TAKE-HOME CHALLENGE CHEAT SHEET

EMMA ding

EMMA ding

emmading.com

info@emmading.com

Take-Home Challenge Cheat Sheet

👀 What Are Interviewers Looking For?

When completing a take-home assignment, it's important to remember that it's similar to an interview. The audience needs to evaluate your qualifications quickly, so it's crucial to keep your response **concise** and **present the most important information first**.

> 🔑 Interviewers want to assess your ability to generate insights quickly.

Here are some basic tips to keep in mind for take-home challenges.

- Keep it short and **concise**, and make it **easy to understand.**
  - Not a place to showcase your theoretical knowledge.
  - Don’t over-complicate it. Make sure the overall message is clear.
- Data visualizations are important. Use **more visuals and less text** to keep the reviewer engaged.

📚 Categories of Take-Home Challenges

Here are some examples of different types of take-home challenges from different companies:

1. **Analytics**
    - Spotify: Analyze the factors that contribute to the success of a playlist.
    - Wayfair: Evaluate website acquisition strategies and identify areas for improvement in the user funnel.
    - Duolingo: Segment users based on their characteristics and provide personalized recommendations.
2. **Metric and A/B Testing Design**
    - Lyft: Conduct a churn analysis to understand and reduce customer attrition.
    - Quora: Analyze user engagement metrics and propose strategies to enhance user participation.
    - Robinhood: Design a push notification system to improve user engagement.
    - Hinge: Develop strategies to increase the retention rate for all users.
    - TikTok: Test the effectiveness of a new repost button feature similar to retweets.
3. **Machine Learning**
    - Doordash: Build a model to predict the total delivery duration in seconds based on relevant variables.
    - Instacart: Develop an algorithm to estimate the number of full stops and turns in a trip.
    - Opendoor: Create a pricing model using the k-nearest neighbors (k-NN) algorithm.
4. **Domain-specific**
    - Upstart: Estimate the percentage of loans that will be charged off by the end of their 3-year terms.

5️⃣ Five-Step Framework

> 🌻 Data challenges may require you to answer specific questions or allow you to present any insights you have in a more open-ended format. Regardless of the format, you can **organize your results** into five sections and present them in a Google Doc, Google Slides, or Jupyter Notebook.
>
> Depending on the requirements of the data challenge, you may not need both Experiment Design and Modeling. Typically, one of them is sufficient. However, the other three sections are mandatory.

▼ Goal and Assumptions

- Clearly state the **goal** of the data challenge and focus on the business impact.
- State your **assumptions** at the beginning.

▼ EDA and Insights

▼ Data Cleaning

- Never assume that data is perfect. There are always data quality issues, and we need to clean the data before performing any analysis.
    - Outlier removal
    - Dealing with missing values
    - Dealing with erroneous values

> 💡 We recommend having a section on data preprocessing at the **beginning** and explaining how you are going to process the data. Then you can use the cleaned data for the rest of your analysis. In this way, your code will be much cleaner.

▼ Variables/Features

Below are some variables you can look into to gather insights:

> ☘️ **Note:** You don't need to present all of them, but rather look at them all and select the ones that convey insights.

1. **Temporal variables**
    - e.g. hour of day and day of the week
2. **Average or median-related**
    - e.g. average revenue per user, median revenue per user
3. **Frequency-related**
    - e.g. repurchase rate, monthly churn rate
4. **Other features**
    - Use bivariable analysis to identify important features.
    - You can also consider using a simple random forest to get feature importance, then investigate important features.
5. **Customer segmentation**
    - We could use tenure, interests (e.g. items bought), and other features to categorize users which may be helpful to generate insights.

