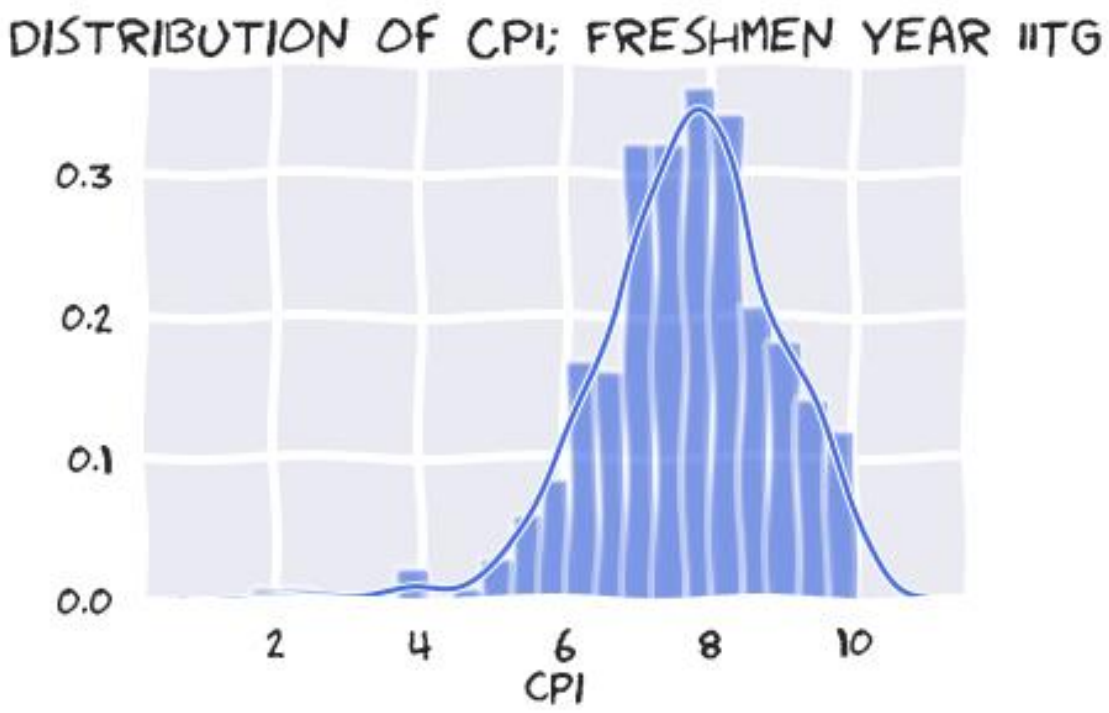
CnA Data Visualization Hackathon – Aditya Patil, 2nd Year CSE

Student Data Visualization

Attack of the plots



Introduction

Why is Data Visualization Important? What do we get out of *fancily* plotting features when we already have powerful libraries that can directly build a powerful model at our discretion?

Well then, how many times have you called this powerful model a black box? I'm pretty sure that if you know the term, you've used the term. And there's nothing wrong in it because that's how far ahead in the future we are. Are we not?

Anyway, not understanding your data, and hence, not being able to use this insights in making your model more powerful, not only makes you a bad Data Scientist, but also yet another blind implementer of an algorithm that you most probably don't understand.

Diving into it

So, without further adieu, let's be good practitioners and dive into the dataset.
The semi-cleaned Dataset consists of information about 379 first-year students that could possibly, in any way, be a factor in their CPI of the first semester.

Throughout the report, I've compared the basic intuition one has to what the data shows.
Since the overall motive of this report is to help the students make informed decisions, I've focused more on columns that are a *choice* to the student rather than meta-data about them. Having said that, all the important columns have been explored.
Sometimes consistent and at others, just extremely contradictory, as a whole, we IITians **prove ourselves to be pretty predictable** in rather unorthodox ways.

I'll follow the student along his journey to and in IITG. And then end my report with a Dendrogram and Feature Importance plots of my model.
I've used the plt.xkcd() style as base and topped it with sns-darkgrid for clarity.

It has been assumed that the basic interpretation of the types of plots is well-versed to the reader.

Cleaning and Preprocessing -

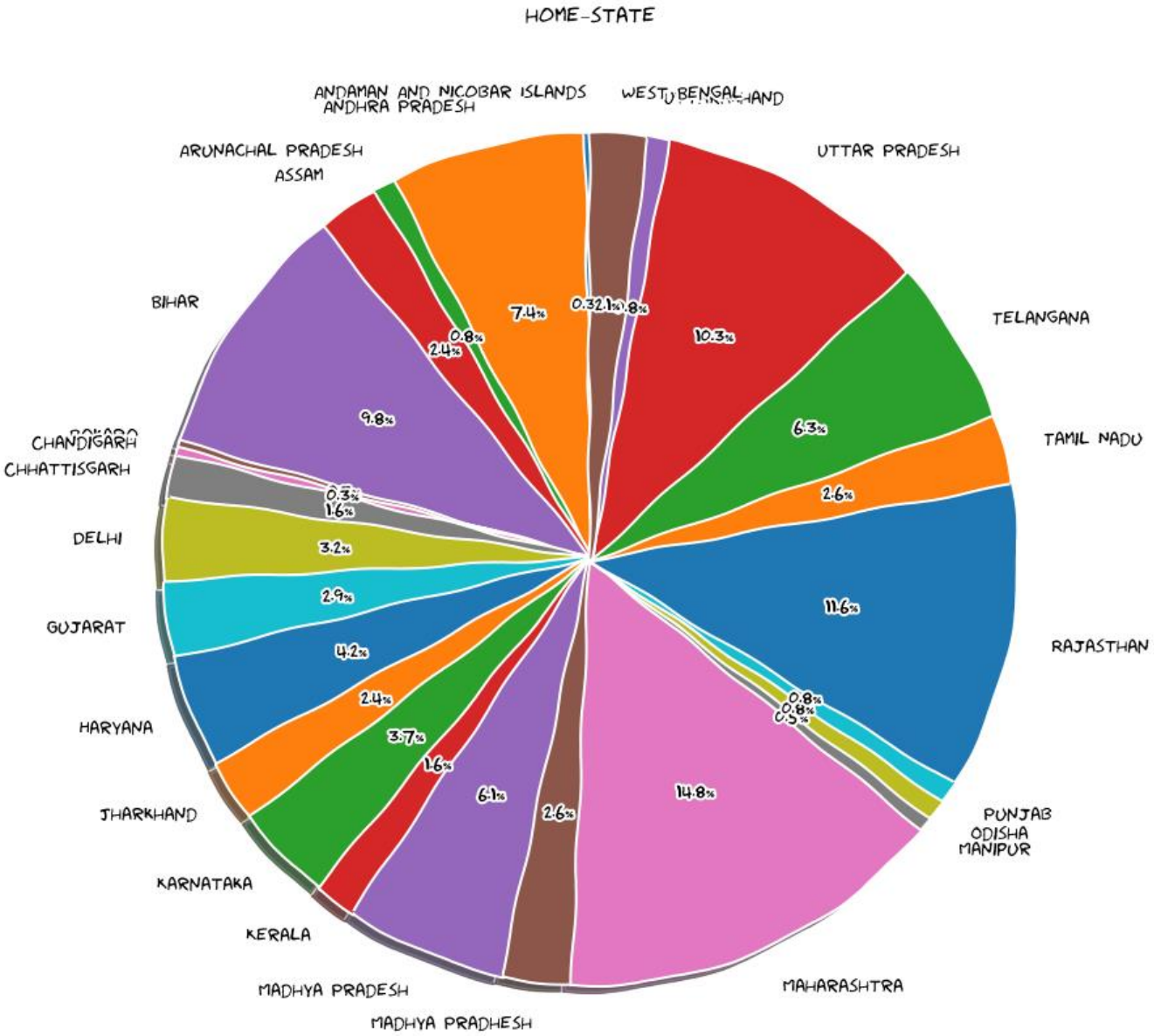
1. Replaced all the NaNs in the Dataset with "N.A"/'None' as per necessity.
2. Capitalized all the string features using `df[col].str.title().str.strip()`
3. Replaced the duplicates and mistyped values in the 'home_state' and 'coaching_city' columns as they were highly cluttered.

As Fests, Cultural Clubs and Technical Clubs were multivalued columns, I exploded them making a separate binary feature out of each club/fest using the following code:

```
cults=['Debsoc','AnR','Cadence','Litsoc','Montage',
                'Lumiere','Octaves','Drama club','Fine-Arts','None']
for club in cults:
    df[f'cult_{club}']=df.cult_clubs.str.contains(club).astype('int')
df['cult_num']=df.cult_clubs.str.split(',').apply(lambda x: len(x))
df.cult_num[df.cult_None==1]=0
```

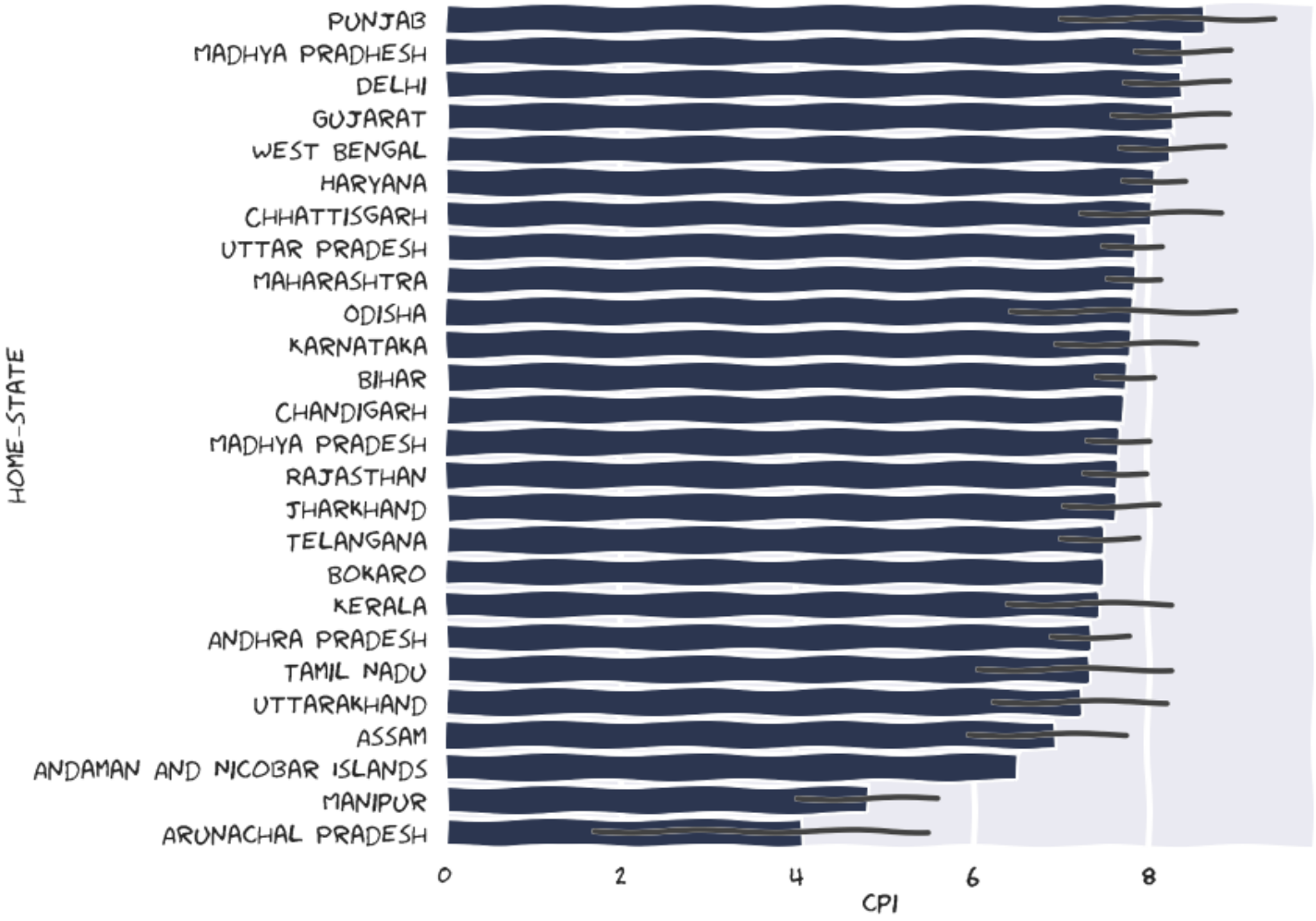
For Example, cult_Cadence, fest_Alcher, tech_FEC are some of the new columns.

