x='Duration',y='Credit amount',data=CreditData)
plt.suptitle('Graph 12: Jointplot between Credit Amount and Duration',fontsize=16)
plt.show()
```

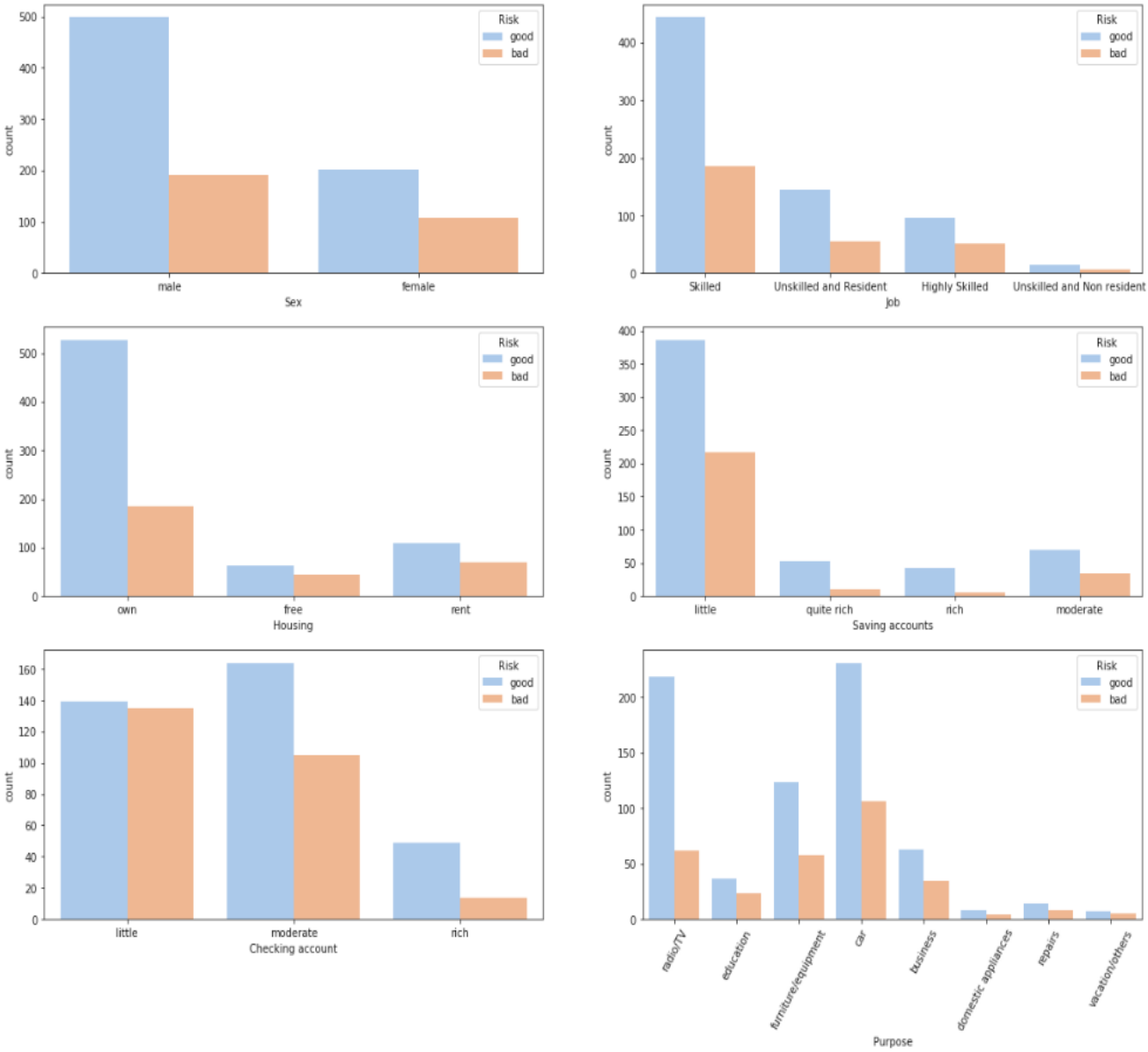


3.2.3 Categorical features segregated by Target feature 'Risk'

Graph 13 displays various comparison count plots for all the 6 categorical features present ion the dataset. With the first subplot of 'Sex' it can be said that 'male' has more good credits as compared to that of 'females'. Second subplot indicates that people working under 'skilled' job category tend to have good credit risk followed by those working under 'Unskilled and resident'. Third subplot displays that people who owns the house/property tend to have good credit records. With the fourth and fifth subplots, it can be said that those people having 'little' 'saving account' and those having 'moderate' 'checking account' have more good credit risk. Whereas people having 'little' 'checking account' have bad credit risks. The last subplot indicates that people taking credit for the 'car' and 'radio/TV' tend to have good credit risk but also higher bad credit risk as compared to other purposes.

