Part_II_slide_deck_template

November 6, 2022

1 Part II - (prosperLoanData)

1.1 by (Ahmed Abdelhady Saleh)

Before you start: You must have the README.md file ready that include a summary of main findings that reflects on the steps taken during the data exploration (Part I notebook). The README.md file should also describes the key insights that will be conveyed by the explanatory slide deck (Part II outcome)

1.2 Investigation Overview

In this presentation we will spread some perceptions about the characteristics of loans to see how they relate to the borrower's Annual Percentage Rate we will make visualizations on some important features such as BorrowerAPR, ProsperScore, and CreditScoreRangeUpper.

Rubric Tip: The key insights in the slideshow must match those documented in the README.md summary.

1.3 Dataset Overview

This data set contains 113,937 loans with 81 variables on each loan, including loan amount, borrower rate (or interest rate), current loan status, borrower income, and many others.

In [1]:
```
# import all packages and set plots to be embedded inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

%matplotlib inline

# suppress warnings from final output
import warnings
warnings.simplefilter("ignore")
```

In [2]:
```
# load in the dataset into a pandas dataframe
df=pd.read_csv('prosperLoanData.csv')
df.head()
```

Out[2]:

	ListingKey	ListingNumber	ListingCreationDate	CreditGrade	Term	LoanStatus	ClosedDate	BorrowerAPR	BorrowerRate	LenderYield	...	LP_ServiceFees	LP_CollectionFees	LP_GrossPrincipalLoss	LP_NetPrincipalLoss	LP_NonPrincipalRecoverypayments	PercentFunded	Recommendations	InvestmentFromFriendsCount	InvestmentFromFriendsAmount	Investors
0	1021339766868145413AB3B	193129	2007-08-26 19:09:29.263000000	C	36	Completed	2009-08-14 00:00:00	0.16516	0.1580	0.1380	...	-133.18	0.0	0.0	0.0	0.0	1.0	0	0	0.0	258
1	1027360249950330B223C1	1209647	2014-02-27 08:28:07.900000000	NaN	36	Current	NaN	0.12016	0.0920	0.0820	...	0.00	0.0	0.0	0.0	0.0	1.0	0	0	0.0	1
2	0EE9337825851032864889A	81716	2007-01-05 15:00:47.090000000	HR	36	Completed	2009-12-17 00:00:00	0.28269	0.2750	0.2400	...	-24.20	0.0	0.0	0.0	0.0	1.0	0	0	0.0	41
3	0EF5356002482715299901A	658116	2012-10-22 11:02:35.010000000	NaN	36	Current	NaN	0.12528	0.0974	0.0874	...	-108.01	0.0	0.0	0.0	0.0	1.0	0	0	0.0	158
4	0F023589499656230C5E3E2	909464	2013-09-14 18:38:39.097000000	NaN	36	Current	NaN	0.24614	0.2085	0.1985	...	-60.27	0.0	0.0	0.0	0.0	1.0	0	0	0.0	20

[5 rows x 81 columns]

In [3]:
```
df.drop(['ListingKey', 'ListingNumber', 'ListingCreationDate', 'CreditGrade', 'ClosedDat
```

```
df2 = df[df['ProsperScore'].isnull()==False]
```

Note that the above cells have been set as "Skip"-type slides. That means that when the notebook is rendered as http slides, those cells won't show up.

1.4 ((BorrowerAPR values Distribution))

Distribution is considered normal except in a specific period in which the distribution is interesting, which is the period when the value of BorrowerAPR between0.35797% and 0.35643%.

