Edinburgh Napier
UNIVERSITY

DATA ANALYTICS CW

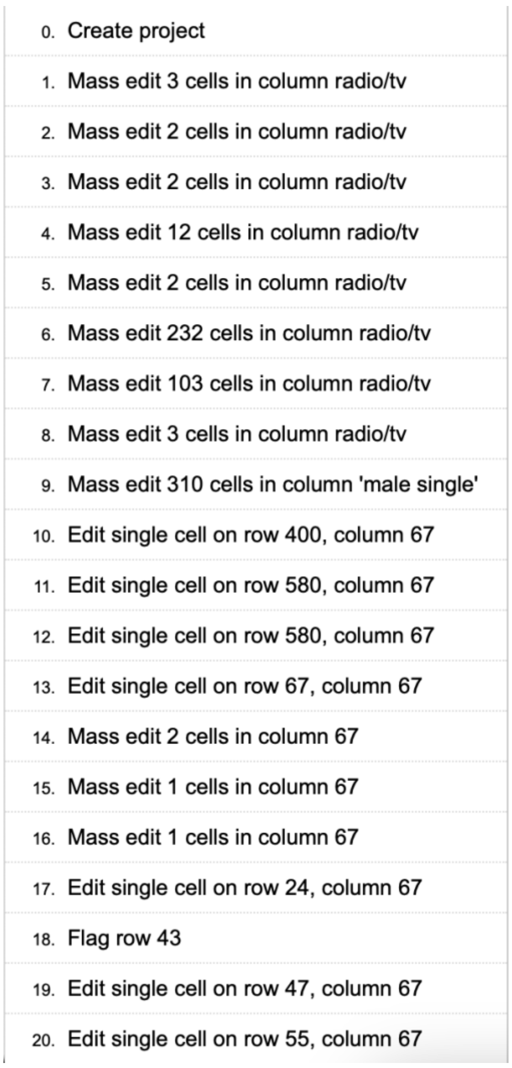
Report

Talalaiko, Kiril
40618094
@live.napier.ac.uk

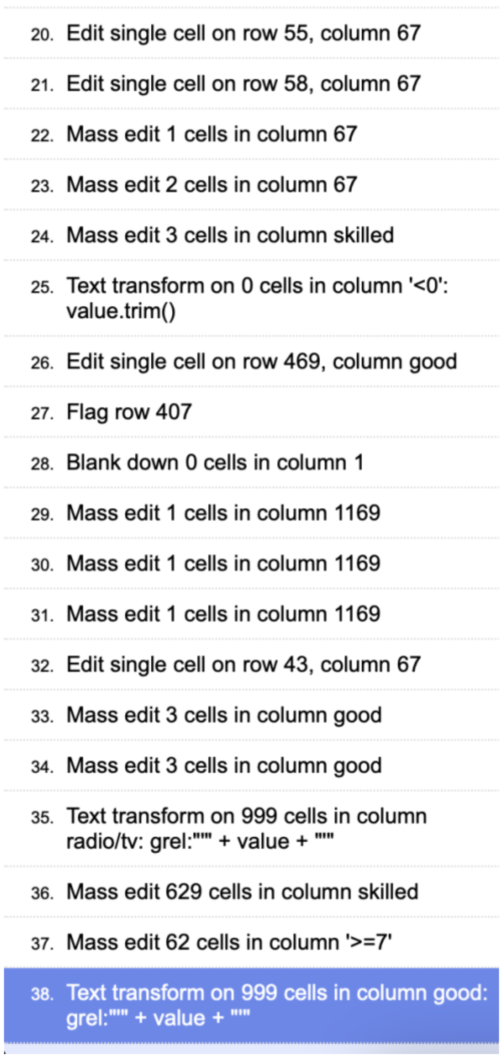
1. Data cleaning and transformation

I’ve used OpenRefine for data cleaning, loading the provided csv dataset. To perform the data cleaning I’ve used the checklist from datacamp.com.

- I’ve checked for missing values and I didn’t find a single missing value. To check that I’ve been using text facet.
- I’ve edited some typos in multiple columns especially in purpose and age, using the same text facet to identify them.
- I’ve mentioned the incorrect datatype in age and saving_amount columns and changed it to numeric.
- I’ve mentioned the inconsistent usage of quotation marks in purpose, job status and class columns so using OpenRefine functions I’ve changed all the values that missed them to “’” + value + “’” so they transformed to ‘value’.
- I’ve seen that the columns are not named and changed it to their names respectively for proper table system.
- To perform uniqueness check on case_no values I’ve used the numerical facet to be sure whether the numbers are unique.
- I’ve used clusters to find and perform some simplifying but if I’d performed it data accuracy would have reduced. Thereafter I found the age error on row 44 and 407 with age = 1 and flagged because I can’t change it since it might have too many different values starting from 1 or 10-19 or even 100.



