Determining if Jupyter Notebook is the Right Tool for Students Based on Industry Recognition

Alec Panattoni and Tyler Tran

0. Contribution Statement

Alec Panattoni and Tyler Tran contributed to the Python and R code necessary for analysis in conducting simulation testing and formal statistical tests. Alec Panattoni focused on code for questions one and three, providing both initial analysis and some visualizations in Tableau (for question three). Tyler Tran completed the remaining code for questions one and two, and was in charge of general formatting. Other questions and sections had programming equally distributed among both contributors. Overall, both students played a role in writing plus debugging any code used and in making changes to the report structure as necessary. Any report features not specifically mentioned were contributed equally to by both students.

1. Introduction

Jupyter Notebook is a widely popular platform for learning about and executing Python code in the classroom. Heralded as a "welcome addition to the learning delivery formats available because of its qualities that allow exploration", Jupyter notebook is a tool very familiar to students learning Python and data science, including those here at UCSD (Kluyver 1). Therefore, it is appropriate to ask: is Jupyter Notebook the right tool for universities to use in teaching Python and data science more specifically? To answer this overarching question, we turn to Kaggle's annual Machine Learning and Data Science Survey from 2020. As a comprehensive survey of the sentiment of industry professionals and students alike, this data provides important insights into the popularity and usability of Jupyter Notebook as a data science tool. To assess Jupyter's popularity among data scientists and students, a point and interval estimation will be conducted and verified using a simulation study. Additionally, given our anecdotal knowledge of the popularity of Jupyter in universities in the US, comparing its domestic popularity with its international popularity (both overall and among students) would help in forming a picture of its popularity overall. Furthermore, the popularity of Jupyter Notebook across experience levels will be analyzed As data science is a wide domain with many specializations and needs, it is also important to compare the recommendations of users based on their survey responses about their ML experiences. Graphical analysis and cross-tabulations can be used to display preference for IDE based on having or lacking machine learning experience. Finally by fitting and refining a logistic model, can we predict whether or not they use a Jupyter notebook based on their years of experience programming? Similar advanced analysis can be carried out using Naive Bayesian classification to see how likely it is for a respondent to use Jupyter tools based on metrics like their educational level, occupation title, and years of experience coding .

Data

The data, which was pulled directly from a survey conducted in the span of 3.5 weeks by Kaggle, provides a wide overview of the data science industry. With over 20,000 responses, the survey includes respondents new to the world of data science, all the way up to those that have worked in the field for many years. Thirty-nine questions were asked in the survey, leaving a single large dataset to be analyzed. Questions are centered around experience, programming preferences, age, education, and much more. This analysis is focused on a very popular IDE that is very familiar to us as UCSD students- JupyterLab. Therefore, we have pulled questions that allow us to look further assess the popularity of Jupyter among students and practicing data scientists alike. More specifically, our data is centered around who these people are, with the goal of determining how their experiences impact their recommended data science tool.

2. Analysis

2.0 Data Processing

Being that the given dataset consists of 355 columns, with many columns being a different answer to a certain question, this dataset was in desperate need of wrangling.Wrangling of the data began with noting the questions that were separated into different columns for every different answer as integers in a list.

```
survey = pd.read_csv("survey.csv", dtype = 'unicode')
morethanoneresponse = [7, 9, 10, 12, 14, 16, 17, 18, 19, 23, 26, 27, 28, 29, 31, 33, 34, 35, 36, 37, 39]
```

Next, these integers were converted into a dictionary of strings ("Q" + int(q)), which would later become each column name. The remaining clean question responses were temporarily placed into a Dataframe which would later become the wrangled survey.