Back to the Beginning



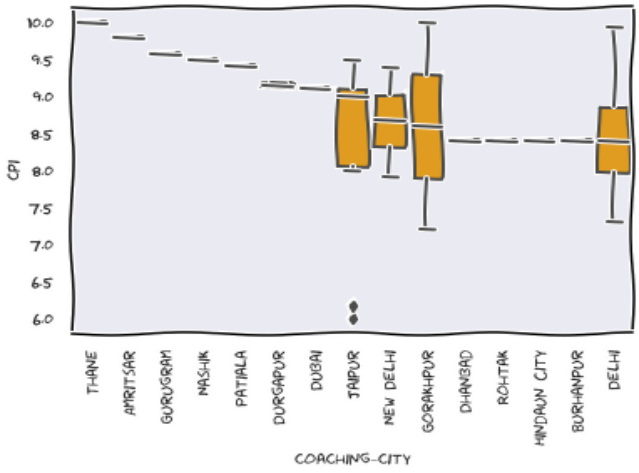
Well, I know Data Scientists are supposed to hate Pie Charts, but I apologize they're still going to pop up wherever they serve their purpose. Before getting into how each feature contributes directly or indirectly to CPI, it is important to understand how the data is distributed internally. The graph shows the students of Punjab to have the highest mean CPI (albeit high STD), although

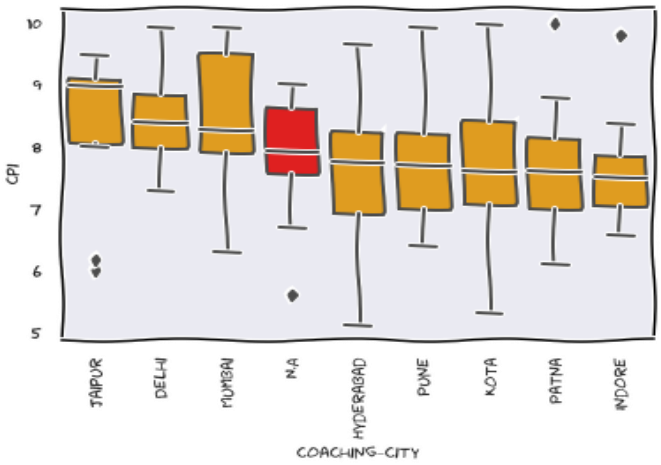
this inference is incomplete without knowing that the students from the state form only 0.8% of the Data i.e only 3 students.



From the states that have a good enough population in the junta, students from Uttar Pradesh and Maharashtra have managed to maintain an average CPI of around 7.9 while the 3 students from Arunachal Pradesh have brought their state's mean the lowest. (Notice the high standard deviation, one of them must've performed well).

The Coaching City

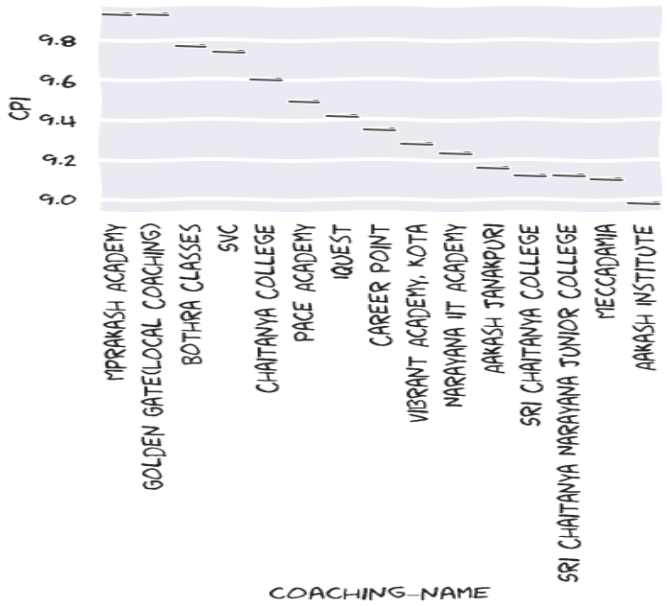


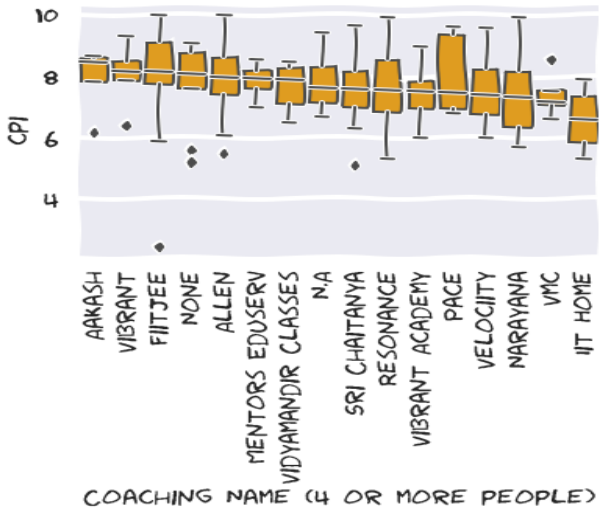


As all of us IITians know that our Coaching is as important a factor as any when we step foot into the college. The first graph shows which Coaching cities enjoy the highest median CPI's. But most of these have just one student hailing from them who's managed to get a good CPI.

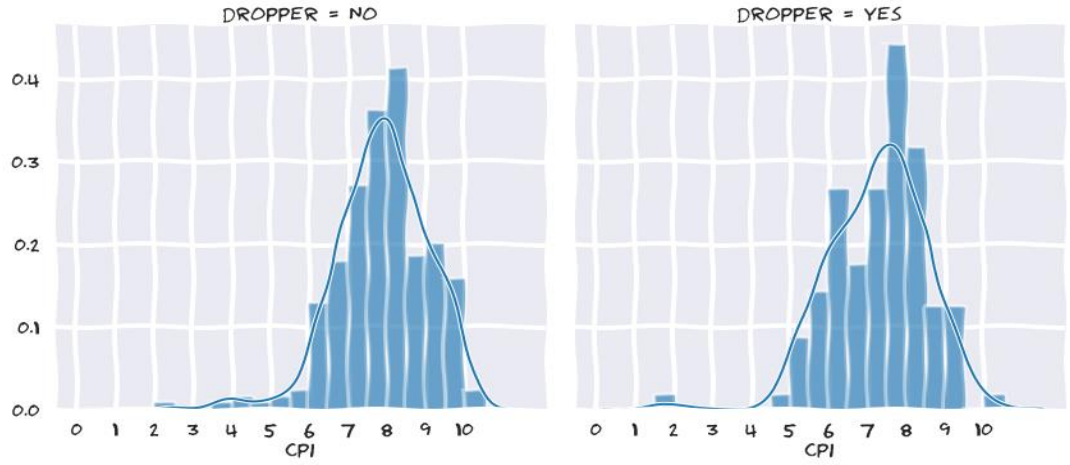
For greater insight, the second graph shows the cities where **more than 10 students** have studied. Jaipur is the clear winner with Delhi as *not-so-close* second. While the rest of the results are not surprising, notice the 'N.A' in red, its pretty striking to realize students having self-studied their Way to IITG maintain a better median CPI than Kota, Hyderabad and Pune peeps (Notice the max of N.A is pretty low anyway).

The Coaching Class





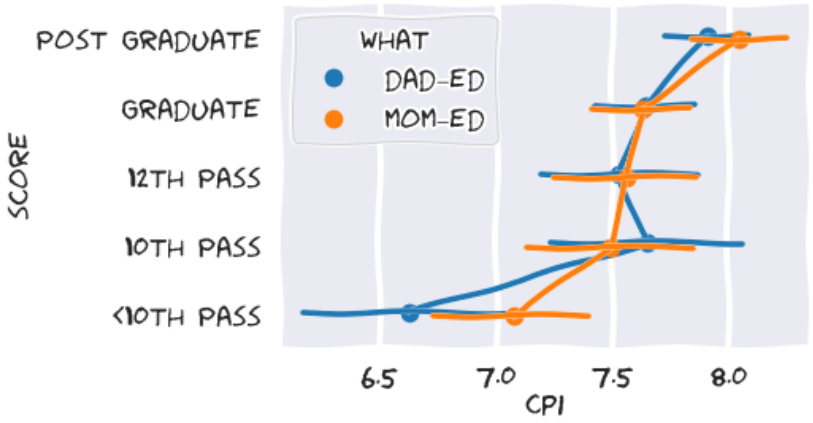
M. Prakash Academy does it as always (*takes a bow*) and I don't think this is going to change any year soon. But from Institutes that have a better population, Aakash takes the stage. By the 14th Institute, the median CPI has gotten rather close to 7.