0. Create project
1. Mass edit 3 cells in column radio/tv
2. Mass edit 2 cells in column radio/tv
3. Mass edit 2 cells in column radio/tv
4. Mass edit 12 cells in column radio/tv
5. Mass edit 2 cells in column radio/tv
6. Mass edit 232 cells in column radio/tv
7. Mass edit 103 cells in column radio/tv
8. Mass edit 3 cells in column radio/tv
9. Mass edit 310 cells in column 'male single'
10. Edit single cell on row 400, column 67
11. Edit single cell on row 580, column 67
12. Edit single cell on row 580, column 67
13. Edit single cell on row 67, column 67
14. Mass edit 2 cells in column 67
15. Mass edit 1 cells in column 67
16. Mass edit 1 cells in column 67
17. Edit single cell on row 24, column 67
18. Flag row 43
19. Edit single cell on row 47, column 67
20. Edit single cell on row 55, column 67



20. Edit single cell on row 55, column 67
21. Edit single cell on row 58, column 67
22. Mass edit 1 cells in column 67
23. Mass edit 2 cells in column 67
24. Mass edit 3 cells in column skilled
25. Text transform on 0 cells in column '<0': value.trim()
26. Edit single cell on row 469, column good
27. Flag row 407
28. Blank down 0 cells in column 1
29. Mass edit 1 cells in column 1169
30. Mass edit 1 cells in column 1169
31. Mass edit 1 cells in column 1169
32. Edit single cell on row 43, column 67
33. Mass edit 3 cells in column good
34. Mass edit 3 cells in column good
35. Text transform on 999 cells in column radio/tv: grel:"'" + value + "'"
36. Mass edit 629 cells in column skilled
37. Mass edit 62 cells in column '>=7'
38. Text transform on 999 cells in column good: grel:"'" + value + "'"

Issue	Fix	Description
radio/tv	‘radio/tv’	Standardization of the data.
eduction	‘education’	Correcting typo and Standardizing data.
businss	‘business’	Correcting typo and Standardizing data.
Furniture/equip & fur/eqi	‘furniture/equipment’	Correcting typos and Standardizing data.
Purpose, job, personal status, status columns	“’” + value + “’” + changing the constraint in .arff so possible values in a column only can be standard	Standardizing the data.
Personal status column ‘female div/dep/mar’	‘female div/sep/mar‘ since it stands for divorced, separated, married. + constraint change	Standardization of data and correction of typo.
Age column	Resolving floating numbers removing the comma to obtain the proper value but flagging them since the value might not be correct.	Standardization of data and correction of multiple typos.
Clustering the case_no to check for uniqueness	All the numbers are nominal and unique	Standardization of data and duplicates removal.
Columns names inexistence	Adding the column names (attributes)	Standardization and allowing to work with the data in python and WEKA properly.
Redundant ‘000’ in credit amount column	Cluster credit_amount to see the similarity and purpose and remove extra ‘000’	Correcting typos and unexpectable amount of credit