	Q1	Q2	Q3	Q4	Q5	Q6	Q8	Q11	Q13	Q15	Q20	Q21	Q22	Q24	
0	What is your age (# years)?	What is your gender? - Selected Choice	In which country do you currently reside?	What is the highest level of formal education ...	Select the title most similar to your current ...	For how many years have you been writing code ...	What programming language would you recommend ...	What type of computing platform do you use mos...	Approximately how many times have you used a T...	For how many years have you used machine learn...	What is the size of the company where you are ...	Approximately how many individuals are respons...	Does your current employer incorporate machine...	What is your current yearly compensation (appr...	m
1	35-39	Man	Colombia	Doctoral degree	Student	5-10 years	Python	A cloud computing platform (AWS, Azure, GCP, h...	2-5 times	1-2 years	NaN	NaN	NaN	NaN	

The difficult portion of the wrangling was placing each response to the questions separated into different columns into a single list for each person that responded to the survey. This would create a list of lists of lists, where each list of lists was a question, and each list within these questions contained sets of responses from the respondents. Once this was done, the lists were stripped to create sets of strings for each observation and nans in the existing df were set to none.

```
[['What programming languages do you use on a regular basis? (Select all that apply)',
  '\'"\\\'Python\\\', \\\'R\\\', \\\'SQL\\\', \\\'C\\\', \\\'Javascript\\\', \\\'MATLAB\\\',
  '\'"\\\'Python\\\', \\\'R\\\', \\\'SQL\\\'"\'',
  '\'"\\\'Java\\\', \\\'Javascript\\\', \\\'Bash\\\'"\'',
```

Each list was then added to the existing DataFrame, each with its corresponding column title.

	Time from Start to Finish (seconds)	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13
0	Duration (in seconds)	What is your age (# years)?	What is your gender? - Selected Choice	In which country do you currently reside?	What is the highest level of formal education ...	Select the title most similar to your current ...	For how many years have you been writing code ...	What programming languages do you use on a reg...	What programming language would you recommend ...	Which of the following integrated development ...	Which of the following hosted notebook product...	What type of computing platform do you use mos...	Which types of specialized hardware do you use...	Approximately how many times have you used a T...
1	1838	35-39	Man	Colombia	Doctoral degree	Student	5-10 years	"\'Python\', \'R\', \'SQL\', \'C\', \'Javascr...	Python	"\'Jupyter (JupyterLab, Jupyter Notebooks, et...	"\'Kaggle Notebooks\', \'Colab Notebooks\'"	A cloud computing platform (AWS, Azure, GCP, h...	"\'GPUs\'"	2-5 times

2.1 Estimating the Popularity of Jupyter By Comparing Students to Industry Professionals

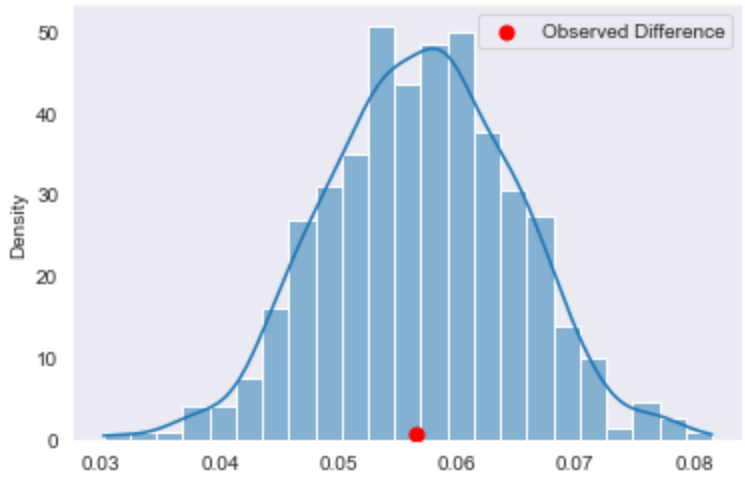
Methods

The data was initially loaded with Python in Jupyter. Data wrangling to aggregate responses where survey participants were able to select more than one response was performed before initial analysis. Then to assess the popularity of Jupyter products, a point and interval estimate was conducted for its usage among students and employed professionals. Any responses marked "currently not employed" were omitted from the analysis, as this could indicate either a non-employed professional or self-taught non-university student with no way of differentiating. Interval estimates created (see the formula below) were of 90% confidence to maintain a high likelihood of capturing the parameter of interest with acceptable levels of interval specificity. After creating interval and point estimates, the difference between the two point estimates was calculated to see any differences in popularity between the two groups. Bootstrapping was performed to assess the accuracy of the estimated difference.