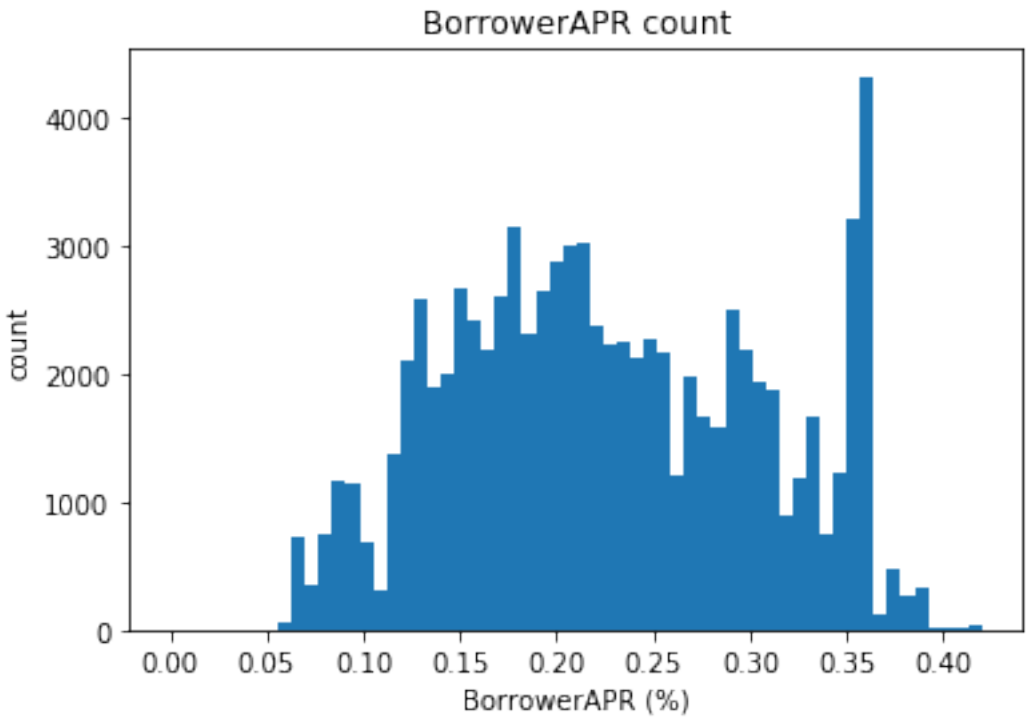
Rubric Tip: Provide at least 3 **polished** visualizations to convey key insights. The total number of visualizations in the slideshow shoould be less than 50% of the number of visualizations in the exploratory analysis. For example, if the exploratory analysis (Part I) has 18 visualizations, the slideshow can have (3 - 8) visualizations.

Rubric Tip: Each visualization in the slideshow is associated with **descriptive comments** that accurately depict their purpose and your observation.

Rubric Tip: All plots in the slideshow are appropriate, meaning the plot type, encodings, and transformations are suitable to the underlying data.

Rubric Tip: All plots in the slideshow are polished, meaning all plots have a title, labeled x/y axes (with units), x/y ticks, and legends.

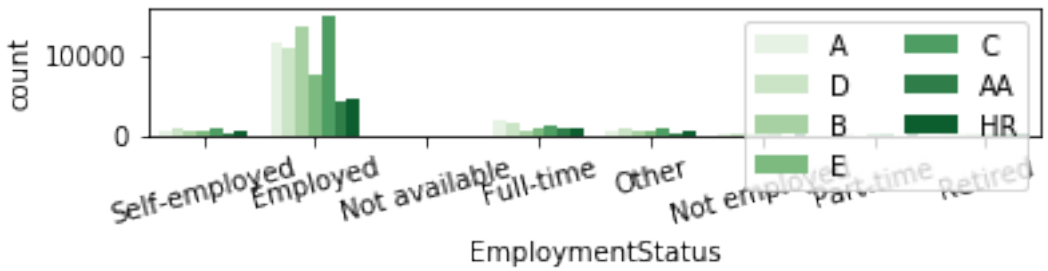
In [4]:
```
bins = np.arange(0, df2['BorrowerAPR'].max(), 0.007)
plt.hist(data = df2, x = 'BorrowerAPR', bins = bins)
plt.title('BorrowerAPR count')
plt.xlabel('BorrowerAPR (%)')
plt.ylabel('count')
plt.xticks(np.arange(0, df2['BorrowerAPR'].max(), 0.05));
```



1.5 ((Prosper rating vs. employment status))

this visual show that how employed people have a strong prosper rating compared to other employment status