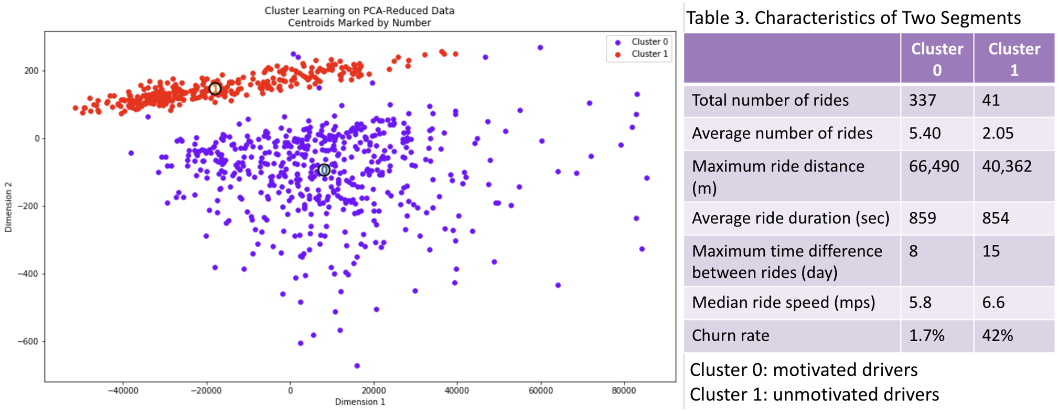


Table 3. Characteristics of Two Segments

	Cluster 0	Cluster 1
Total number of rides	337	41
Average number of rides	5.40	2.05
Maximum ride distance (m)	66,490	40,362
Average ride duration (sec)	859	854
Maximum time difference between rides (day)	8	15
Median ride speed (mps)	5.8	6.6
Churn rate	1.7%	42%

Cluster 0: motivated drivers
Cluster 1: unmotivated drivers

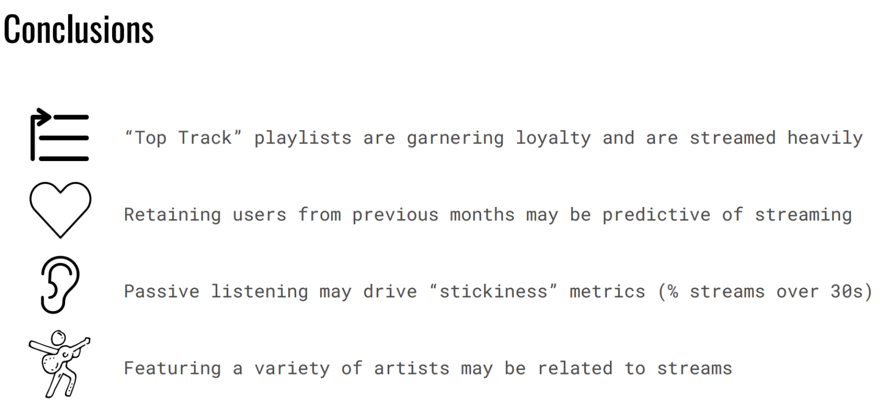
▼ Structure Your Insights

When analyzing data and answering specific questions on the data challenge, it's important to summarize and highlight your findings at the beginning of the section. This makes it easier for the reviewer to quickly understand your insights.

For example, if you're exploring behavioral or demographic factors that might explain a difference in user purchase behavior, you could examine factors such as gender, platform, and education. The summary might then look like this:

> The following factors can explain the difference:
> Gender: Females prefer ... than ...
> Platform: IOS ...
> Education: People with high education ...

Another example of summarizing your insights:



Conclusions

“Top Track” playlists are garnering loyalty and are streamed heavily

Retaining users from previous months may be predictive of streaming

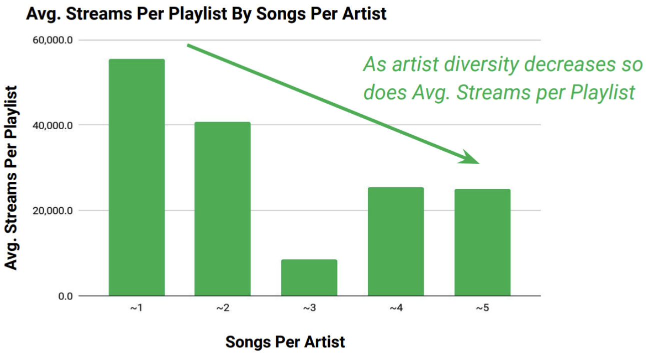
Passive listening may drive “stickiness” metrics (% streams over 30s)