Analysis

Proportion of Users Using Jupyter	0.56
Proportion of Students Using Jupyter	0.6
Proportion of Pros Using Jupyter	0.54
Observed Difference	0.06
P-Value	0.52

Interval Estimate (90% Confidence)	Low	High
Proportion of Users Using Jupyter	0.55	0.57
Proportion of Students Using Jupyter	0.59	0.61
Proportion of Pros Using Jupyter	0.53	0.55



Conclusion

Given that Jupyter is a tool built for Python, a popular programming language for data science, it's not too surprising that a large number of survey respondents actively use Jupyter Notebook and other Jupyter-based tools. This finding holds true for students, with an estimated 59 to 61% of students likely using Jupyter as their main tool. Considering its popularity in university classrooms, such a high percentage is within the realm of expectation. More interestingly, Jupyter's popularity persists for non-student respondents: a slightly lower 53 to 55% of working industry professionals are estimated to use Jupyter. Thus, the drop off in popularity is estimated at about 6%. To assess the accuracy of the observed difference, bootstrapping was conducted to find the range of variance estimation. With a p-value of 0.52, the observed difference is noted to be well-within the range of possible differences. As bootstrapping was conducted primarily to calculate the potential range of values for the difference between proportion of students and employed professionals using Jupyter, the range of values is of greater importance. Through the bootstrap, we found that the range of differences was between 3 and 8%, indicating a slight drop off in popularity of Jupyter among professionals, although Jupyter is still popular overall.

2.2 Considering Jupyter's Popularity Both Domestically and Abroad

Methods

Here in the United States, Jupyter is a popular platform used by universities to introduce students to data science and Python in general. Introductory courses here at UCSD and other acclaimed institutions often use Jupyter as their development environment of choice, potentially boosting the popularity of Jupyter domestically. Given the widespread use of Jupyter found in early analysis, we would expect Jupyter's popularity to hold both domestically and abroad. We can similarly expect that universities abroad hold Jupyter in a similar regard. Therefore we can test to see if the proportion of respondents in the US using Jupyter is different from international respondents using Jupyter via binomial tests. Similarly, we can adopt the same set of hypotheses to see if Jupyter is more popular specifically among students domestically or abroad.

Null Hypothesis: The proportion of US respondents who use Jupyter is not different from the proportion of abroad respondents who use Jupyter.

Alternative Hypothesis: The proportion of US respondents who use Jupyter is different from the proportion of abroad respondents who use Jupyter.

Null Hypothesis: The proportion of US students who use Jupyter is not different from the proportion of abroad students who use Jupyter.

Alternative Hypothesis: The proportion of US students who use Jupyter is different from the proportion of abroad students who use Jupyter.

Analysis

Statistic	Value
Proportion of US Respondents Using Jupyter	0.55
Proportion of Abroad Respondents Using Jupyter	0.56
P-Value for Respondents Binomal Test	0.24
Proportion of US Students Using Jupyter	0.55
Proportion of Abroad Students Using Jupyter	0.6
P-Value for Students Binomal Test	0.057

Discussion

Initial calculations of the proportions of Jupyter usage both domestically and abroad yielded interesting results: for both students and respondents overall, Jupyter seemed slightly more popular internationally. After running binomial tests to compare the distributions, in both cases we are unable to reject the null hypothesis. Specifically, in both cases there is insufficient evidence to say that the proportion of US students and respondents (respectively) using Jupyter tools are statistically different from the proportion of international students and respondents (respectively) using Jupyter. This indicates that Jupyter likely has worldwide popularity, among students and Kaggle users alike. From this we can conclude that if further analysis shows that Jupyter is justified as a preferred tool for data science, this recommendation would hold internationally as it seems to have garnered recognition across the globe.