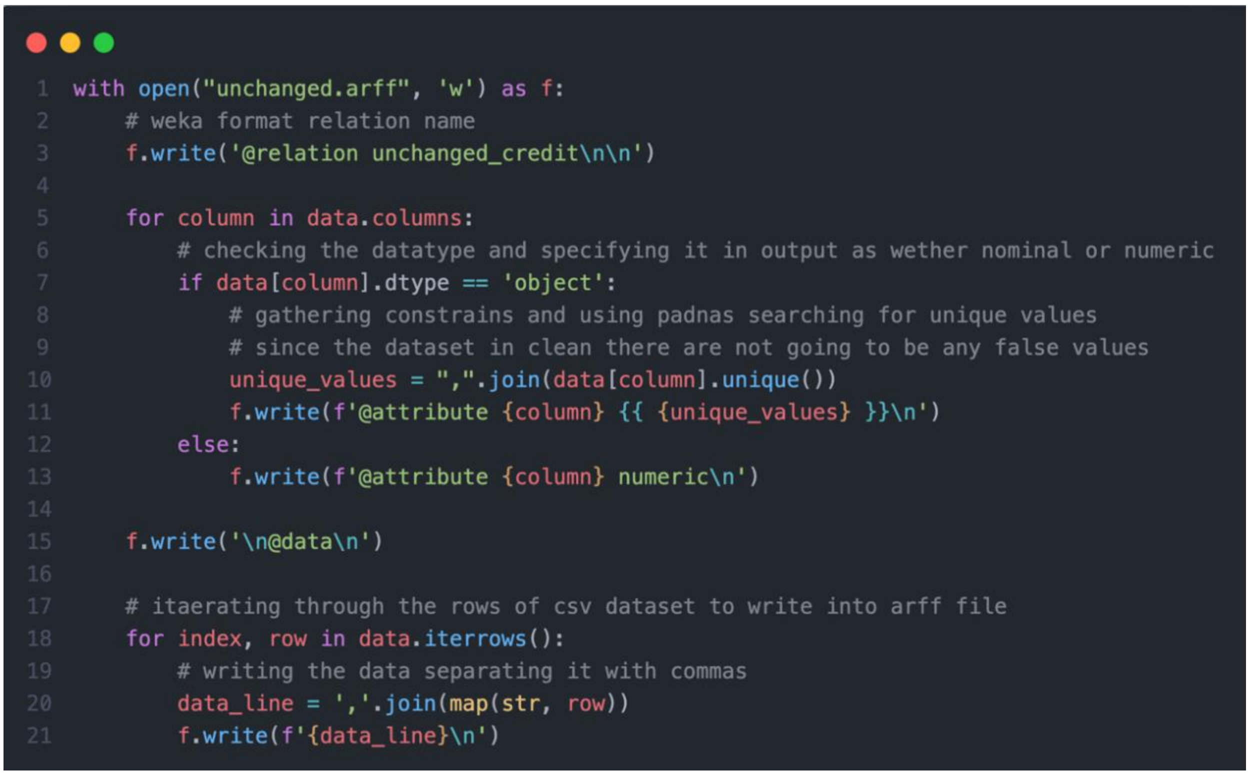
2. Data transformation and conversion

To transform data into required formats I've used the WEKA simply uploading the dataset in csv format that I acquired exporting from OpenRefine, exploring to see whether the data is displayed properly to understand the dependencies and check whether constrains are obeyed. Thereafter, I export the data in weka format (.arff) to use it in Python Data Analysis.

To perform data conversion using python regarding the technical request of unchanged, nominal and numeric, I've used a structure of exported .arff file to understand what has got to be in place and thereafter concluded to manually set the data constraints and decided with strategy of iterating throughout the file.

1. Unchanged

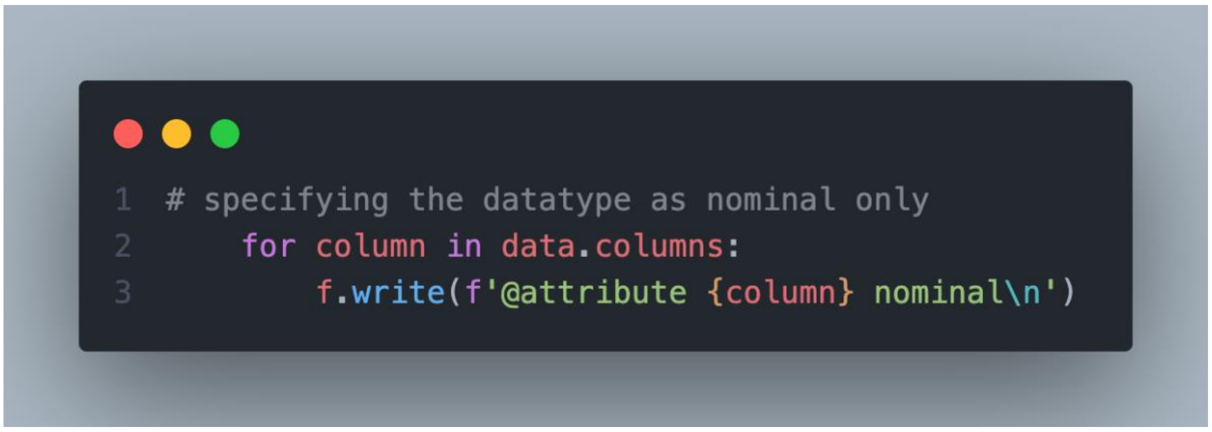
For unchanged I set the relation's name to unchanged credit manually and set the name of output file as unchanged.arff. Then I proceed iterating through the columns to gather data columns and obtain the unique values to proceed setting them as constraints. Finally I start to iterate through the rows to gather the dataset itself and separate it with commas since I'm using it as pandas dataset object and not csv and concatenate all the data into the file.



```python
with open("unchanged.arff", 'w') as f:
    # weka format relation name
    f.write('@relation unchanged_credit\n\n')

    for column in data.columns:
        # checking the datatype and specifying it in output as wether nominal or numeric
        if data[column].dtype == 'object':
            # gathering constrains and using padnas searching for unique values
            # since the dataset in clean there are not going to be any false values
            unique_values = ",".join(data[column].unique())
            f.write(f'@attribute {column} {{ {unique_values} }}\n')
        else:
            f.write(f'@attribute {column} numeric\n')

    f.write('\n@data\n')

    # itaerating through the rows of csv dataset to write into arff file
    for index, row in data.iterrows():
        # writing the data separating it with commas
        data_line = ','.join(map(str, row))
        f.write(f'{data_line}\n')
```