Lastly, with only small differences, the not-a-dropper graph is observed to have a steeper curve that peaks at 8, while its contemporary is flatter and peaks at 7.5

Education Factor

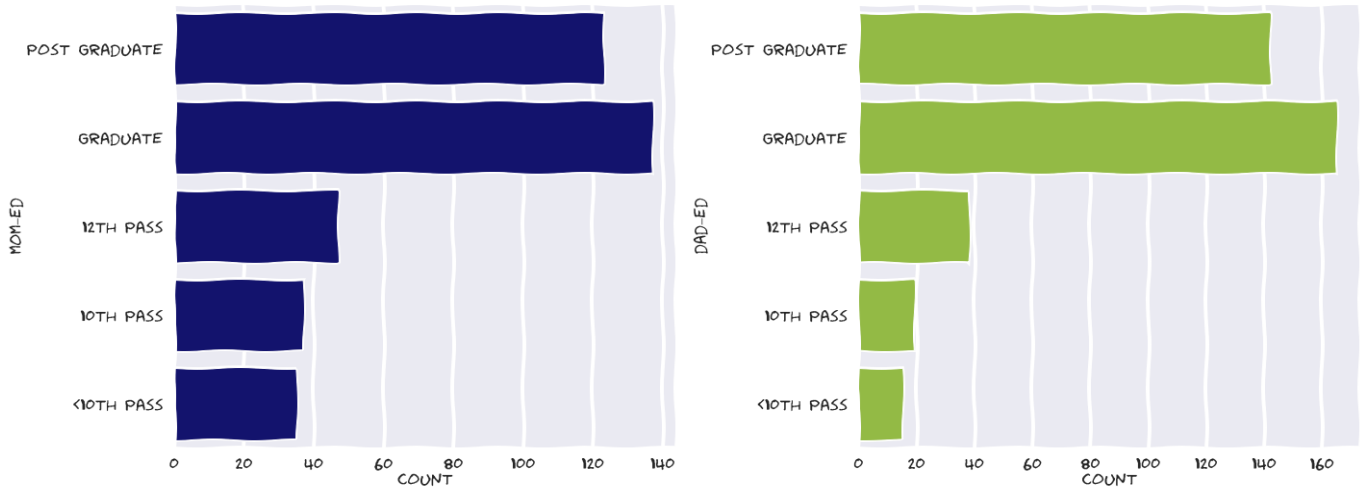
Let's see how the child prodigy's performance depends on how educated his/her parents were -



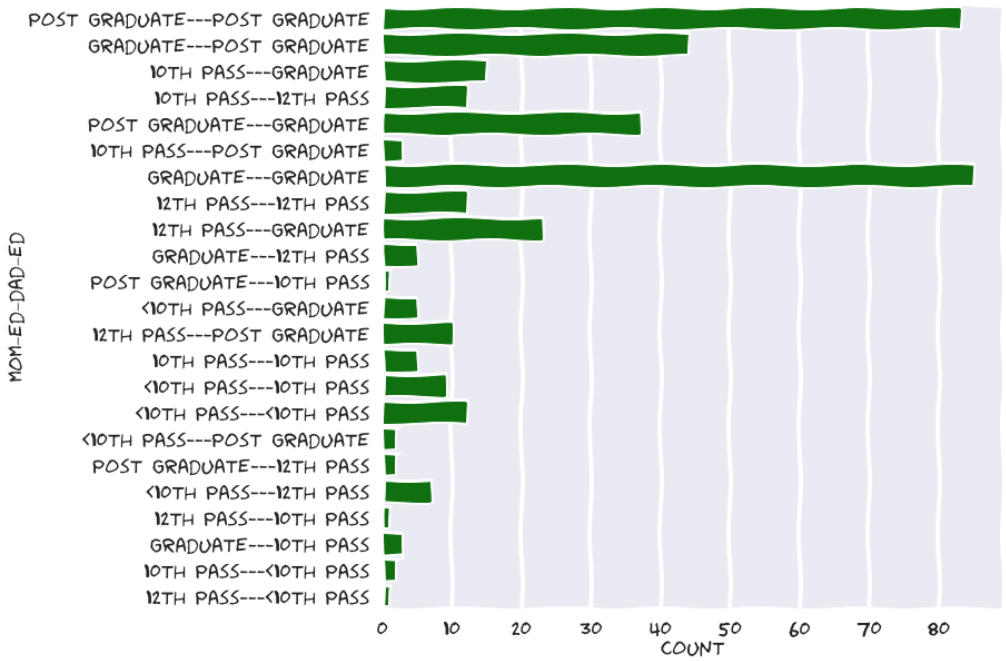
Woah, the trends show that the child's CPI increases directly with their mothers' education.
Highly positive correlation!

And while the trends with the Father's education are a little less uncertain, it's important for us to see that the Father being less educated i.e <10th Pass, has a rather drastic crop to the kid's CPI.

While most parents are pretty educated, a substantial number of them are not.
Although the overall shape of the graphs are similar, around 80 Dads are below-graduate while close to 120 moms are below-graduates.



This interesting trend led me to combine both these features and plot CPI against the "Parent's Education" -



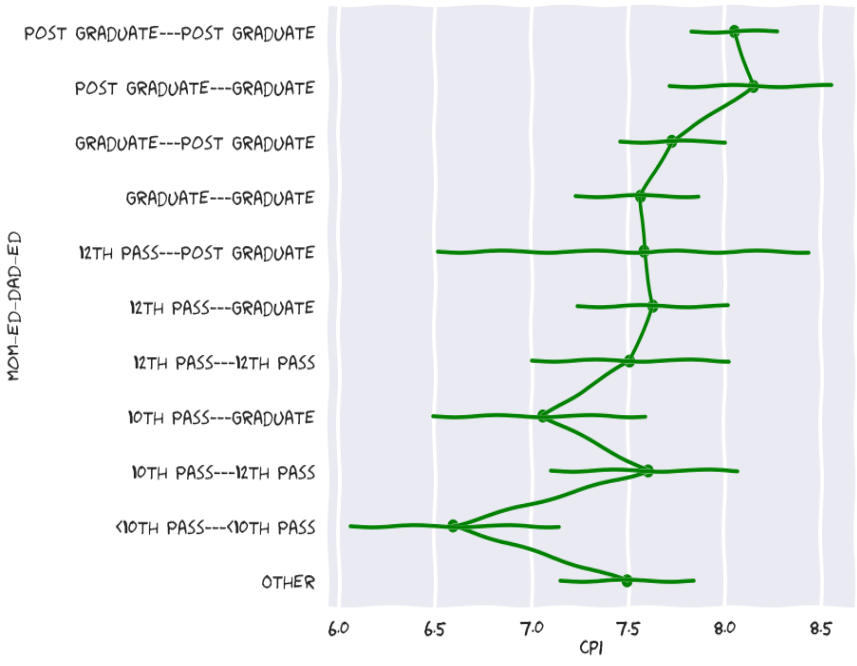
This is the distribution that this new feature showed, mom's education first.

(Notice that nothing is impossible in India, but equal education shows the highest number)

I replaced combinations with a frequency of less than 10 with "Other" and plotted this against CPI.

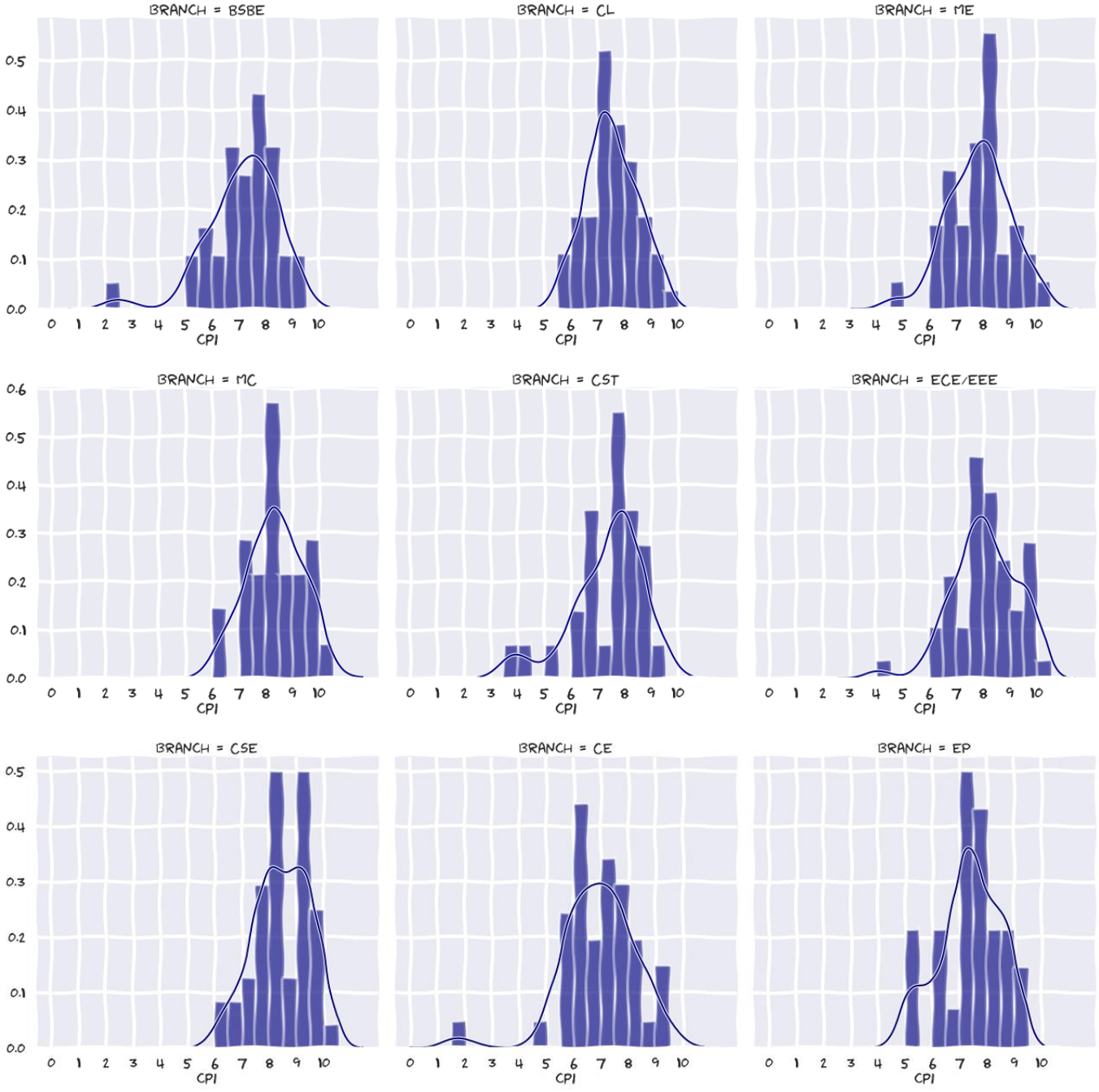
The Labels were arranged in increasing order of Education, mother's first because CPI seemed to depend on it more than the Father's (and not because of Feminist reasons)