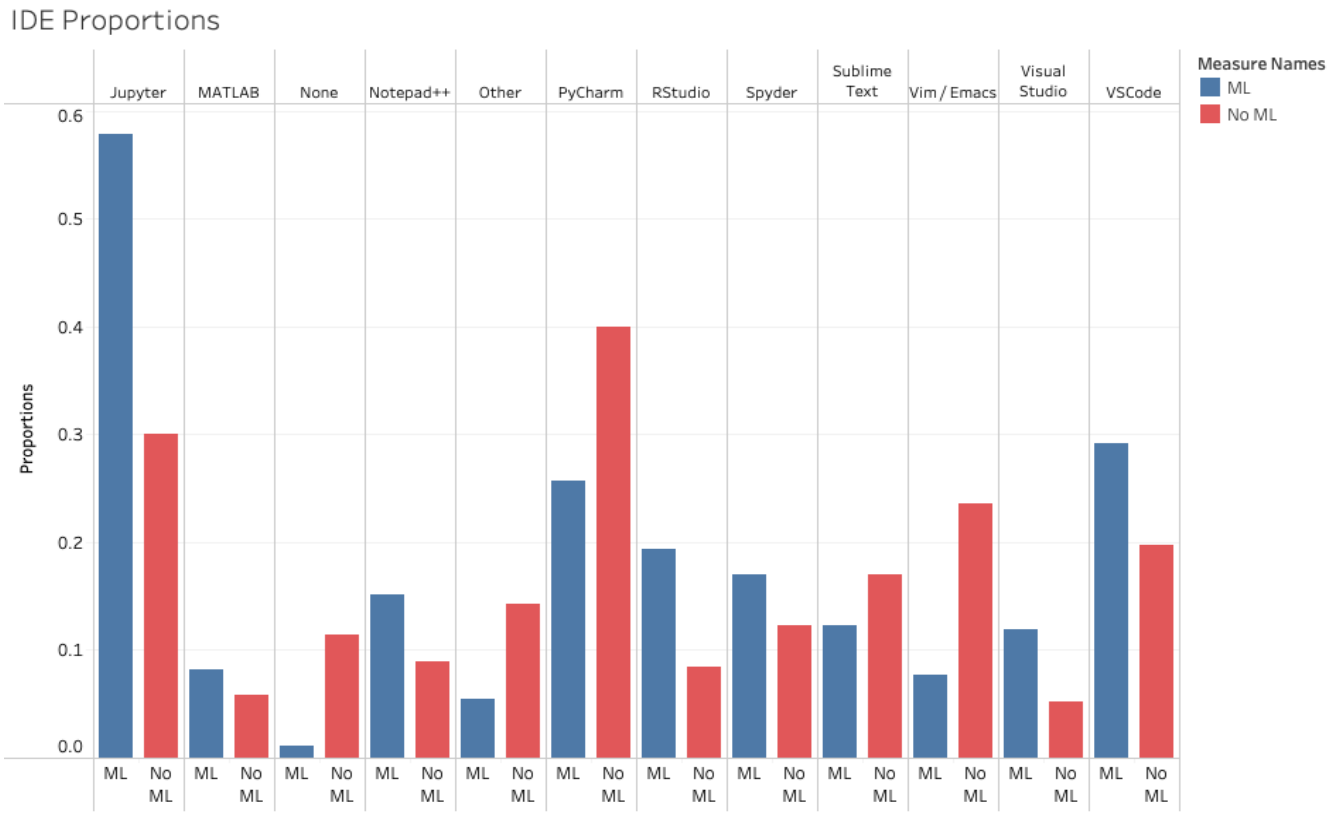
2.3 How Does Machine Learning Experience Impact Preference for Recommended Development Environment

Methods

Another important aspect of data science is the field of machine learning. In continuing assessment of Jupyter's popularity, we now consider survey responses to ML related questions about a user's experience with various topics in machine learning. Through graphical displays and cross tabulation, we can examine how familiarity with some machine learning impacts a respondents preference for IDE. In order to separate those with and without ML experience, the data will be subsetted into two separate groups. One of these groups will include only users with zero ML experience, while the other will include those with at least less than a year of ML experience. Next, the proportions of use for each IDE will be calculated within each group. Their differences will be plotted and analyzed through graphical analysis. Finally, a chi-square test for independence will be used to determine whether or not ML experience and preferred IDE are independent. This will help us determine whether the experience in ML impacts the choice of IDEs.