2. Nominal only

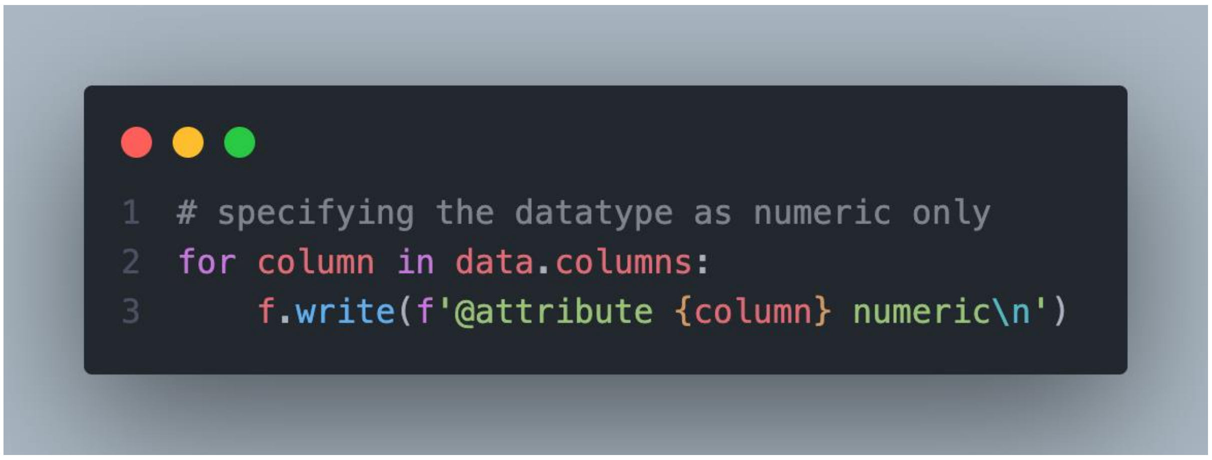
For nominal only it's pretty much the same, but I skip the step of checking and saving the datatype and just specify all the values as nominal.



```python
# specifying the datatype as nominal only
    for column in data.columns:
        f.write(f'@attribute {column} nominal\n')
```

3. Numeric only

Absolutely the same as nominal only, but instead of datatype of nominal I set it as numeric only.



```python
# specifying the datatype as numeric only
for column in data.columns:
    f.write(f'@attribute {column} numeric\n')
```

3. Data Framework and Visualization

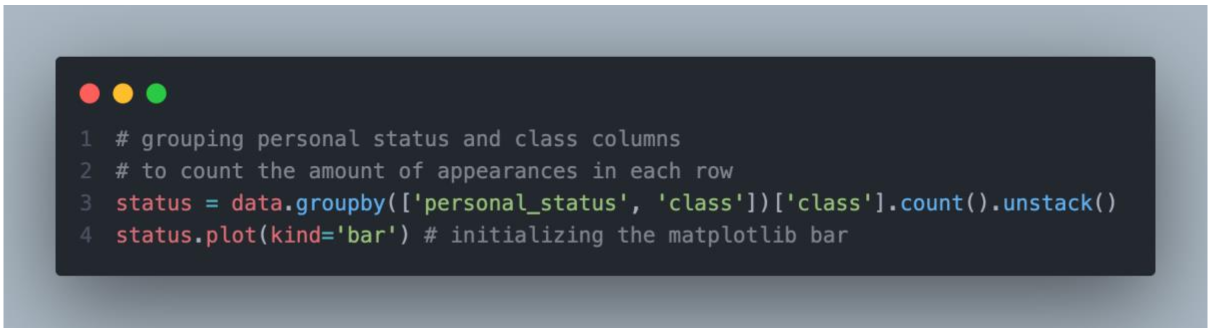
To perform all the data work and visualization I’ve used pandas and matplotlib libraries. Basically specifying the relation I want to visualize with pandas and rendering the diagram with matplotlib.

1. Relation of personal status to credit class

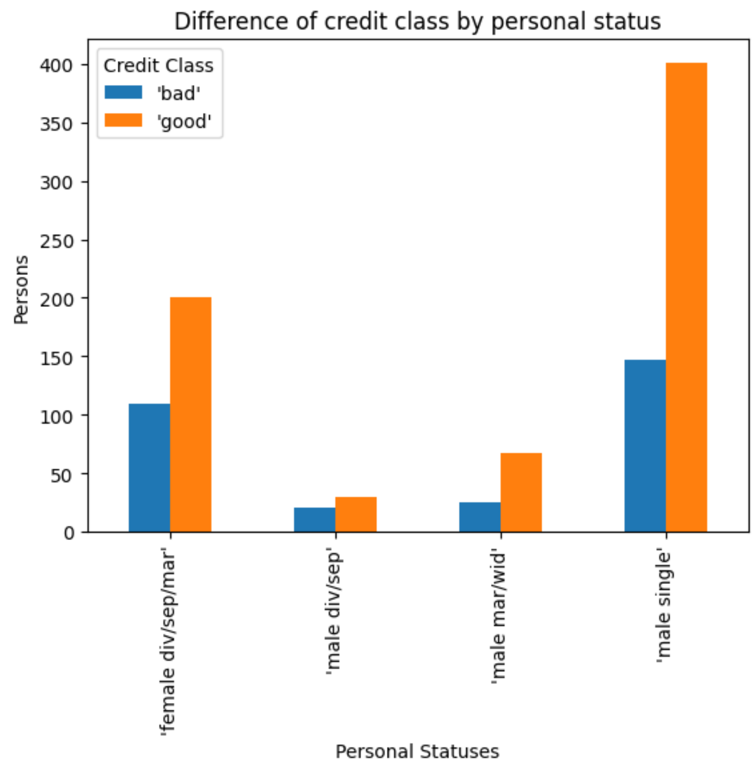
The code groups personal status and credit class columns to calculate the appearances of each other in dataset and saves in its status variable.