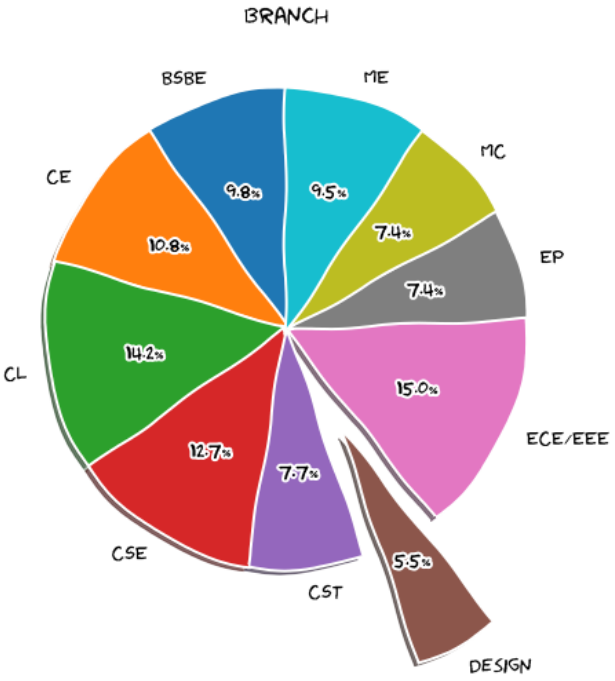
While both parents being less educated certainly shows a lower CPI, the combination of Post Graduate - Graduate seems to be the most promising for your child.



Entering IITG - The Branch Factor

According to me, FacetGrids are the most successful in providing the maximum amount of insight into the data. And so, let's look at how CPI soars and falls through every branch; facet by facet.





Similar Distributions, peaking between 7-8.

1. Only CSE has two peaks, one at around 9.2..!!
2. Civil peaks the lowest at 7.0 while MnC the seconding CSE at 8. 4
3. The CL and EP have rather steeper curves indicating more people closer to the average.
4. The Students with the lowest CPI's are in BSBE, CST and ECE/EEE.
5. The dataset doesn't have equally sampled branches.

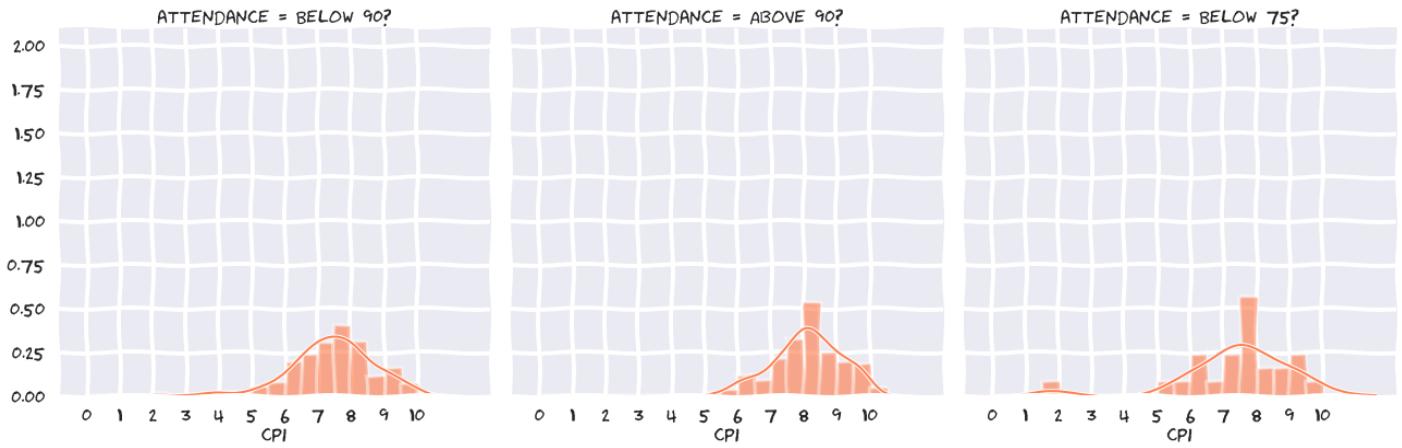
And you're in

It is our choices, Harry, that show what we truly are, far more than our abilities

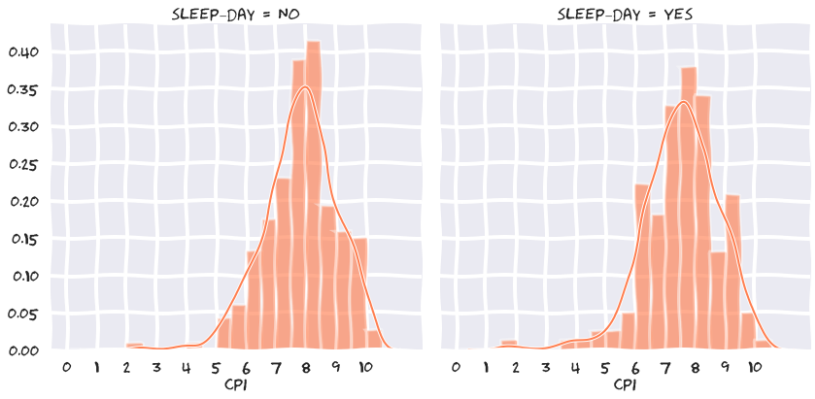
-- Albus Dumbledore

Shall we let EDA decide the truth value of Dumbledore's statement?

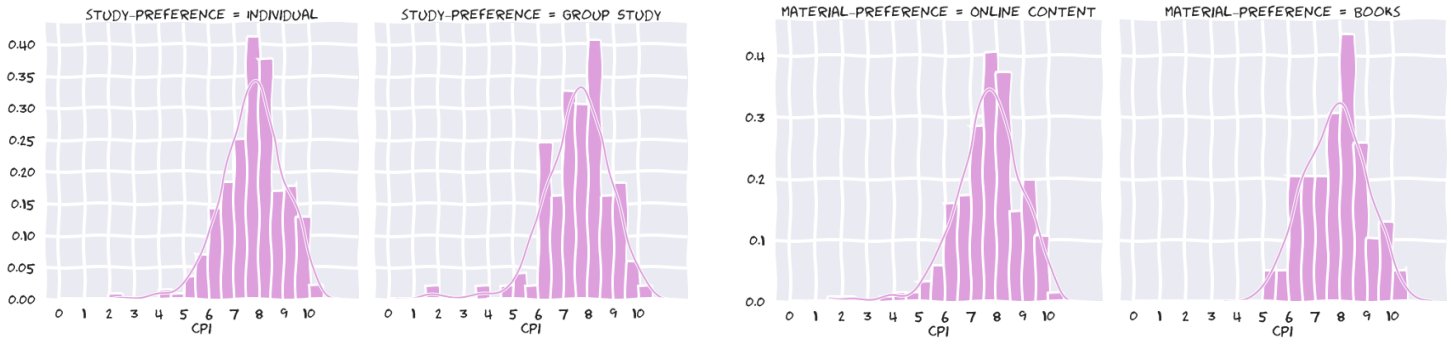
A quick rapid-fire of some FacetGrids to explore some of our most important features -



Most people attending more than 90% classes have a above 8 CPI, while the curve peaks at 7.5 at the other two branches. The below 75? Curve is reasonably flatter.

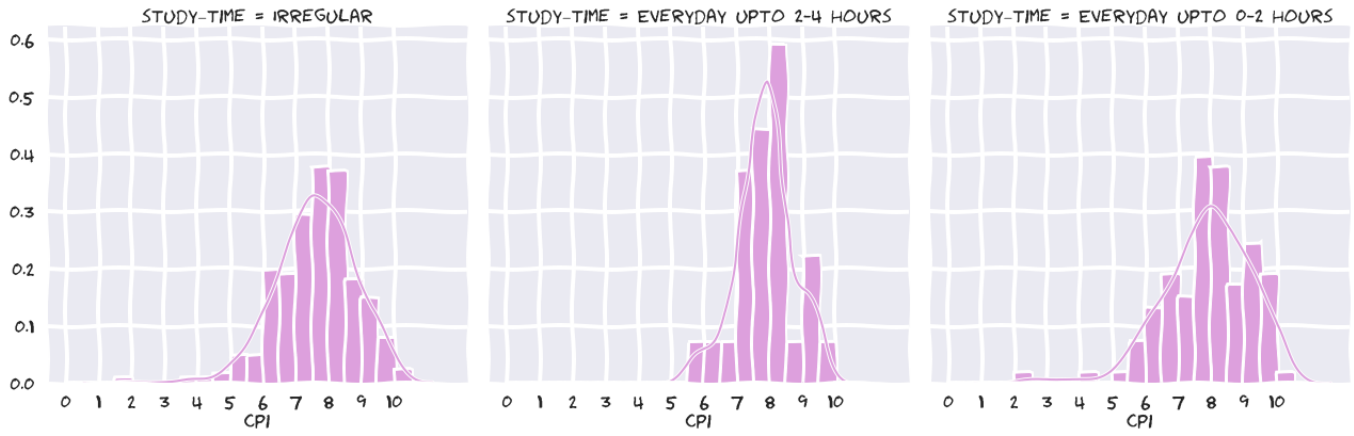


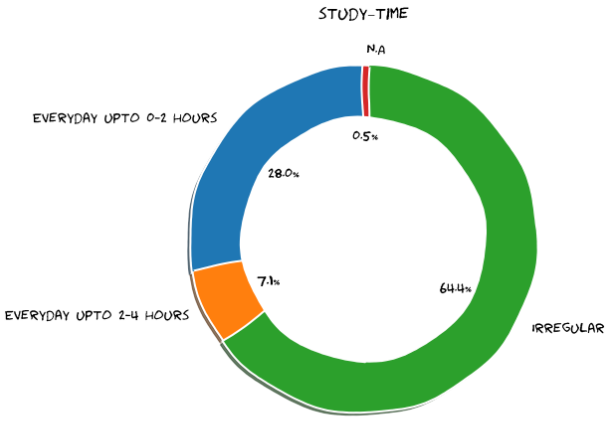
Who'd thought this could make a difference? The No Sleep Day curve is steeper and peaks precisely at 8 while that of people who take a sleep day is flatter and peaks at 7.5



While the curves peak at the same value, a larger proportion of people preferring books are below 7.5. Smart-working done right?

People studying are slightly at an edge than those who study individually, but the difference looks minute and it's difficult to determine whether or not this matters.





Why am I not surprised?