Featuring a variety of artists may be related to streams

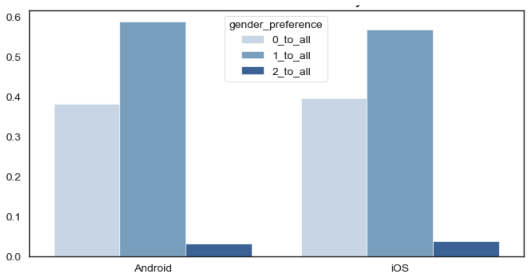
▼ Data Visualization

When presenting visuals, it is important to ensure that they are easy to understand. Each chart or graph should have a **clear legend**, and there should only be one thing depicted in each chart or graph.

✅ **Good Example**



❌ **Bad Example**



▼ Experiment Design

In some take-home challenges, A/B testing is required. To successfully tackle the A/B testing-related data challenge, it is important to have a solid grasp of A/B testing and be able to address key aspects of the process. The following components should be covered:

Be sure to cover:

- Metric Selection
- Analyze the treatment effect
- Make launch decision

You can also talk about:

- Pitfalls
  - The difference in treatment effect between new users and existing users indicates the presence of a novelty effect and a primacy effect.
  - Post-test segmentation may lead to Simpson's Paradox

To complete the A/B testing-related data challenge, you need to have a basic understanding of A/B testing. You can obtain this knowledge by going through my Youtube playlist here. The data challenge will require you to discuss various aspects related to A/B testing, and you must cover at least the first three parts.

▼ Modeling

- Begin by selecting a group of model candidates for evaluation.
- Assess the performance of each model through methods like cross-validation and choose the most suitable model.
- After identifying the model, iteratively eliminate irrelevant features to determine the best feature set.
- Once the feature selection process is complete, optimize the model's performance by fine-tuning the hyperparameters.

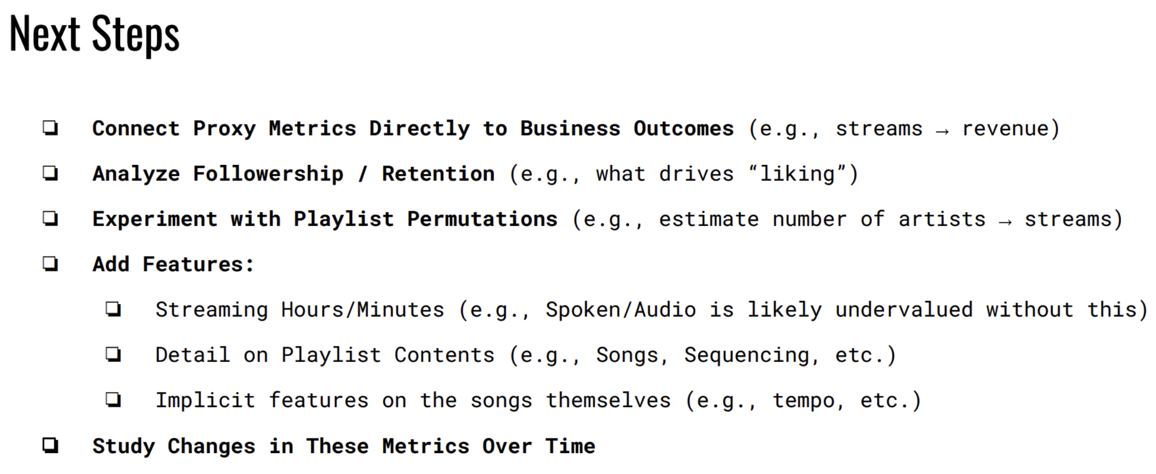
▼ Recommendations and Next Steps

> 💡 It is **always** good to end with some recommendations, suggestions, and next steps.

What can you recommend?