Thereafter, I’ve “unstacked” the data saved in status that means instead of saving the number of occurrences I do save it as separate dataset of only personal status and credit class columns.

Finally I initialize and render the bar-diagram to show the difference visually.

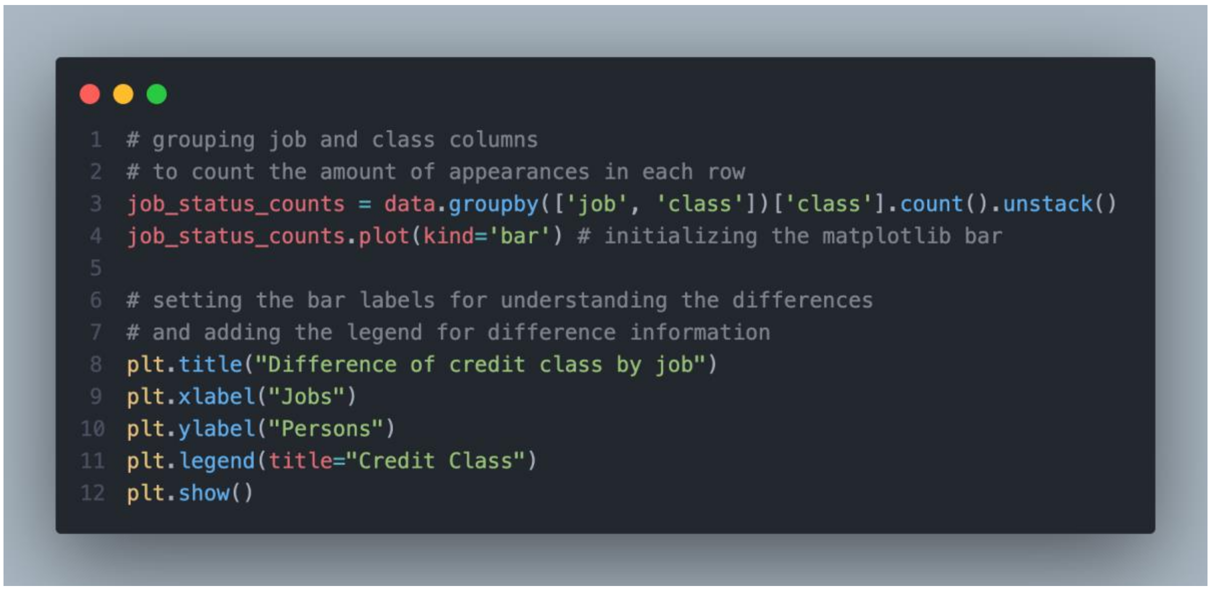


```
# grouping personal status and class columns
# to count the amount of appearances in each row
status = data.groupby(['personal_status', 'class'])['class'].count().unstack()
status.plot(kind='bar') # initializing the matplotlib bar
```