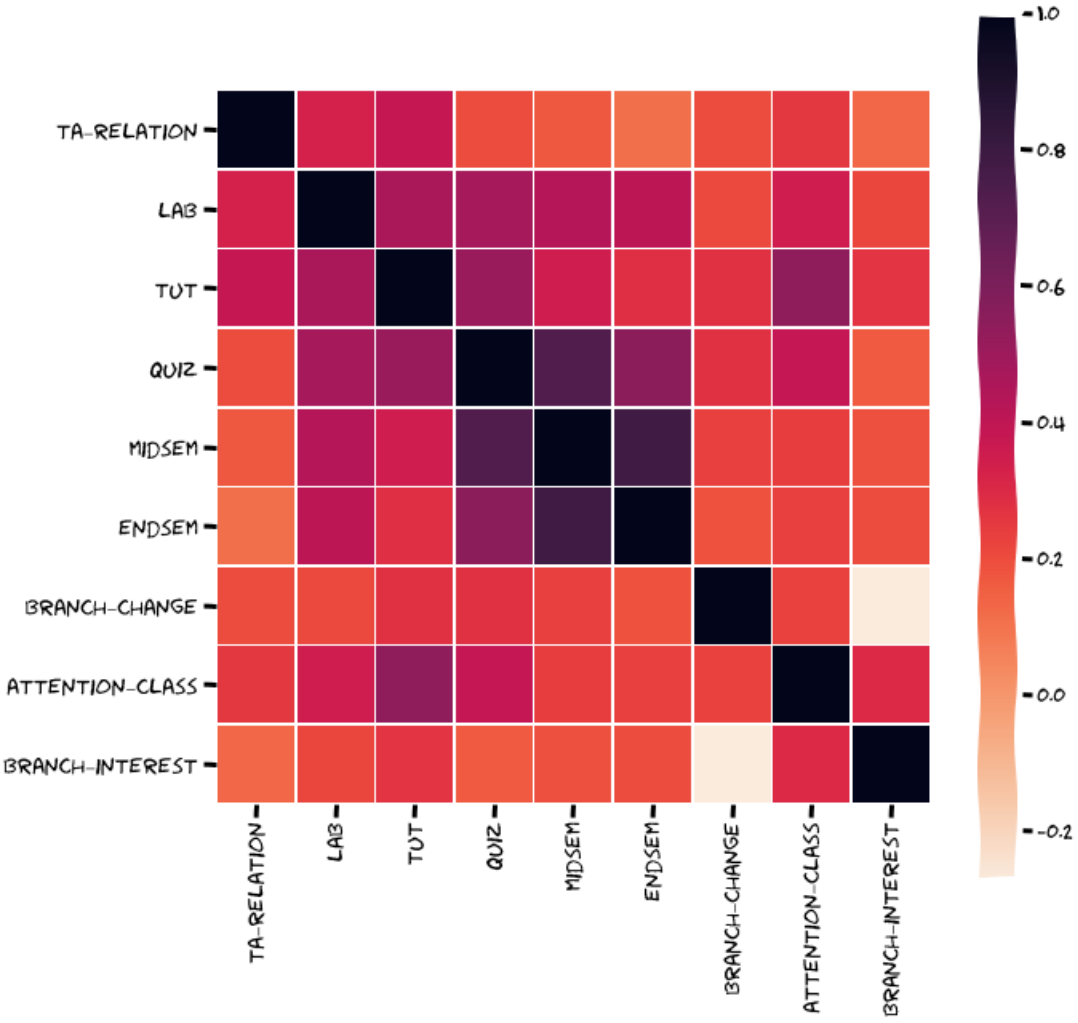
On one side, it's funny to see that the people who study everyday for about 2 hours are in the same, if not worse, statistic bracket as the ones who study irregularly.

On the other, it's frightening to see the steepness of the ones studying regularly for 2-4 hours. When I first saw this graph, I thought how many actually do that, and my guess was 3-4%.

(Sorry for the Donut Graph) 7.1% means around 25 from a data of 379. Not as low as one might expect!

But then, we live in an IIT after all.

The Interests



The above heatmap shows the correlation between the 'Interest Columns'

While most columns see mild positive correlation, the trends are more or less expected.

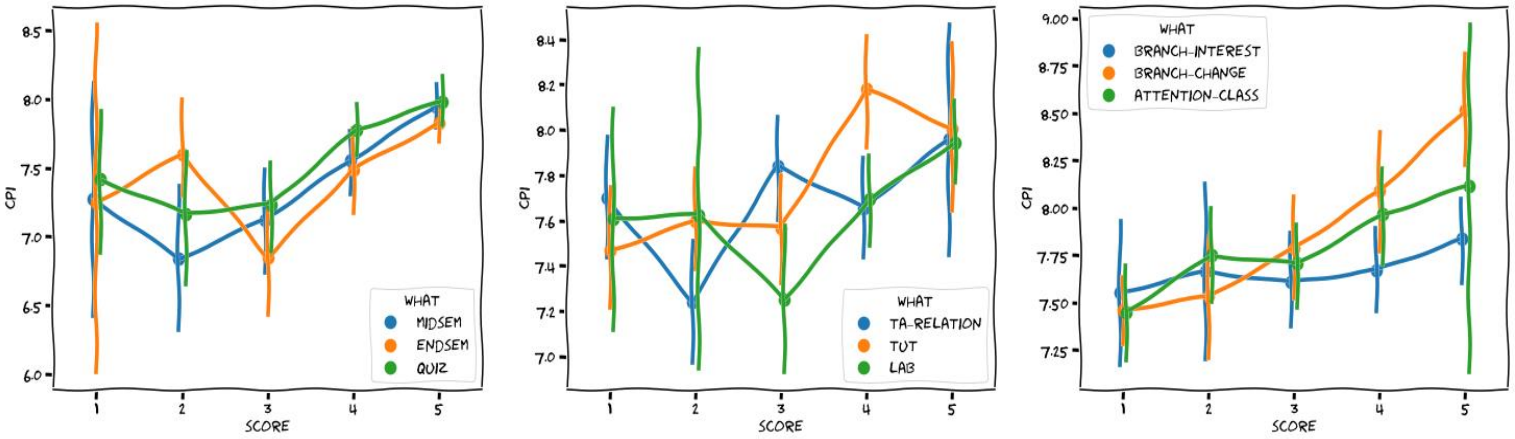
1. Lab increases with Tut, Quiz, Midsem and Endsem

2. Quiz, midsem and endsem have a better than average correlation of about 0.8

3. Attention in Class also seems to increase with Lab, Tut and Quiz, but not as much with Endsem and Midsem

4. The most interesting and honest square that I found was the white one at the bottom right corner. Branch Interest drops as interest in branch change increases.

Let's explore how these affect the CPI.



Most of the trends are abrupt and it's difficult to draw a lot of inference from them.

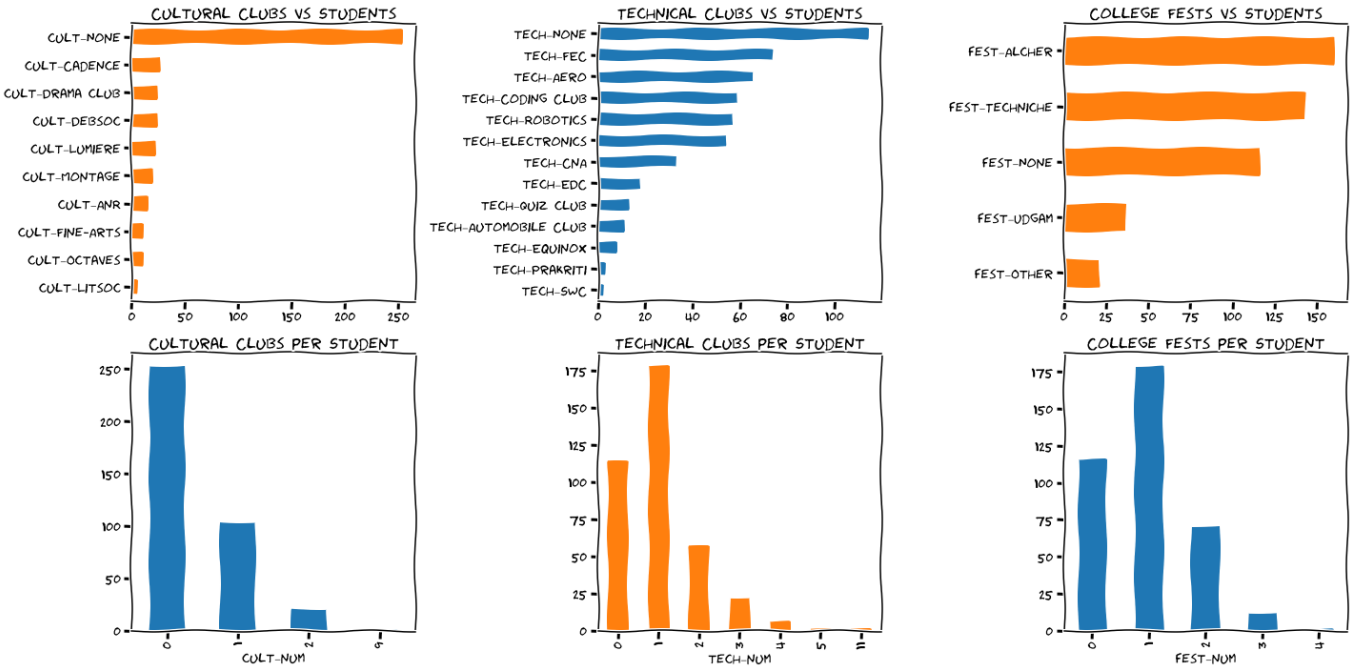
Turns out not being interested in exams does not directly lead to a low CPI

The third graph is interesting – CPI vs **Branch Interest** line has a small positive slope meaning, being interested in your branch does work slightly in your favour; and the same goes with **paying attention in class.**

But the single highest motivator to a good CPI remains **Branch Change**, and this will be shown again when I do feature importance.

The Extracurriculars

Before looking at how each activity affects one's CPI, or if it really does, I want to explore how many students did go to clubs and work in festivals and which of these attracted more people?