Analysis

IDE	MLexperience	noMLexperience	difference
Jupyter (JupyterLab, Jupyter Notebooks, etc)	0.578030	0.300723	0.277307
Visual Studio Code (VSCode)	0.292244	0.197108	0.095136
PyCharm	0.256612	0.399518	-0.142907
RStudio	0.193363	0.084819	0.108544
Spyder	0.170035	0.122892	0.047144
Notepad++	0.151606	0.089639	0.061968
Sublime Text	0.122321	0.170120	-0.047800
Visual Studio	0.119704	0.052530	0.067174
MATLAB	0.082512	0.058795	0.023717
Vim / Emacs	0.077557	0.236145	-0.158588
Other	0.054340	0.142169	-0.087829
None	0.011692	0.113735	-0.102043



Null Hypothesis:
Whether the user has ML experience or not and preferred IDE are independent

Alternative Hypothesis:
Whether the user has ML experience or not and preferred IDE are not independent

IDE	MLobs	MLexp	noMLobs	noMLexp
Jupyter (JupyterLab, Jupyter Notebooks, etc)	10382	10050	624	325
Visual Studio	2150	5265	109	156
PyCharm	4609	4571	829	1162
RStudio	3473	3430	176	40
Spyder	3054	2950	255	254
Visual Studio Code (VSCode)	5249	2808	409	325
Notepad++	2723	2808	186	121
Sublime Text	2197	2199	353	397
MATLAB	1482	1438	122	167
Vim / Emacs	1393	1347	490	529
Other	976	1042	295	609
None	210	347	236	341

df	11
chi-sq value	2486.17
p-value	0.00

With a p-value of approximately 0, we find insufficient support for the null hypothesis. Therefore, it is extremely likely that ML experience or not and preferred IDE are not independent.

Discussion

It becomes apparent that there are some very large differences in user proportions between those with ML experience and those with no ML experience. We see the largest differences in these proportions with IDEs: Jupyter, Vim/emacs, and Pycharm. Most surprisingly, Jupyter, with the largest difference in proportions of IDE preference, is almost 30% more popular among those with ML experience. Similar to the previous analysis pertaining to geographical location, this is the opposite of what we expected, considering Jupyer's beginner-friendly functionality. It was also surprising to see Vim/emacs much more popular amongst those not familiar with ML, being that it is an old-school IDE. To take things a step further, a chi-square test of independence was conducted to test whether ML experience impacts preferred IDE. With a p-value of approximately 0, the null hypothesis was rejected. Therefore, it cannot be assumed that there is no relationship between whether a respondent has ML experience and their preferred IDE. With this, it seems whether one has ML experience could influence preferred IDE. As this analysis pertains to Jupyter, shown user preference for Jupyter when the user has machine learning experience shows that Jupyter is a trusted tool for handling complex work such as machine learning, which students may be interested in pursuing throughout their data science careers.