In [41]:
```
Subset = CreditData[['Sex','Job','Housing','Saving accounts','Checking account','Purpose','Risk']]
f,axes = plt.subplots(3,2,figsize=(20,15),facecolor= 'white')
f.suptitle('Graph 13: Frequency of Categorical Variables by Target Variable',fontsize=16)
ax1 = sns.countplot(x="Sex",hue="Risk",data=Subset,palette="pastel",ax=axes[0,0])
ax2 = sns.countplot(x="Job",hue="Risk",data=Subset,palette="pastel",ax=axes[0,1])
ax3 = sns.countplot(x="Housing",hue="Risk",data=Subset,palette="pastel",ax=axes[1,0])
ax4 = sns.countplot(x="Saving accounts",hue="Risk",data=Subset,palette="pastel",ax=axes[1,1])
ax5 = sns.countplot(x="Checking account",hue="Risk",data=Subset,palette="pastel",ax=axes[2,0])
ax6 = sns.countplot(x="Purpose",hue="Risk",data=Subset,palette="pastel",ax=axes[2,1])
plt.xticks(rotation=60)
```

Graph 13: Frequency of Categorical Variables by Target Variable

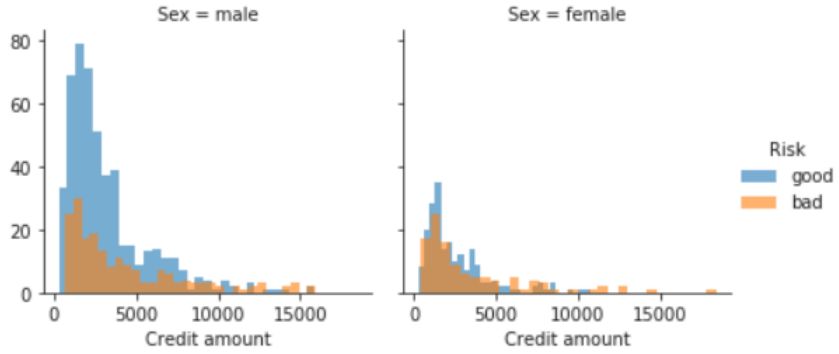


3.2.4 Relationship between categorical and numerical features

The below histogram indicates that 'male' taking 'credit amount' ranged between DM 1000 to DM 5000 tend to have higher 'good' credit risk as compared to that of 'female' taking credit amount between the same range. Whereas, the bad credit risk is almost same for both males and females.

```
In [46]: m = sns.FacetGrid(CreditData, col='Sex', hue='Risk')
         m.map(plt.hist, 'Credit amount', alpha = 0.6, bins = 30)
         m.add_legend()
```

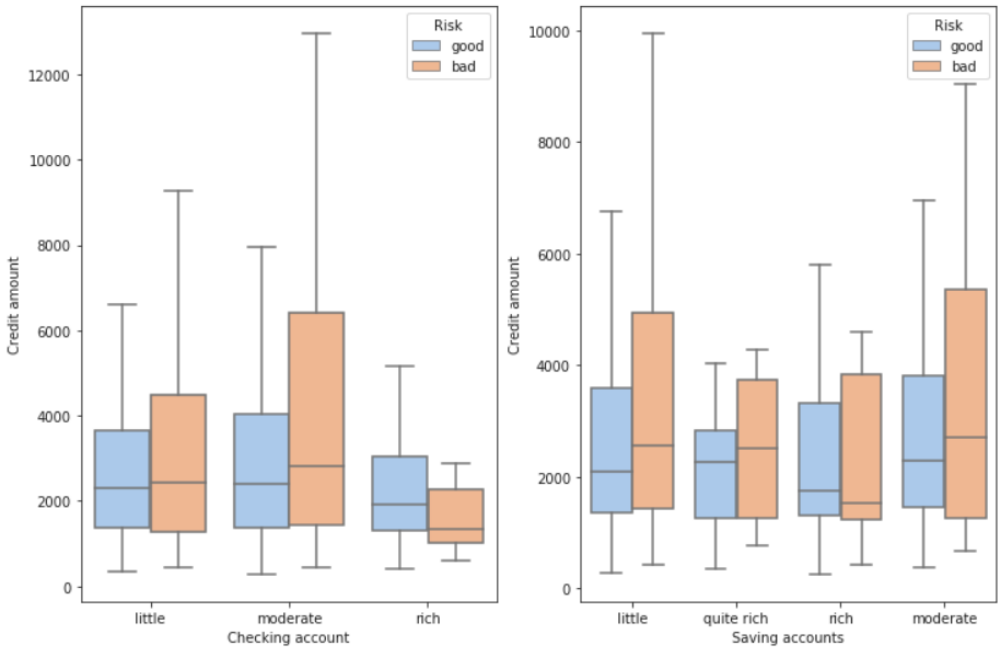
```
Out[46]: <seaborn.axisgrid.FacetGrid at 0x12c5a518>
```



When checking the 'Risk' based upon credit amount and types of account, it can be seen that people having 'moderate' checking account tend to take more credit amount but have higher bad credit risk. Whereas, for those taking more credit amount with 'little' and 'moderate' saving account tend to have higher bad credit risk as compared to the good credit risk for the same category.