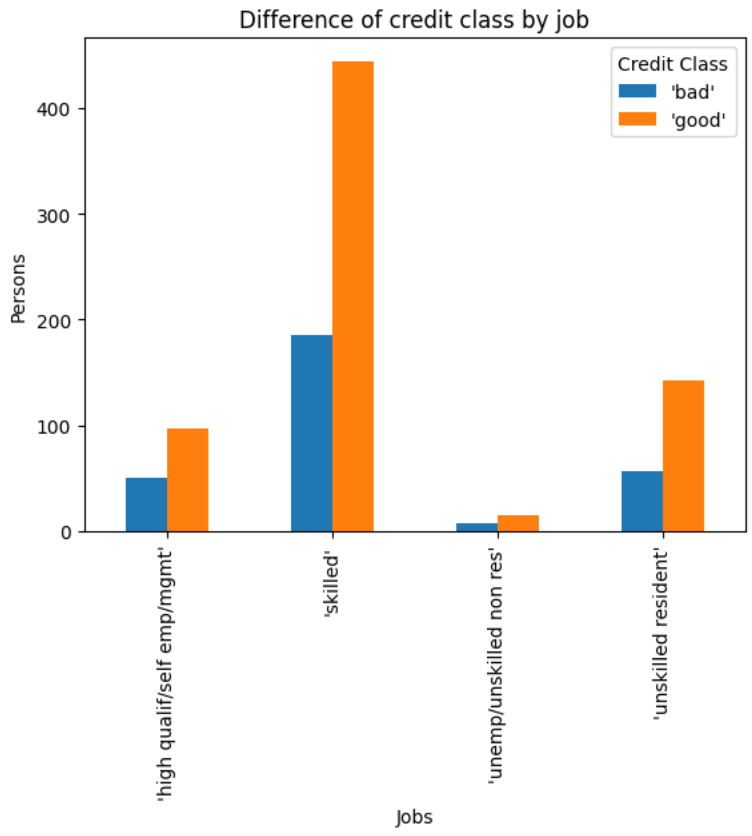


2. Relation of Job status to credit class

Code is almost unchanged in comparison to previous relation visualization, so the code proceeds to group two columns of job status and credit class and count it's appearances in each other into stack, thereafter the dataset is unstacked giving us access to new dataset relation of job status and credit class, so using matplotlib I've visualized the data and rendered the bar diagram.



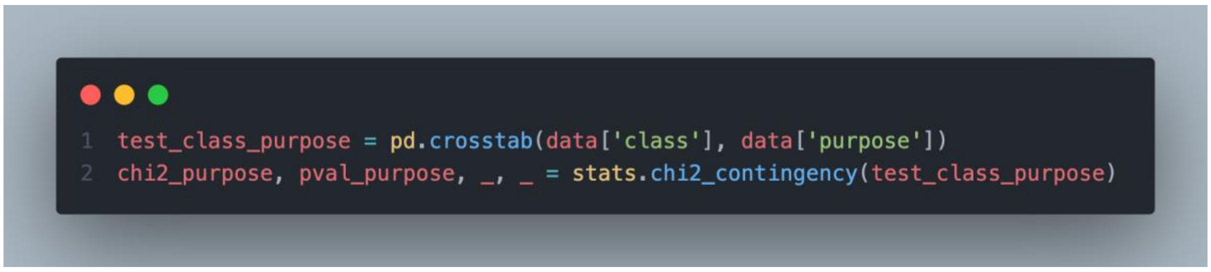
```python
# grouping job and class columns
# to count the amount of appearances in each row
job_status_counts = data.groupby(['job', 'class'])['class'].count().unstack()
job_status_counts.plot(kind='bar') # initializing the matplotlib bar

# setting the bar labels for understanding the differences
# and adding the legend for difference information
plt.title("Difference of credit class by job")
plt.xlabel("Jobs")
plt.ylabel("Persons")
plt.legend(title="Credit Class")
plt.show()
```



4. Scientific analysis

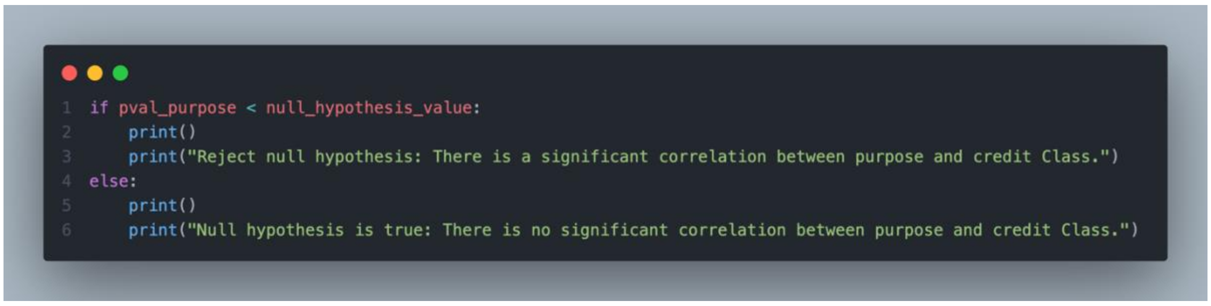
I’m checking the null hypothesis for the statements of Relation between: purpose and credit class, personal status and credit class, and job and credit class. Using the scipy and matplotlib libraries I’ve help chi tests obtaining the chi squared statistics and p-value that gives the basic correlation idea to understand whether our parameters of purpose, personal status and job influence the credit class.

Code illustrated with Purpose and Credit class relation



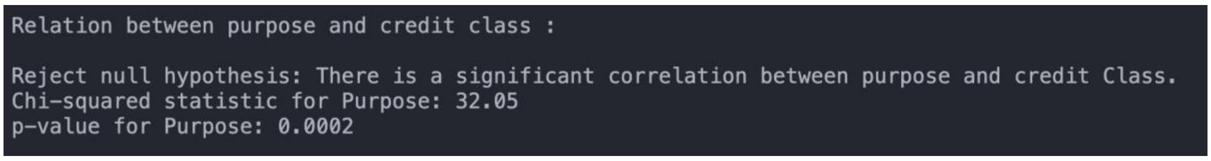
```python
test_class_purpose = pd.crosstab(data['class'], data['purpose'])
chi2_purpose, pval_purpose, _, _ = stats.chi2_contingency(test_class_purpose)
```

I’ve held the chi2 test using the cross tabulation function of pandas that basically returned me the frequency of good and bad options between all the purposes that gives me statistical information about how many good and bad options appearances are in each option of purpose.