2.4 Predicting Salary of Industry Professionals Recommending Jupyter Notebook Based On Their Years of Experience

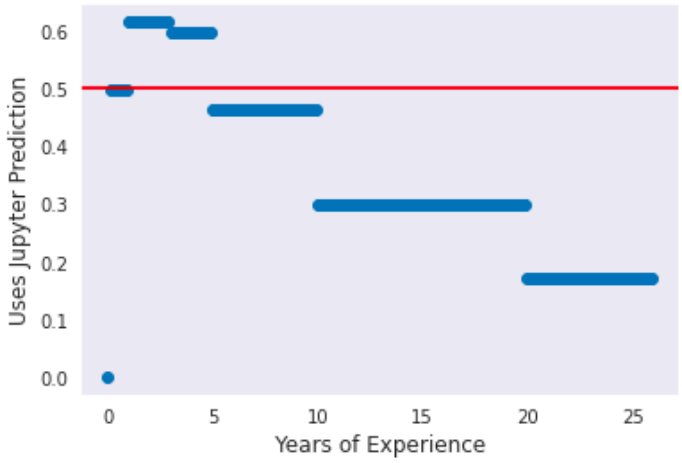
Methods

Students new to data science often highly regard advice of experienced programmers when selecting the right IDE to guide them, making it important to consider years of experience programming when analyzing the popularity of Jupyter. The data for years of experience programming provided by Kaggle is divided into subgroups of years, such as "< 1 year", "3-5 years", and "10-20 years" (including a subgroup for those with no coding experience). Thus, for programmers of varying experience levels, we will attempt to predict how likely they are to favor Jupyter as a function of their years of experience programming through a log model, given that the response variable is not continuous.