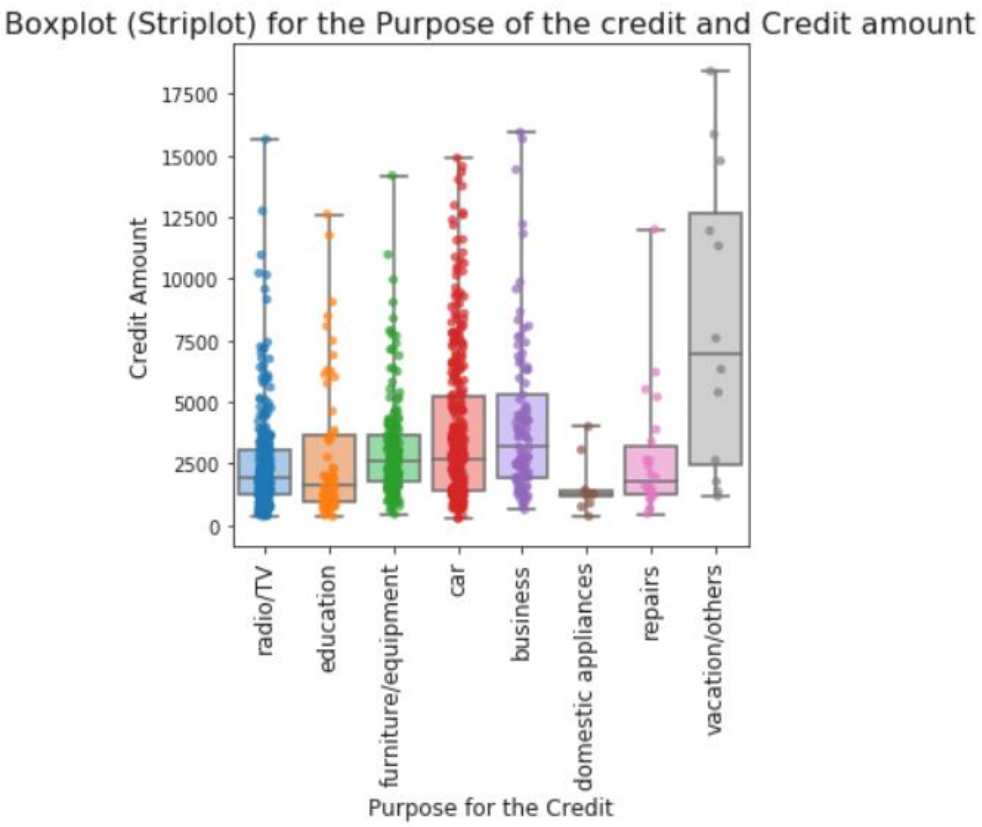
```
In [47]: fig,(ax1,ax2) = plt.subplots(ncols=2,figsize=(12,8))
         s=sns.boxplot(ax=ax1,x='Checking account',y='Credit amount',hue='Risk',data=CreditData,palette="pastel",showfliers=False)
         s=sns.boxplot(ax=ax2,x='Saving accounts',y='Credit amount',hue='Risk',data=CreditData,palette="pastel",showfliers=False)
         fig.suptitle('Box Plot comparison for Checking account and Saving accounts based on Credit Amount and Risk', fontsize=16)
         plt.show()
```

Box Plot comparison for Checking account and Saving accounts based on Credit Amount and Risk



The below boxplot (strip plot) shows all the purposes for which the credit amount has been taken. It is very much clear that highest credit amount has been taken for vacation/others purposes, whereas least credit amount is taken for domestic appliances. Credit amount taken for 'Car' and 'Business' are approximately the same and second highest as compared to other purposes.