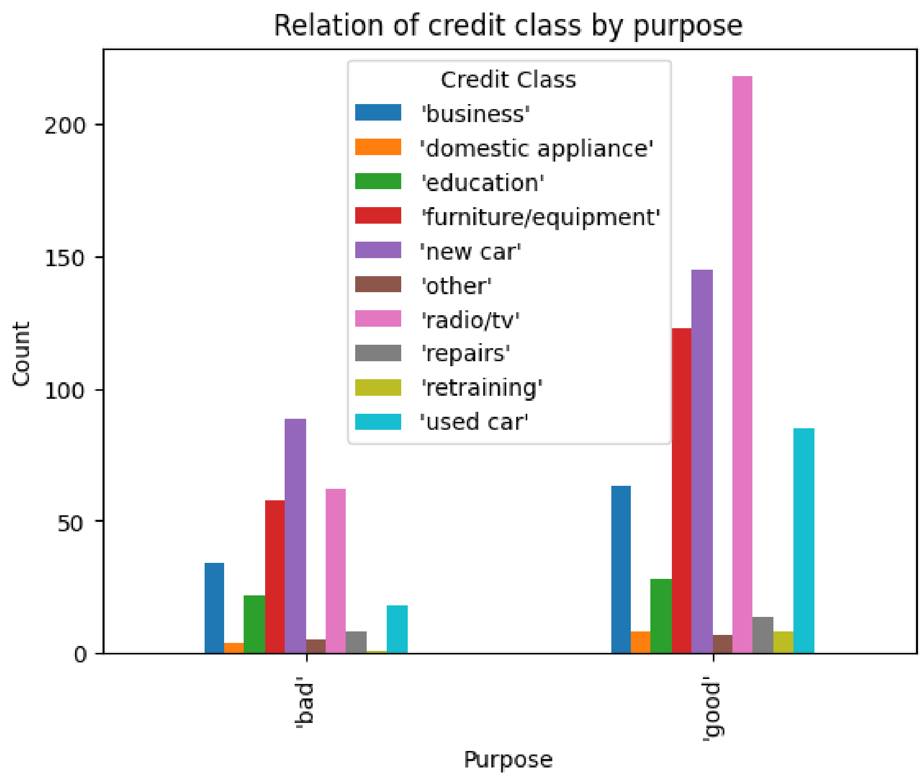
```python
if pval_purpose < null_hypothesis_value:
    print()
    print("Reject null hypothesis: There is a significant correlation between purpose and credit Class.")
else:
    print()
    print("Null hypothesis is true: There is no significant correlation between purpose and credit Class.")
```

Thereafter, using if statement I’ve checked whether null hypothesis is confirmed or not.



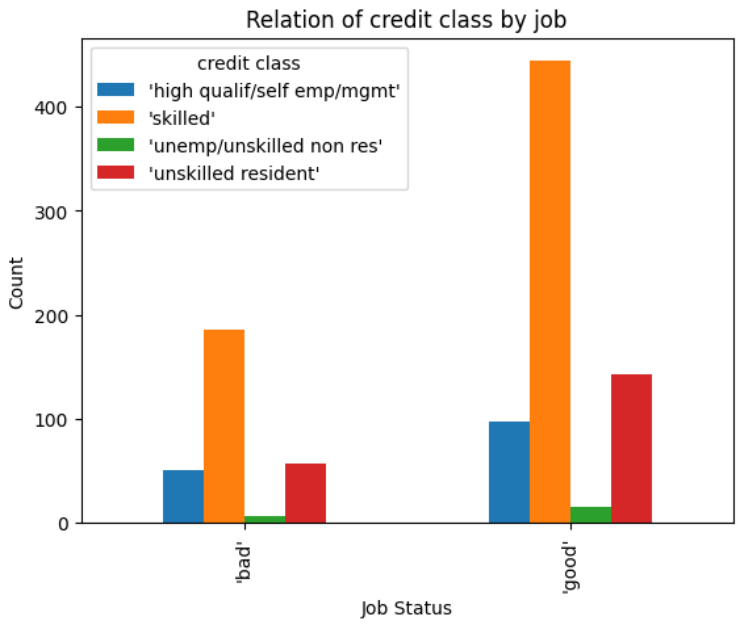
```
Relation between purpose and credit class :

Reject null hypothesis: There is a significant correlation between purpose and credit Class.
Chi-squared statistic for Purpose: 32.05
p-value for Purpose: 0.0002
```

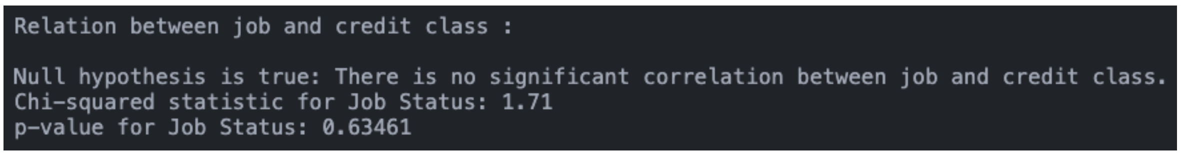
Next, independently of the answer I’ve printed the chi squared and p-value variables to show the numbers.