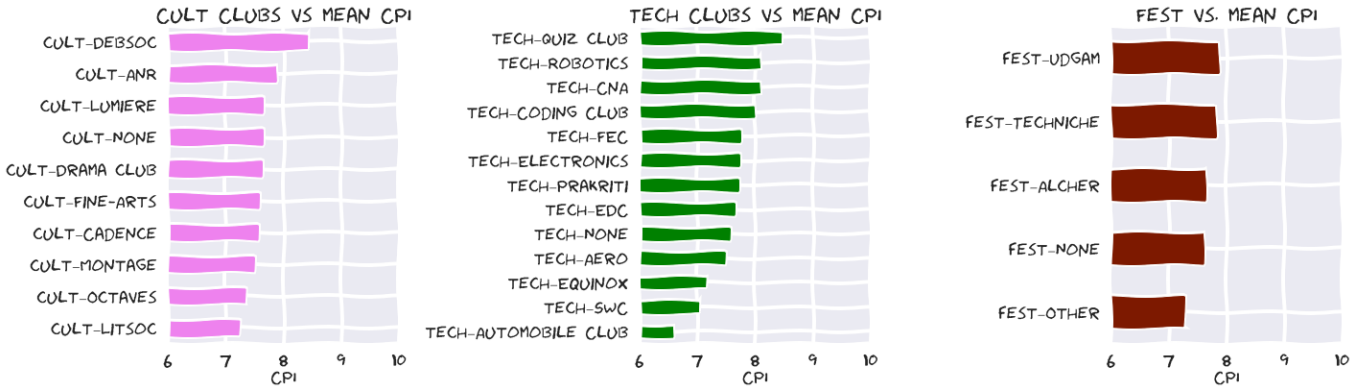
The first column of the above plot shows Cultural Club statistics. The first is the number of students going to each club and the second is the the number of clubs chosen by students. Around 66% of the students hadn't been to a cultural club by the time of Data collection.

would it be out of place to say that this batch was disappointing from the beginning?

Cadence attracted the maximum of the left-out ones and Litsoc the least. Most of the students going to clubs went to only 1 cultural club while a very few to two. There was someone who's said they've been to 9 :-)

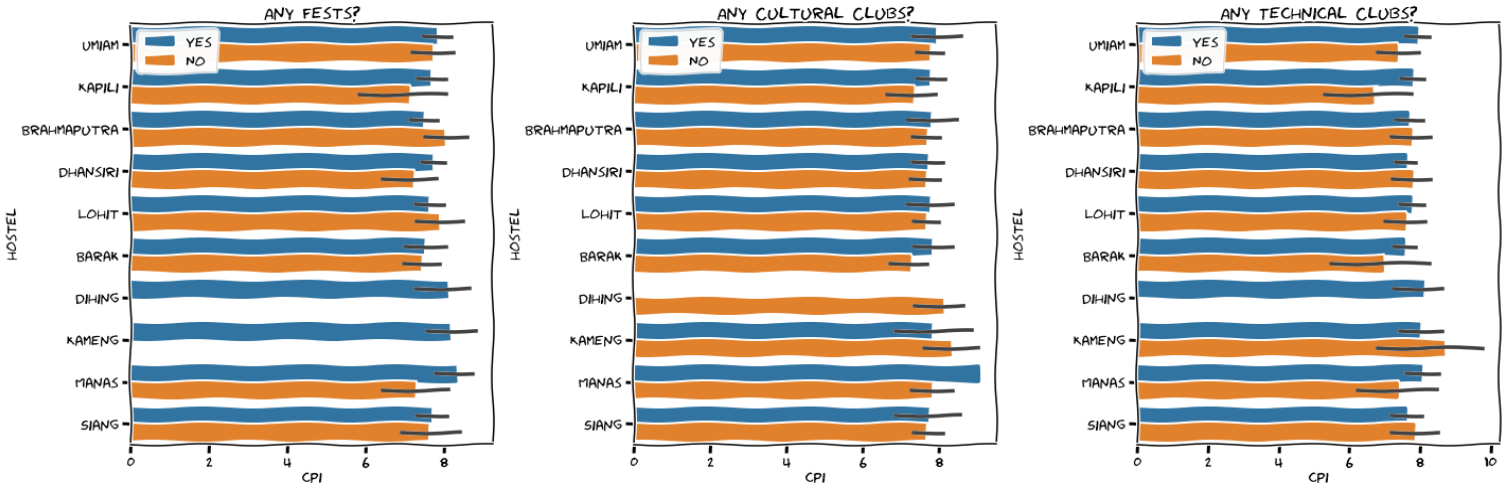
Technical Clubs are in a much better position with only 30% never having visited one. The Finance Club attracted the most freshers while Prakriti and SWC the least.

A similar trend follows with fests with 30% never having worked for one while a lot of freshers joining Alcheringa and Techniche.



Now, to CPI, while it can't directly be said that going or not going to a particular club increases or decreases one's CPI, it might still prove helpful in determining one's CPI by looking at the activities he's engaged with.

Example, going to DebSoc won't increase your CPI but rather if you go to Debsoc, you might be intelligent enough to get a good CPI. And from the tiny amount of people who visit these clubs, the clubs enjoying the highest rankers are the Debsoc and the Quiz Club. As for fests, not a lot of difference is observed but Udgam tops while the people not joining fests seem to have a lower CPI.

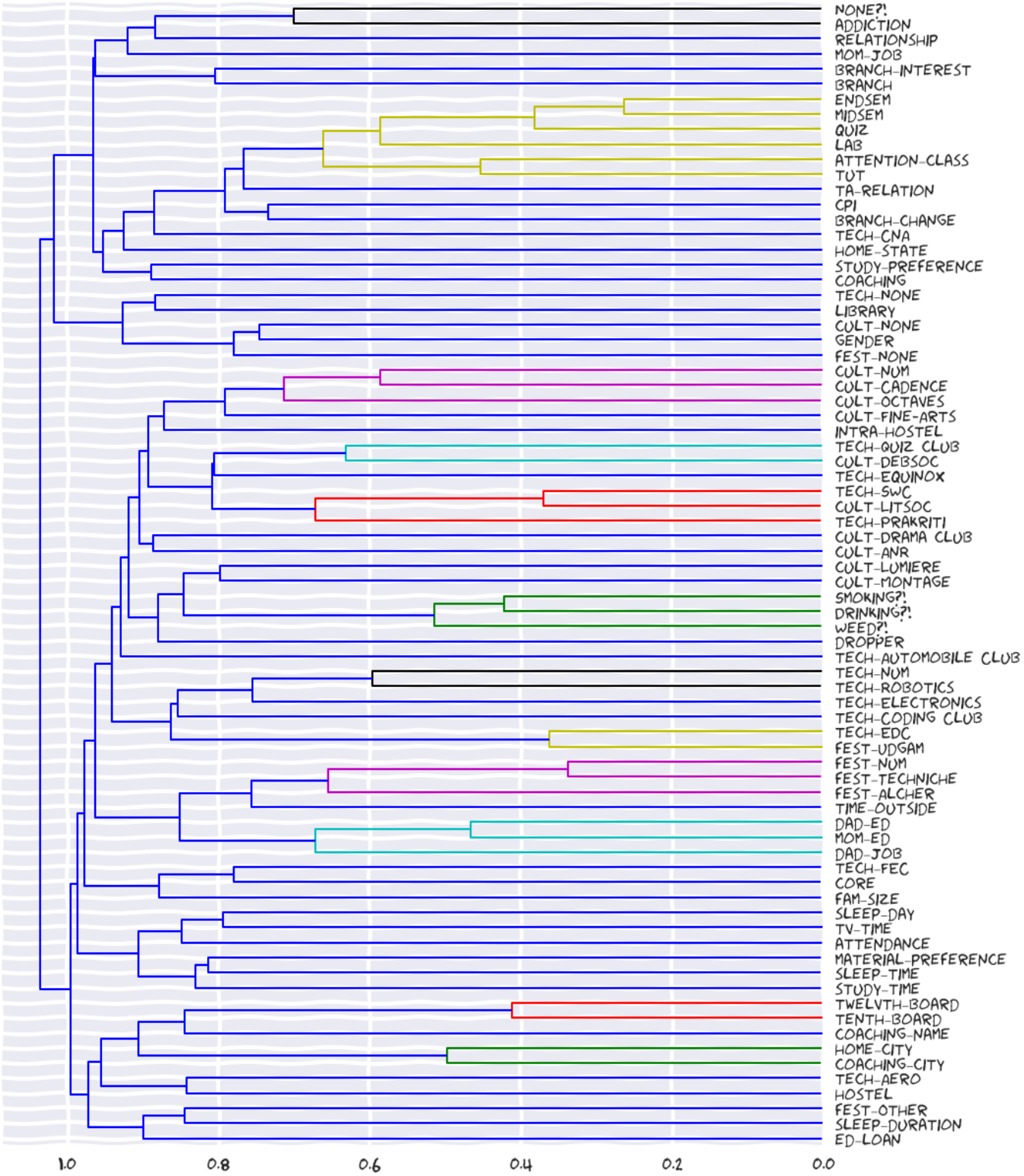


This next interesting plot shows how going or not going to clubs/fests affects the students of every hostel. And guess what, it does not that much of a difference..

And wherever it does, more often than not, most hostels perform better when they take part in extracurriculars.

As for Dumbledore, I'll just say your CPI does not define who you are and hence the analysis was invalid.

Bringing it all together –



The **Dendrogram** on the previous page was made using a clever trick of how the close the features are to each other according to the Spearman-Correlation technique. The Dataframe was label-encoded beforehand.

No model had to be passed. To interpret the Dendrogram – The distance of the tree leaves from the split is the measure of closeness.

- If the split is very near to the leaves, the features are highly correlated
- If the split is closer to the root, the features don't observe high similarity in trends.

Some interesting insights –

- The Interest columns – Endsem, Midsem, Lab, Tut, Quiz, Attention are in the same subtree i.e closely related.
- Branch Interest is more related to the branch you're in than anything else.
- Mom-ed and Dad-ed are very near to each other and shows interdependency. Interestingly, Dad's job is also related to these features, while the Mom's Job doesn't really depend on her education. Typical Indian Society?
- Smoking, Drinking and Weed have found each other in the wide plethora of columns (Only 13 students agreed to addiction out of 379)
- Lumiere and Montage are neighbours, and so are Quiz Club and Debsoc, EDC and Udgam.
- Most cult clubs, tech clubs and fests are in their own group suggesting a similar pattern between the students joining these.
- Sleep Day is close to TV time and Attendance; on some further inspection of the data, it is rather suggestive that people who take sleep days are prone to having more TV time and lower Attendance
- And finally (and unsurprisingly) the closest leaf to CPI is Branch Change.

The Modelling –

I trained two models on two different combinations of Train and Validation Sets.

The first one used the first 200 rows as the Training Set while the next 179 as Validation.