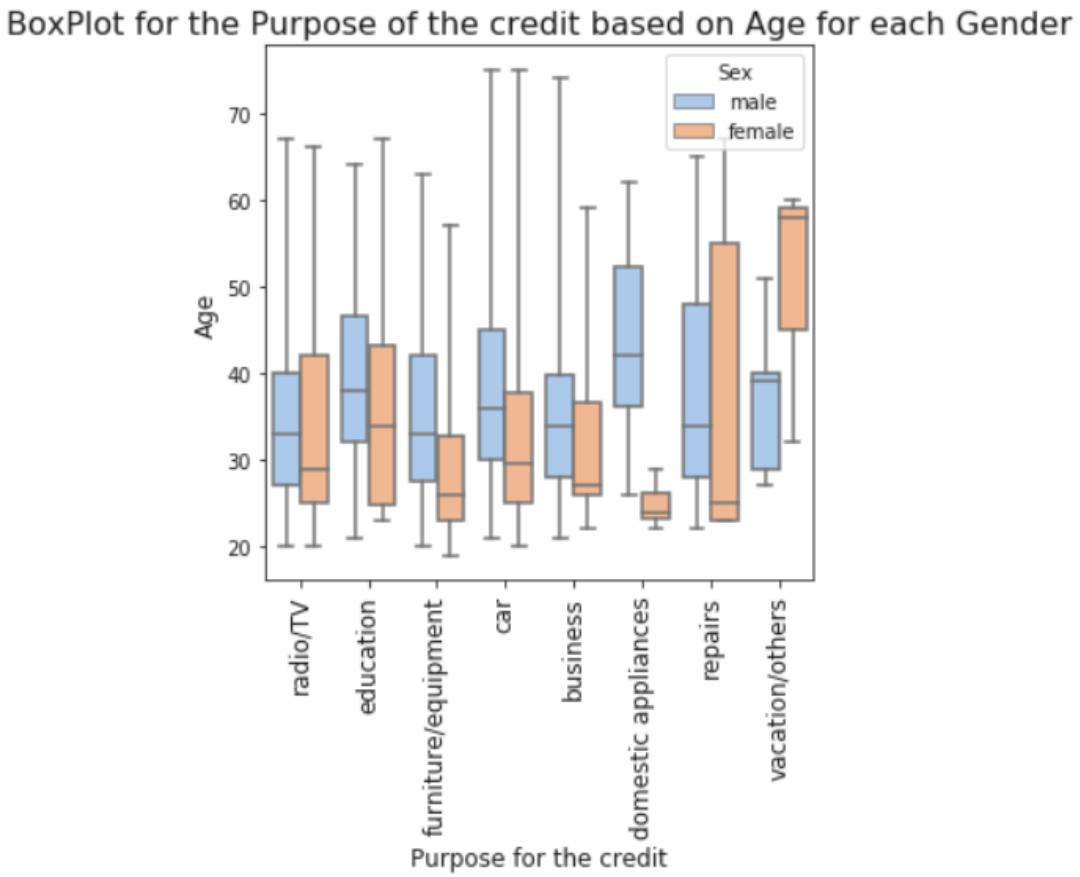
In [48]:
```
ax = sns.boxplot(x="Purpose", y="Credit amount", data=CreditData, whis=np.inf,palette="pastel")
ax = sns.stripplot(x="Purpose",y="Credit amount",data=CreditData,alpha=0.7)
plt.xlabel('Purpose for the Credit',fontsize = 12)
plt.ylabel('Credit Amount',fontsize = 12)
plt.xticks(rotation=90,fontsize = 12)
plt.title('Boxplot (Striplot) for the Purpose of the credit and Credit amount',fontsize=16)
plt.show()
```



3.2.5 Relationship between two categorical and one numerical feature