There are various ways to recommend the next steps, especially after analyzing data. Here are some simple examples of some possible recommendations:

- Based on the analysis of product popularity and users' purchase behavior, we could recommend different products to different users, possibly even offering discounts, to retain them.
- Which users (e.g. new/old, location, previous purchase behavior, etc.) prefer to buy which types of items? Can we recommend similar items to similar users to retain them? For instance, if new users tend to buy pillows, we could recommend pillows to new users.
- We could examine the revenue distribution among different users to see which users generate the most revenue. We could even offer discounts based on their interests. Compared to the customer's lifetime value (LTV), offering a discount or free item can be a no-brainer if the LTV is much larger than the discount or free item.



Next Steps

❏ **Connect Proxy Metrics Directly to Business Outcomes** (e.g., streams → revenue)

❏ **Analyze Followership / Retention** (e.g., what drives “liking”)

❏ **Experiment with Playlist Permutations** (e.g., estimate number of artists → streams)

❏ **Add Features:**

  ❏ Streaming Hours/Minutes (e.g., Spoken/Audio is likely undervalued without this)

  ❏ Detail on Playlist Contents (e.g., Songs, Sequencing, etc.)

  ❏ Implicit features on the songs themselves (e.g., tempo, etc.)

❏ **Study Changes in These Metrics Over Time**

> 👌 Depending on the requirement of the data challenge, you may not need both Experiment Design and Modeling. Typically, either is enough. But the other three sections are required.

✨ Pro Tips

Table of content

It's always helpful to have a table of contents at the beginning of a document so that reviewers can navigate it easily.

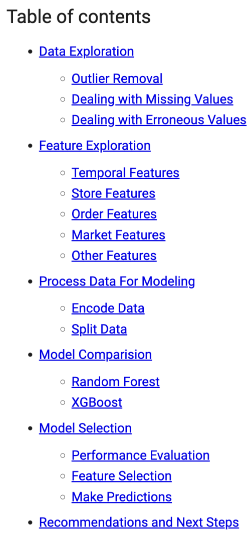


Table of contents

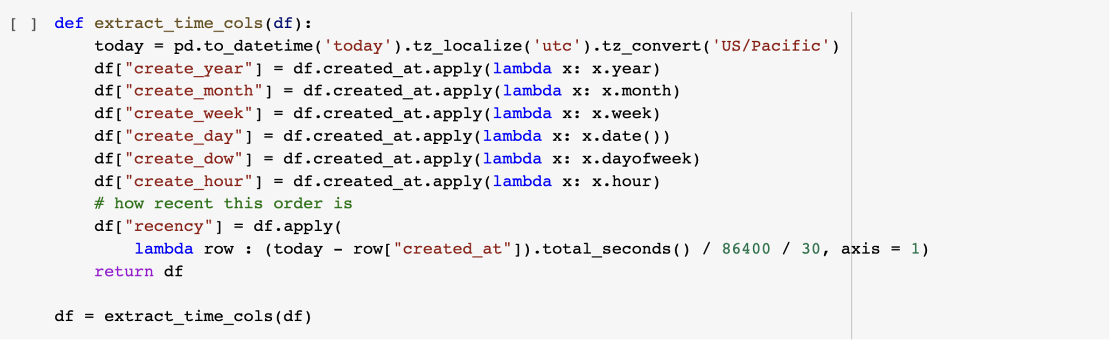
- Data Exploration
  - Outlier Removal
  - Dealing with Missing Values
  - Dealing with Erroneous Values
- Feature Exploration
  - Temporal Features
  - Store Features
  - Order Features
  - Market Features
  - Other Features
- Process Data For Modeling
  - Encode Data
  - Split Data
- Model Comparision
  - Random Forest
  - XGBoost
- Model Selection
  - Performance Evaluation
  - Feature Selection
  - Make Predictions
- Recommendations and Next Steps