Analysis

```
Coefficients:
                                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                                0.49845    0.03583  13.911  < 2e-16 ***
Years.Experience1-2 years                  0.11742    0.04753   2.471 0.013486 *
Years.Experience10-20 years               -0.20043    0.06016  -3.332 0.000864 ***
Years.Experience20+ years                 -0.32798    0.06569  -4.993 5.96e-07 ***
Years.Experience3-5 years                  0.09852    0.04738   2.080 0.037571 *
Years.Experience5-10 years                -0.03417    0.05420  -0.631 0.528353
Years.ExperienceI have never written code -18.06452  118.00297  -0.153 0.878331
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Years Experience	Prefers Jupyter Prediction
I have never written code	0.00000
20+ years	0.17047
10-20 years	0.29802
5-10 years	0.46428
< 1 years	0.49845
3-5 years	0.59697
1-2 years	0.61587



Discussion

Our predictions displayed through our logistic model show that for those with years of experience programming between 1 and 5 years, Jupyter is generally a favored IDE. We can loosely categorize Jupyter as favorable for these year intervals, since their predictions are well over 0.5. This makes sense, since this experience level makes up mostly students and those at the

beginnings of their careers. Jupyter is favored likely due to its convenience and to the fact that there is little incentive to explore alternative IDEs. However, for those with over 5 years of experience programming, we see a large decrease for Jupyter as a favored IDE. This is justified, as with more experience, industry professionals are exposed to more complex IDEs that better suit their capabilities. What's interesting is that the survey respondents with under one year of experience coding hold a prediction right around 0.5. This could be for a multitude of reasons, such as: this person is learning introductory programming through a university/online course that uses a specific IDE, or perhaps they have not yet discovered Jupyter. Ultimately, the subgroups for years of experience that favor Jupyter (1-5 years) seem to be a sweet spot for students investigating preferred IDE for the reasons discussed.

3. Advanced Analysis

Methods

As a final method of exploring the popularity of Jupyter Notebook we can attempt to "classify" a survey respondent as either using or not using Jupyter based specifically on their job title. Previous analysis showed that students and industry professionals both had high proportions of survey respondents that actively used Jupyter. To further our understanding of Jupyter's popularity we can explore employment further to understand Jupyter's popularity based on the job title held. Kaggle's survey asked participants to mark titles most similar to their current (or most recent) position, such as "Business Analyst", "Data Scientist", "Student", "Statistician", and other data-related titles. We can similarly explore the level of formal education planned or attained and years they've spent programming, as noted by the survey respondent.Given these categories, naive bayes classification will be performed to estimate the likelihood of their usage of Jupyter.

Analysis

To ensure that there wasn't an overly concerning amount of multicollinearity in the features selected for a model, variance inflation factors were calculated.

Feature	Variance Infation Factor
Education	1.87
Experience	2.2
Occupation	2.63

None of the variance inflation factors for the selected features was particularly high, allowing us to continue with our categorical naive bayes classification. To conduct classification the data was split using a sklearn built-in splitter for training and testing data. 25% of the data was reserved for testing, with the remaining data used to train our classification model.

Classification	Count
Correct	3113
Incorrect	1667

Statistic	Use Jupyter	Does Not Use
Precision Score	0.65	0.68
Recall Score	0.91	0.29
F1-Score	0.75	0.4

Overall Classifier Accuracy: 65%.

Discussion

After training the categorical naive bayes classifier with a set of over 14, 000 observations and testing it on a set of about 5, 000 observations we arrived at a classification accuracy of only 65%. Although not exactly the highest accuracy, the poor performance of the classifier is not unexpected. Naive bayes firstly assumes independence, which in real data such as this is not completely achievable. Moreover, it may be indicative that the choice of using Jupyter Notebook is not well correlated with a respondent's educational level, occupation title, or years of experience coding. Of greater interest is the specific statistics related to the classification report. The precision score was similar between both groups (using and not-using Jupyter). As the precision score is the ability of a classifier to prevent itself from negative samples as positive. We see that for both groups the performance was not incredible, indicating that the chosen features alone may not be sufficient to predict a user's Jupyter use. The recall scores for both groups were drastically different. With a recall score of 0.91 (with one being the best possible value), the classifier was adept at finding users who use Jupyter products. However, it was also very bad at correctly marking users who did not use Jupyter. These two metrics are combined in the F1-score, which is a weighted average of the precision and recall. Notably, the F1 score shows that the classifier is significantly better for those using Jupyter than for those not using Jupyter. Suggestions for further classification would be based on programming language of choice, though the heavy presence of Python and R in data science as a field may create imbalances not healthy for a naive bayes classifier.

4. Conclusion

Jupyter is a set of tools commonly used for data science in both university classrooms and in professional settings. Given its reputation, this paper aimed to see just how popular Jupyter products are, especially beyond the context of the classrooms that students here at UCSD know. To answer this question we turned to the 2020 Kaggle Machine Learning & Data Science survey. From the data, initial findings supported our assumptions: Jupyter notebook was extremely popular among both students and professionals alike, with an estimated usage of over 50% for both categories. With a range of estimated percentage difference between 3 and 8%, Jupyter products seem to dip slightly in popularity between students and industry professionals, but remains widely popular overall. The next angle of analysis was to compare the popularity of Jupyter both domestically and abroad, overall and for students specifically. In both cases, with p-values over the cutoff of 0.05, binomial tests failed to support a significant difference in the proportion of students or total respondents in the US being different from the respective category abroad. Therefore, we can conclude that Jupyter is likely equally popular internationally as it is domestically, holding widespread recognition as a tool suitable for data science. Additionally, we explored how experience with machine learning impacted a respondent's usage of Jupyter related products. With a null hypothesis that a user's machine learning experience and preference for IDE being independent, a chi-squared test of independence was conducted. There was insufficient evidence to support this null hypothesis, indicating that the two are likely dependent. Graphical breakdown of popularity for IDEs based on machine learning experience showed that Jupyter was significantly more popular among respondents that had experience with machine learning, taking up a proportion of almost 60%. Its surprising popularity among those with machine learning experience shows that Jupyter is a trusted tool for handling work in a wide field like machine learning, and is a suitable tool for students to pick up for machine learning. To predict Jupyter usage solely based on respondent programming experience, a logistic regression model was used. Notably, it found that Jupyter was widely popular among programmers with less experience, indicating it would be a solid choice for newer programmers (such as students). While still popular overall, there was a noticeable drop off in popularity for users with over 10 years of programming experience. Lastly, categorical Naive Bayes was performed to do preliminary assessment for prediction of Jupyter usage given several features of interest. The classifier achieved moderate levels of predictive accuracy, at 65%, but had significantly varying levels of performance: doing very well for predicting a user's usage of Jupyter, but very poorly for predicting when a user might not use Jupyter. This can be reconciled by IDE potentially being dependent on programming language of choice, though oversaturation of Python prevented us from fitting a model to accurately test this. Overall, our findings support the notion that Jupyter is widely popular, and finds ample reasoning to suggest that it can be recommended to students. As a popular tool among industry professionals and students, and as a tool capable of handling tasks like machine learning, Jupyter places itself as a jack of all trades in the data science field and has justified its own widespread use.

5. Work Cited

1. Kluyver, Thomas. “Jupyter Notebooks in Higher Education.” *GenR*, 9 Feb. 2021, genr.eu/wp/jupyter/.