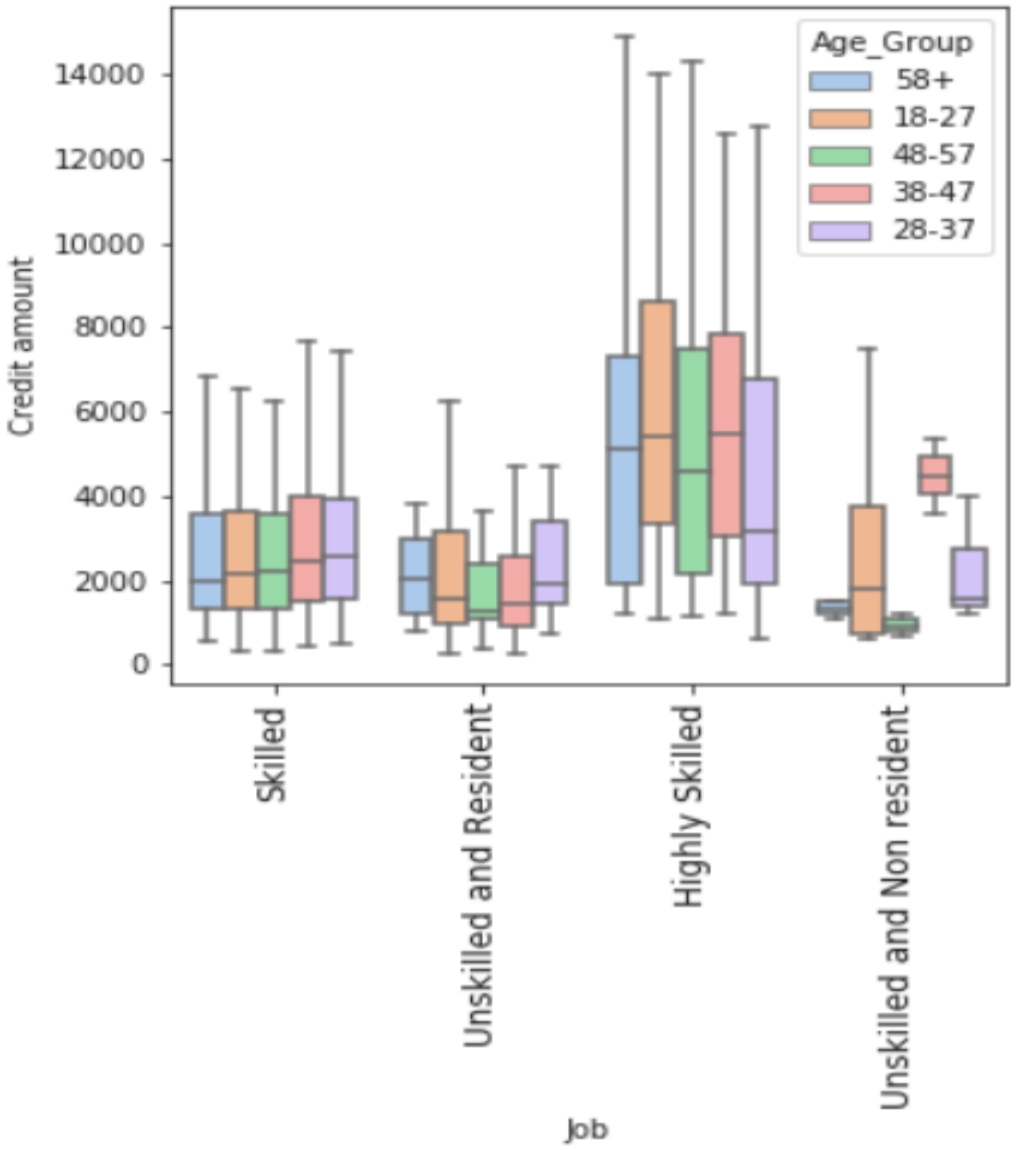
The boxplot below, shows the purposes for the credit, among people of different age. It is drawing this relationship on the basis of 'Sex' as well. It is noticed that people aged more than 25 takes credits for 'repairs' purpose and among those people, the females are more as compared to males. Whereas, credits taken for 'domestic appliances' are taken by the people aged either more than 35 or less than 25, among such people, males are more than females. Older females aged more than 45 take credit for vacation/others purposes.