Code requirement

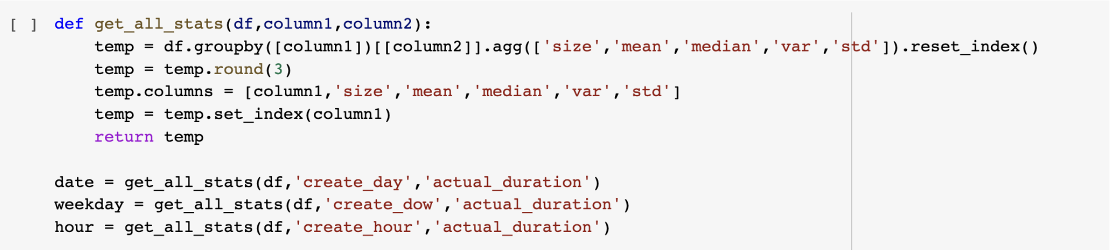
> ☘️ Some data challenges require you to submit your code, while others do not. Even if code submission is not requested, it can be beneficial to include snippets of your code.

- Add comments on the code explaining your approach.
- Clarity and conciseness of code
  - To make the code concise and reusable, I recommend having a section on data preprocessing at the **beginning** and explaining how you are going to process the data. Then you can use the cleaned data for the rest of your analysis. In this way, your code will be much cleaner.

  ✅ Good example



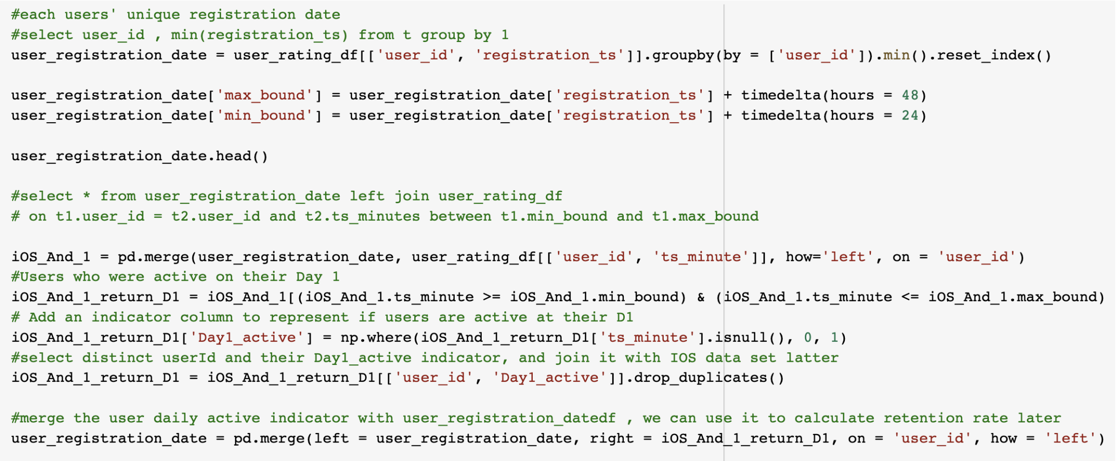
```
[ ] def extract_time_cols(df):
        today = pd.to_datetime('today').tz_localize('utc').tz_convert('US/Pacific')
        df["create_year"] = df.created_at.apply(lambda x: x.year)
        df["create_month"] = df.created_at.apply(lambda x: x.month)
        df["create_week"] = df.created_at.apply(lambda x: x.week)
        df["create_day"] = df.created_at.apply(lambda x: x.date())
        df["create_dow"] = df.created_at.apply(lambda x: x.dayofweek)
        df["create_hour"] = df.created_at.apply(lambda x: x.hour)
        # how recent this order is
        df["recency"] = df.apply(
            lambda row : (today - row["created_at"]).total_seconds() / 86400 / 30, axis = 1)
        return df

    df = extract_time_cols(df)
```



```
[ ] def get_all_stats(df,column1,column2):
        temp = df.groupby([column1])[[column2]].agg(['size','mean','median','var','std']).reset_index()
        temp = temp.round(3)
        temp.columns = [column1,'size','mean','median','var','std']
        temp = temp.set_index(column1)
        return temp

    date = get_all_stats(df,'create_day','actual_duration')
    weekday = get_all_stats(df,'create_dow','actual_duration')
    hour = get_all_stats(df,'create_hour','actual_duration')
```

❌ Bad example:

The example below is hard to read and can be improved by using shorter variable names and functions.