The second was trained on the training set having 200 distinct rows chosen randomly and the rest 179 as the Validation Set. The model used was the LightGBM Regressor due to its speed as well performance (plus feature importance methods readily available)

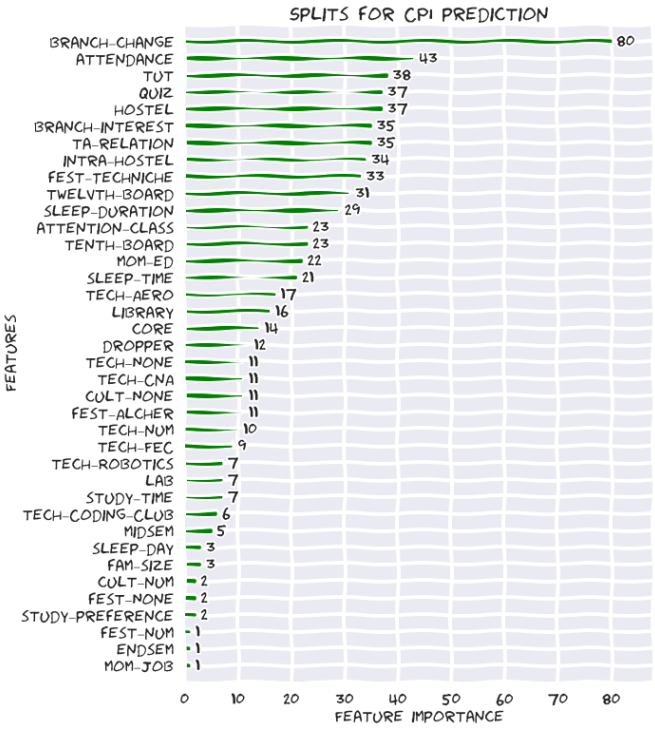
```
params = {
   'task': 'train', 'boosting_type': 'gbdt', 'objective': 'regression',
                'metric': {'rmse'}, 'is_training_metric': True, 'metric_freq': 5,
                'num_leaves': 31, 'learning_rate': 0.02,
                'lambda_l2': 0.0042,
                'feature_fraction': 0.7, 'bagging_fraction': 1, 'bagging_freq': 5,
                'verbose': 1,
}
cats=['branch', 'dropper', 'tenth_board',
                'coaching_city', 'coaching_name', 'home_state', 'home_city',
                'dad_ed', 'dad_job', 'time_outside', 'attendance',
                'tv_time', 'library', 'sleep_time',
                'material_preference', 'core']
cpi_model=lgb.train(params, train_data, valid_sets=[train_data,valid_data],
                early_stopping_rounds=100,  num_boost_round=1000,
                categorical_feature=cat
                )
```

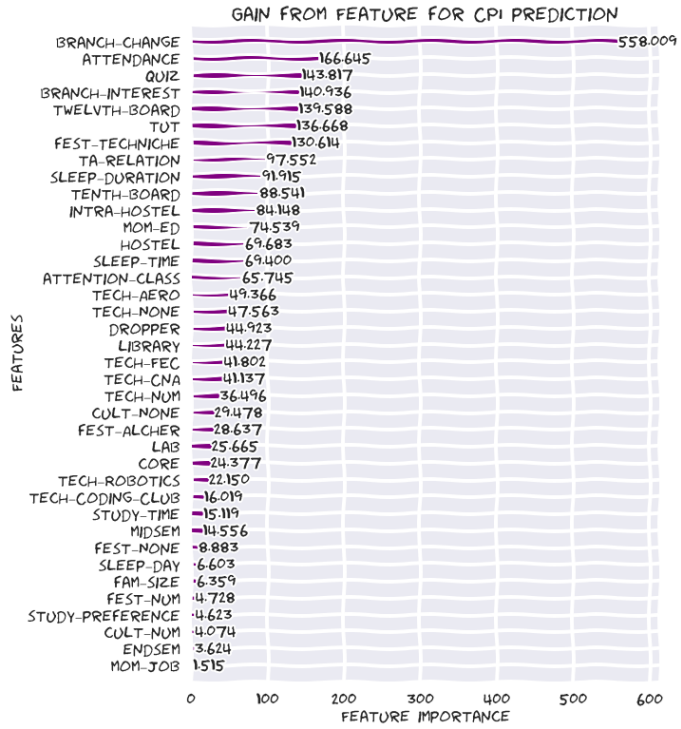
I ran a loop over all probable categorical features and checking whether it functioned better as a categorical feature or as a continuous feature, and hence they were chosen. Because the database was pretty small, this wasn't as time consuming as it normally would've been.

Other Hyperparameters were fine-tuned and I tried my best to keep overfitting in check.

LightGBM, being a tree based model provides two types of feature importance assesements –

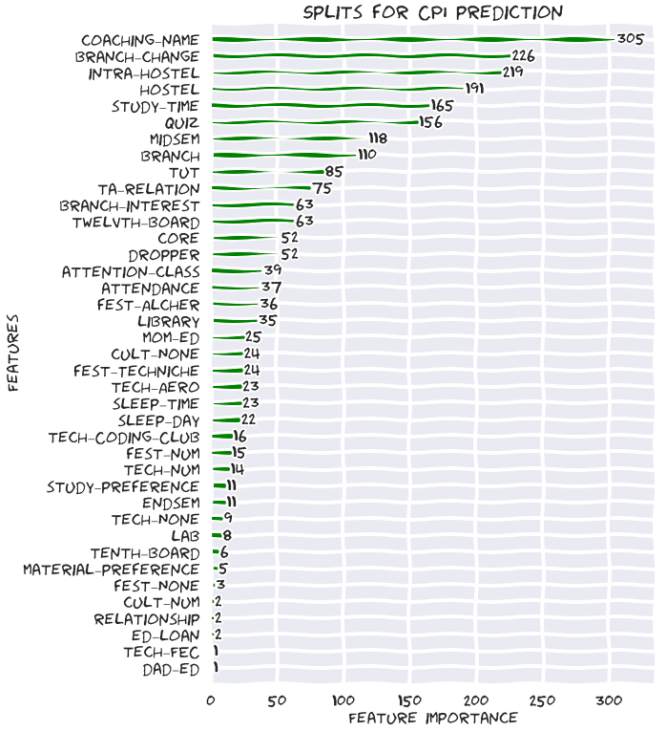
1. Split – The number of splits that were made depending on that particular feature.
2. Gain – How the metric provided improved when the split depending on the particular feature was made.

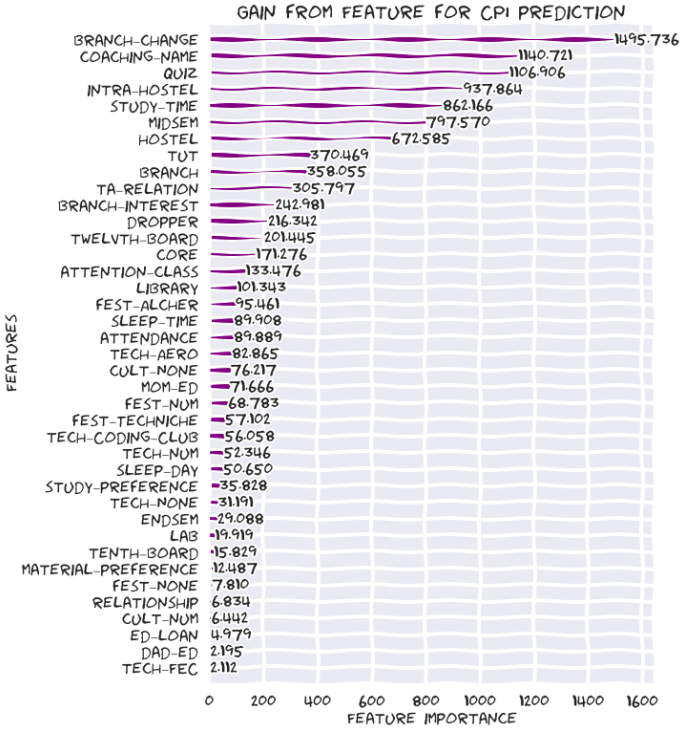




Notice the columns that are entirely absent i.e. no splits were made based on that feature.

I worked a little more on the second model –





The two models show use of different features for prediction. Most striking is that Coaching Name gives the second best split here while it was not used at all in the first model. This suggests that Cross Validation would be a better way to proceed, but due to time constraints, I decided to analyze this model a little more. My initial model after some fine-tuning was –

```
params = {
            'task': 'train', 'boosting_type': 'gbdt', 'objective': 'regression',
            'metric': {'rmse'}, 'is_training_metric': True, 'metric_freq': 5,
            'learning_rate': 0.005,
            'lambda_l2': 0.01,
            'feature_fraction': 0.74, 'bagging_fraction': 0.86, 'bagging_freq': 1,
            'verbose': 1,
}

Cats = ['gender', 'dropper',
            'twelvth_board', 'coaching_city',
            'home_state', 'home_city', 'fam_size', 'mom_job',
            'dad_job', 'ed_loan','mom_ed',
            'time_outside', 'attendance','tv_time',
            'sleep_duration', 'sleep_day', 'addiction',
            'material_preference']

cpi_model_1=lgb.train(params, train_df, valid_sets=[train_df,valid_df],
            early_stopping_rounds=50,
            num_boost_round=10000,
            categorical_feature=cats
)
```

Then I ran the following line of code for another type of feature importance. It's another neat trick that randomly shuffles the values in a column i.e. keeping the distribution same but everything else null and void. The new Validation RMSE is calculated and the difference between the original value and this value is taken in as the feature importance. Because the column is "randomly" shuffled, it is prone to coincidently causing deviated results. So, I took the average of this difference over 20 random shuffles; thereby normalizing it.

This so-called Permutation Importance is more reliable because it shows us how each column directly affects the "results", while Gain or Split show us how each column has affected our "model".