In [49]:
```
ax = sns.boxplot(x="Purpose", y="Age", data=CreditData, whis=np.inf,palette="pastel",hue='Sex')
plt.xlabel('Purpose for the credit',fontsize = 12)
plt.ylabel('Age',fontsize = 12)
plt.xticks(rotation=90,fontsize = 12)
plt.title('BoxPlot for the Purpose of the credit based on Age for each Gender',fontsize=16)
plt.show()
```



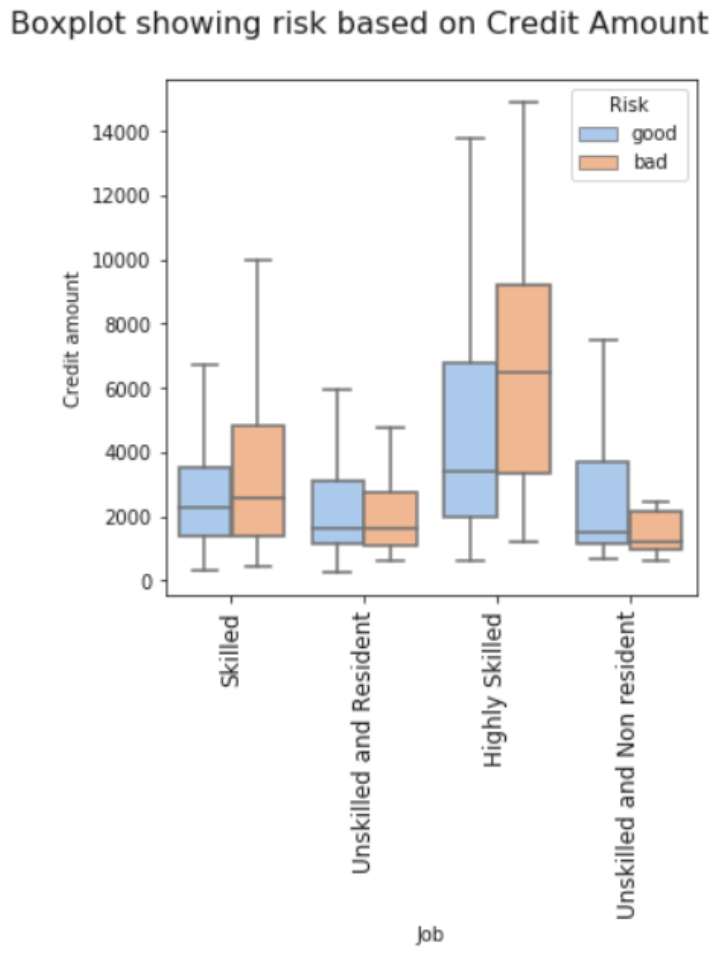
The boxplot shown below, draws a relationship among 'Job' and 'Age_Group' categories to that with 'Credit amount' numerical category. It can be observed that people working under 'highly skilled' category for all the age groups take the highest credit amount, and the highest among this category is the people aged 18-27. Whereas, the least credit amount among all the age group are those people who are working under 'Unskilled and Non-resident' category. Skilled people of all age groups falls next to the highly skilled when it comes to taking higher credit amount.

In [50]:
```
sns.boxplot(x='Job',y='Credit amount',hue='Age_Group',data=CreditData,palette="pastel",showfliers=False)
plt.xticks(rotation=90,fontsize = 12)
plt.show()
```



The below boxplot is plotted in accordance with the above boxplot. This boxplot below is important for the fact that even though it was inferred from the above boxplot that 'highly skilled' Job category people take the highest credit amount but the boxplot below indicates that these people have higher 'bad' credit risk. Whereas 'Unskilled and non-resident' takes lesser credit amount but also have lesser bad credit risk and more good credit risk.

```
In [51]: sns.boxplot(x='Job',y='Credit amount',hue='Risk',data=CreditData,palette="pastel",showfliers=False)
         plt.suptitle('Boxplot showing risk based on Credit Amount and Job',fontsize=16)
         plt.xticks(rotation=90,fontsize = 12)
         plt.show()
```



4. Summary

It can be concluded that for phase 1, the main part of data preprocessing is done by converting the Job variable into categorical feature, grouping the age variable and also by summarizing various descriptive and statistical features. Dataset did not contain any whitespaces and based on the assumption the 'Nan' values were not removed. From the visualizations, it can be inferred that among all the descriptive features, 'Age', 'Sex', 'Job', 'Housing', 'Credit amount' and 'Purpose' are the most important features to describe the target feature 'Risk'.