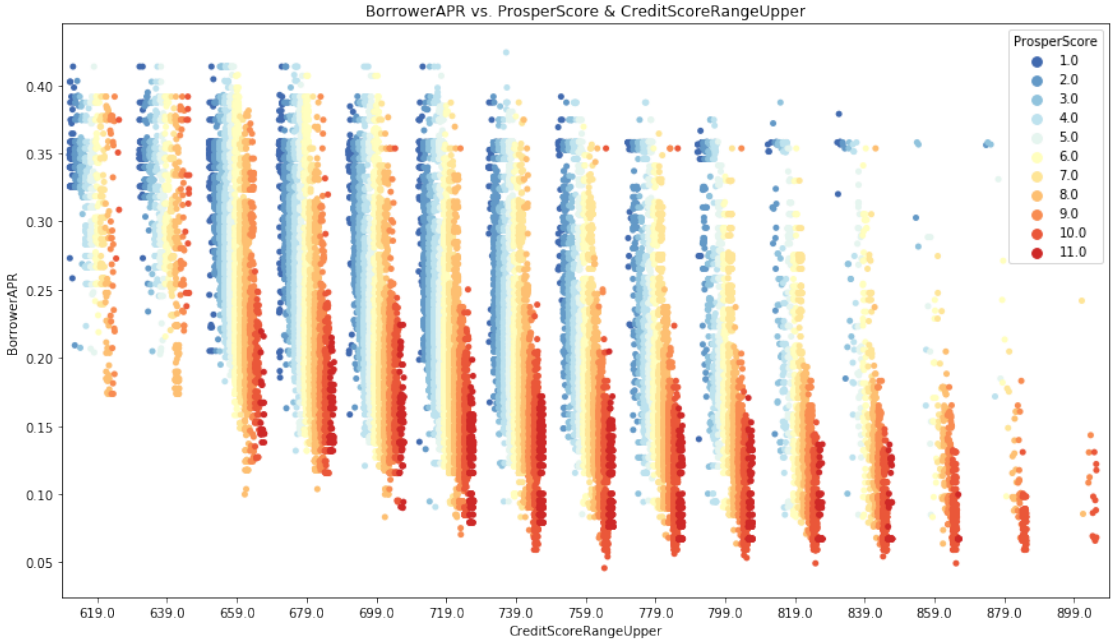
```
In [5]: # Prosper rating vs. employment status, use different color palette
        ax = plt.subplot(4, 1, 3)
        sb.countplot(data = df, x = 'EmploymentStatus', hue = 'ProsperRating (Alpha)', palette =
        ax.legend(loc = 1, ncol = 2);
        plt.xticks(rotation = 15);
```



1.6 ((BorrowerAPR & ProsperScore & CreditScoreRangeUpper))

We can see the creditScoreRangeUpper rise with the borrowerAPR fall in parcels. By adding ProsperScore to the colour encodings, the BorrowerAPR decreases as the ProsperScore rise. This proves that CreditScoreRangeUpper and ProsperScore are adversely correlated with BorrowerAPR.

In [6]:
```
plt.figure(figsize = [14.7, 8.27])
sb.stripplot(data = df2, x = 'CreditScoreRangeUpper', y = 'BorrowerAPR', hue ='ProsperSc
plt.title('BorrowerAPR vs. ProsperScore & CreditScoreRangeUpper')
plt.xlabel('CreditScoreRangeUpper')
plt.ylabel('BorrowerAPR');
```



1.6.1 Generate Slideshow

Once you're ready to generate your slideshow, use the `jupyter nbconvert` command to generate the HTML slide show.

In [ ]:
```
# Use this command if you are running this file in local
!jupyter nbconvert Part_II_slide_deck_template.ipynb --to slides --post serve  --no-inpu
```

```
[NbConvertApp] Converting notebook Part_II_slide_deck_template.ipynb to slides
[NbConvertApp] Writing 597366 bytes to Part_II_slide_deck_template.slides.html
[NbConvertApp] Redirecting reveal.js requests to https://cdnjs.cloudflare.com/ajax/libs/reveal.j
Serving your slides at http://127.0.0.1:8000/Part_II_slide_deck_template.slides.html
Use Control-C to stop this server
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: x-www-browser: not found
```

```
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: firefox: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: iceweasel: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: seamonkey: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: mozilla: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: epiphany: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: konqueror: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: chromium-browser: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: google-chrome: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: www-browser: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: links2: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: elinks: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: links: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: lynx: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: w3m: not found
xdg-open: no method available for opening 'http://127.0.0.1:8000/Part_II_slide_deck_template.sli
```

```
In [ ]:
```

In the classroom workspace, the generated HTML slideshow will be placed in the home folder.

In local machines, the command above should open a tab in your web browser where you can scroll through your presentation. Sub-slides can be accessed by pressing 'down' when viewing its parent slide. Make sure you remove all of the quote-formatted guide notes like this one before you finish your presentation! At last, you can stop the Kernel.

1.6.2 Submission

If you are using classroom workspace, you can choose from the following two ways of submission:

1. **Submit from the workspace**. Make sure you have removed the example project from the /home/workspace directory. You must submit the following files:

- Part_I_notebook.ipynb
- Part_I_notebook.html or pdf
- Part_II_notebook.ipynb
- Part_I_slides.html
- README.md
- dataset (optional)

2. **Submit a zip file on the last page of this project lesson**. In this case, open the Jupyter terminal and run the command below to generate a ZIP file.

   ```
   zip -r my_project.zip .
   ```

   The command abobve will ZIP every file present in your /home/workspace directory. Next, you can download the zip to your local, and follow the instructions on the last page of this project lesson.