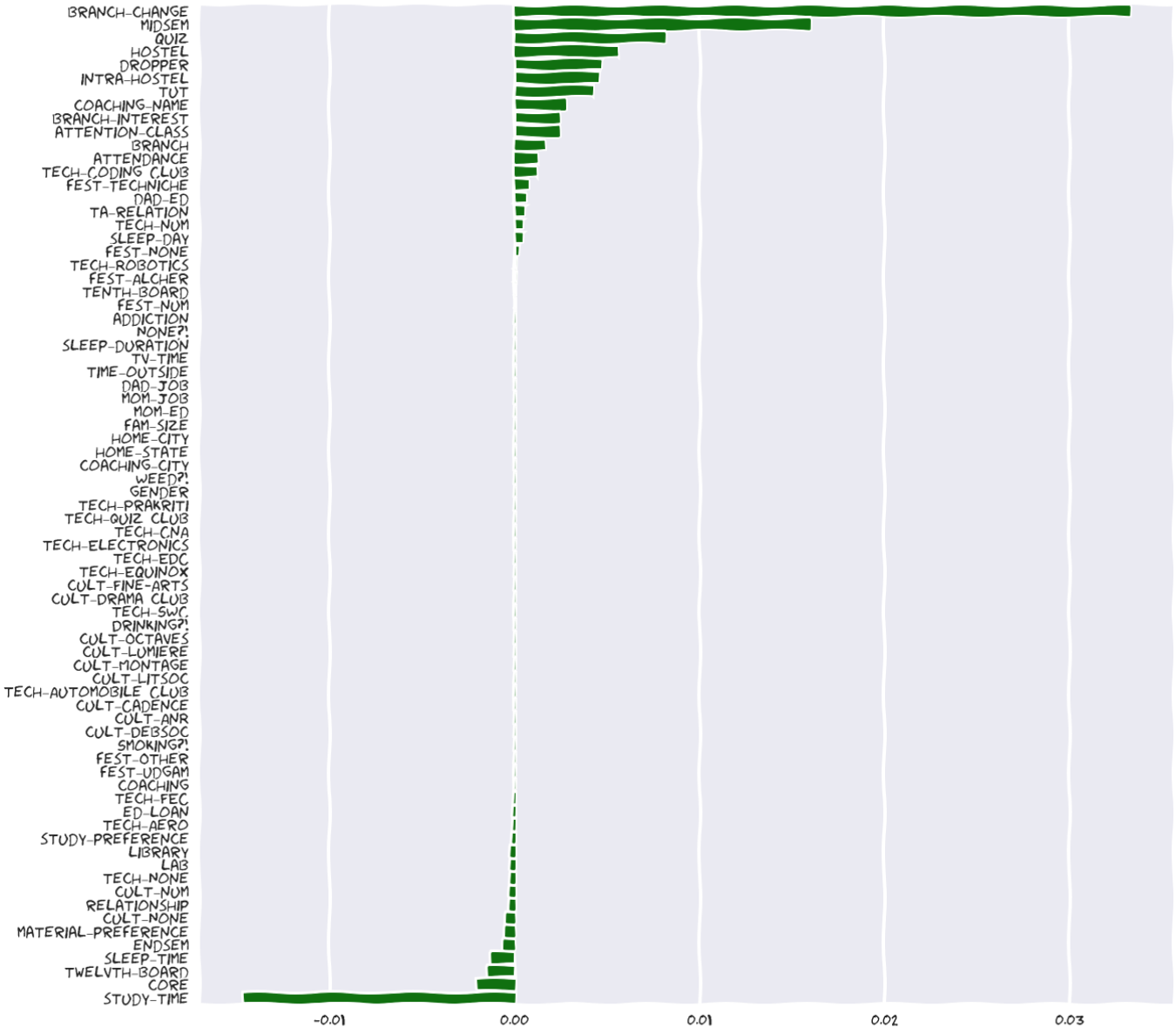
Permutation Importance -

```
%%time
from sklearn.metrics import mean_squared_error as rmse

def score(X, y):
    y_pred = cpi_model_2.predict(X)
    return rmse(y_pred,y, squared=False)

X_valid.reset_index(inplace=True,drop=True)
y_valid.reset_index(inplace=True,drop=True)
# base_score, score_decreases = get_score_importances(score, X_valid, y_valid)
# feature_importances = np.mean(score_decreases, axis=0)

base_score=score(X_valid,y_valid)
fi=pd.Series(index=X_valid.columns)
for col in X_valid.columns:
    X_bot=X_valid.copy()
    diff=[]
    curr=X_valid[col].values.copy()
    for _ in range(20):
        random.shuffle(curr)
        X_bot[col]=curr
        now_score=score(X_bot,y_valid)
        diff.append(now_score-base_score)
    fi[col]=np.array(diff).mean()

fig,axes=plt.subplots(figsize=(6,8))
sns.barplot(x=fi,y=fi.index,order=fi.sort_values(ascending=False).index,ax=axes,color='y');
```

This Importance is computationally intensive as it took 14.4 seconds to run on such a small dataset.

The Validation RMSE after fine-tuned model – 1.32027

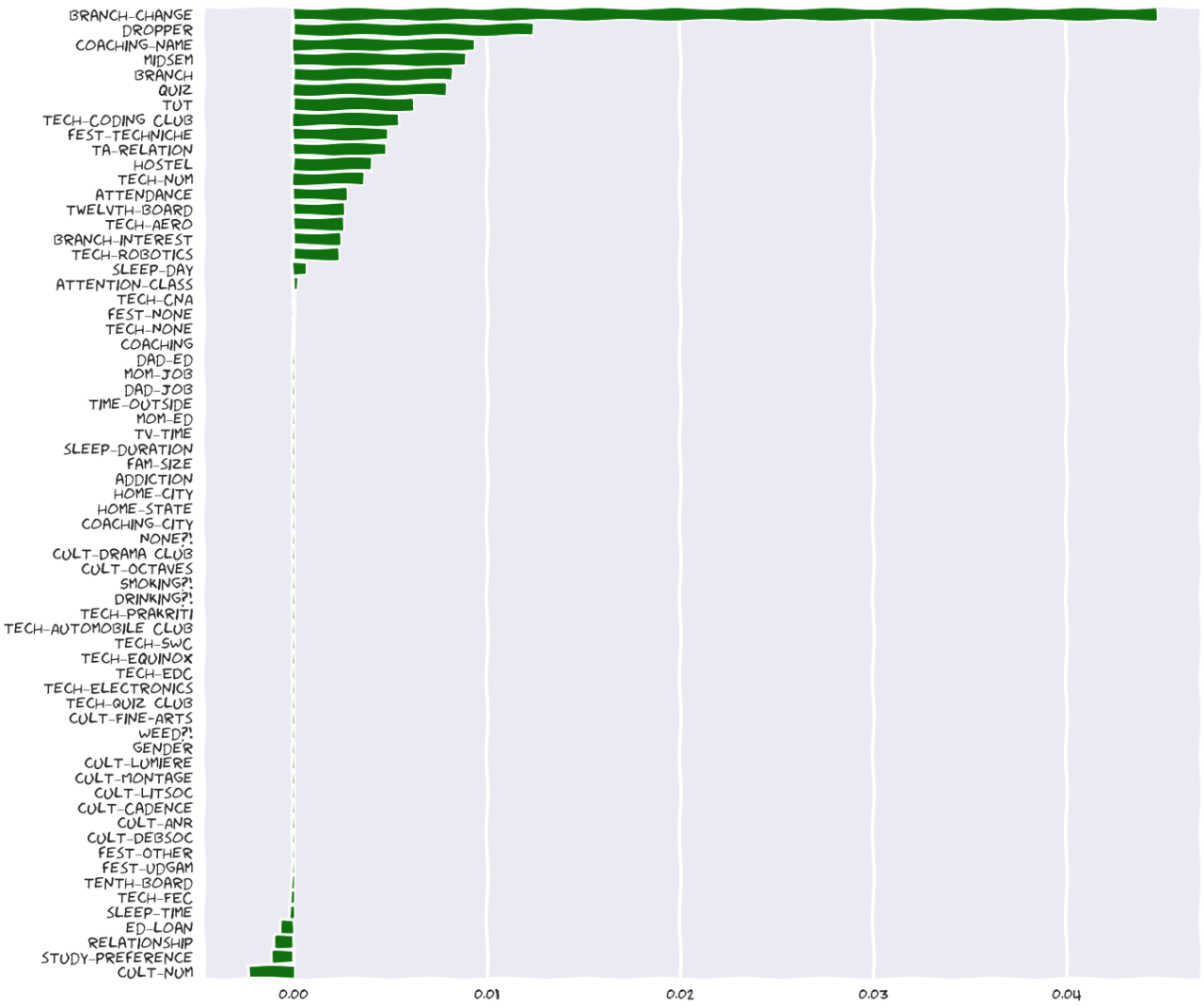
But the results are worth the wait -



Negative differences were what led me to add 20 iterations of the random loop because otherwise it could be just a bad coincidence. But after 20 iterations, I don't think it's a coincidence anymore.

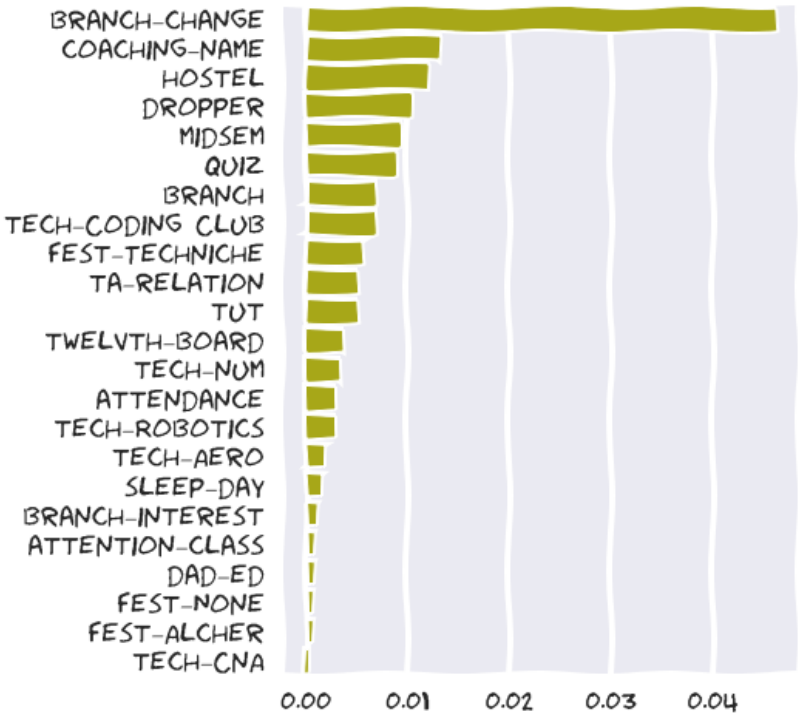
I decided, rather obviously to drop the *Study Time* column. The RMSE went down from 1.32027 to 1.29927. After dropping *Core* it further went down to 1.29457. After *Endsems* – 1.29187

I plott the Feature Importance again -



Post this, I dropped all the columns below *Coaching* and bam, the Validation RMSE fell down to 1.27002. Having said this, the negative columns seem to depend on the combination chosen for the train and validation set. But the positive ones seem to be more or less the same for all of them – Branch Change, Branch, Coaching, Intra Hostel, Hostel, *Interest Columns,* Mom-ed seem to be important always. While the negative columns show which are the columns more likely to mislead the model. It's also worth noticing the columns that often don't make a difference nor does the model use them, most notably of them *Addiction* and *Gender.*

The final model's Feature Importance looked like –



Conclusion –

I hoped you had as much fun going through this analysis as much as I had creating it. I apologize for the 25 pages, but then the insights were key, and I hope it has given you a proper glimpse into the data.

On an ending note, I'm going to leave this graph (un-analyzed of, course) here –