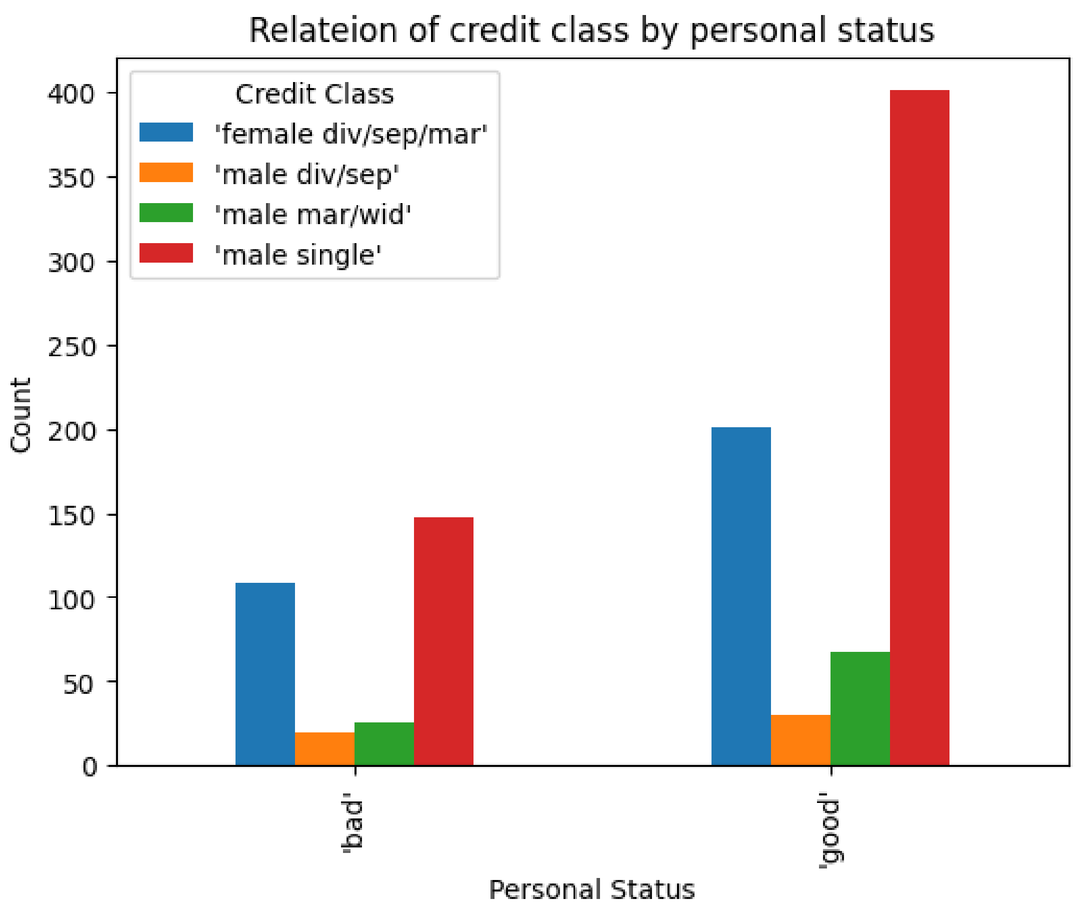
Finally, using the maplotlib, I've rendered a bar diagram to illustrate the difference visually, and reject null hypothesis since the values difference is significant.

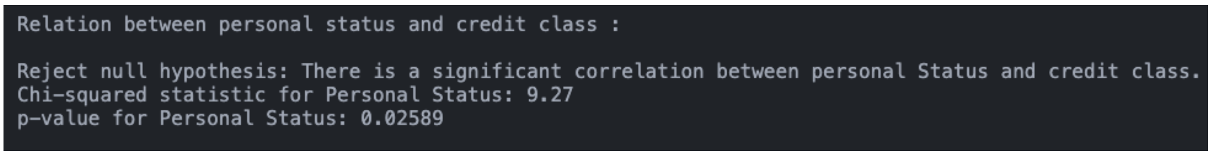
Results





```
Relation between job and credit class :

Null hypothesis is true: There is no significant correlation between job and credit class.
Chi-squared statistic for Job Status: 1.71
p-value for Job Status: 0.63461
```





```
Relation between personal status and credit class :

Reject null hypothesis: There is a significant correlation between personal Status and credit class.
Chi-squared statistic for Personal Status: 9.27
p-value for Personal Status: 0.02589
```