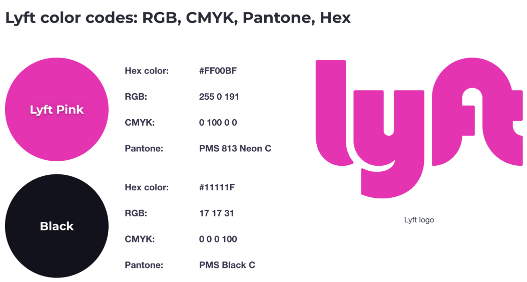


```
#each users' unique registration date
#select user_id , min(registration_ts) from t group by 1
user_registration_date = user_rating_df[['user_id', 'registration_ts']].groupby(by = ['user_id']).min().reset_index()

user_registration_date['max_bound'] = user_registration_date['registration_ts'] + timedelta(hours = 48)
user_registration_date['min_bound'] = user_registration_date['registration_ts'] + timedelta(hours = 24)

user_registration_date.head()

#select * from user_registration_date left join user_rating_df
# on t1.user_id = t2.user_id and t2.ts_minutes between t1.min_bound and t1.max_bound

iOS_And_1 = pd.merge(user_registration_date, user_rating_df[['user_id', 'ts_minute']], how='left', on = 'user_id')
#Users who were active on their Day 1
iOS_And_1_return_D1 = iOS_And_1[(iOS_And_1.ts_minute >= iOS_And_1.min_bound) & (iOS_And_1.ts_minute <= iOS_And_1.max_bound)
# Add an indicator column to represent if users are active at their D1
iOS_And_1_return_D1['Day1_active'] = np.where(iOS_And_1_return_D1['ts_minute'].isnull(), 0, 1)
#select distinct userId and their Day1_active indicator, and join it with IOS data set latter
iOS_And_1_return_D1 = iOS_And_1_return_D1[['user_id', 'Day1_active']].drop_duplicates()

#merge the user daily active indicator with user_registration_datedf , we can use it to calculate retention rate later
user_registration_date = pd.merge(left = user_registration_date, right = iOS_And_1_return_D1, on = 'user_id', how = 'left')
```

▼ Use the brand color

When designing a slide deck, incorporating the brand color demonstrates your awareness of the company brand and passion. It can evoke an emotional connection with the audience by using the brand's unique color scheme, thus improving engagement and retention of the presented information.

> 👉 You can simply search ***[Company] brand color*** on Google to find it.



Lyft color codes: RGB, CMYK, Pantone, Hex

Lyft Pink	
Hex color:	#FF00BF
RGB:	255 0 191
CMYK:	0 100 0 0
Pantone:	PMS 813 Neon C

Black	
Hex color:	#11111F
RGB:	17 17 31
CMYK:	0 0 0 100
Pantone:	PMS Black C

▼ More visualizations and less written text

- Keep in mind that the reviewer may not have time to review everything. To help them, make sections shorter by using headings and including only 2 or 3 bullet points to describe the purpose of each section.
- Use plots and diagrams to visualize your ideas.

▼ Add engaging icons, diagrams, and graphics

You can use these websites to find things to make your presentation more engaging.

- https://www.flaticon.com/
- https://www.freepik.com/

▼ Interesting findings

- If you come across interesting insights in the data, present them at the end of the slide deck or in the appendix (if you're using Google Docs), even if they're not directly related to the questions.
    - For example, when examining the product popularity of an e-commerce site, you may notice that area rugs generate the most revenue. However, it is unclear whether this is due to a high volume of sales or a higher average price per rug.
- When presenting onsite, you can discuss this topic to make your presentation more engaging.

🔗 Useful Resources

- Reading a File
    - https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html
- Data Cleaning
    - https://realpython.com/python-data-cleaning-numpy-pandas/
- Visualization
    - https://matplotlib.org/stable/gallery/index
- A/B Testing

    - https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html
- Machine Learning
    - https://scikit-learn.org/stable/user_guide.html