SmartMatch 0.2 implementation

PRIMARY OBJECTIVE

Provide suggestion of top matches to matchmakers.

TECHNOLOGIES USED

- Google Sheets
    - Contains sample data that resembles our Postgres
    - https://docs.google.com/a/seekpanda.com/spreadsheets/d/1_G-YDPJBpLBOzCMxcYSVradpXjL35SysgzZJHldI9gM/edit?usp=sharing
- Sample job creation form
    - Mimics our live job creation flow
    - Note that submissions sent to this job creation form will automatically show up in the Google Sheets link provided
    - **These submissions may take up to 15 minutes to show up in Google Sheets**
    - https://seekpanda.typeform.com/to/GhsdMR
- Slack
    - Used to communicate the results of the match
- Github
    - To host the repo containing your source code
- AWS / Heroku
    - To deploy your app whose purpose is reading new jobs and pushing the results of the matchmaking to Slack

SCOPE & DELIVERABLES

Phase 1: Data engineering

- □ Review the Google Sheets doc and decide if you want to migrate it to your own database or if you want to keep the data on Google Sheets or put it on a Postgres instance
- □ Write a script `jobs` that migrates jobs data currently in `job_details` into the relevant fields which are currently blank
- □ Write a script for `pandas` that migrates the price data currently in `prices` into the relevant fields which are currently blank

Phase 2: Write a script to generate sorted matches for one type of service only

- □ Identify jobs having `service_type` = interpreting (in-person)
- □ Introduce a temporary variable called `match_percentage`
- □ For each job, calculate `match_percentage` (detail below) for all pandas
- □ Return a sorted array (in descending order of match percentage) of the `name` and `target_price` fields of the top matching pandas

Phase 3: Set up a webhook to push the results to Slack

- □ Use the job creation form to create sample jobs, and send those to Slack
- □ Inform us about the top 3 matches on the #sandbox channel: “Hey guys! For job www.seekpanda.com/#/jobs/`jobid`, please invite (in order) these people: `name ($target_price)`, `name ($target_price)`, `name ($target_price)`”. Pay attention to the formatting such that the result looks like (for example) “Hey guys! For job www.seekpanda.com/#/jobs/653, please invite (in order) these people: Rachel Huang ($250), Skye Wu ($125), Kate Xiao ($90)”

Match percentage

	Component	Definition	Possible values
1	location_match	Boolean variable to see if locations match (if jobs.location = pandas.basecity)	1 if TRUE 0 if FALSE
2	industry_weight	How much does a customer care about industry?	0.8 if jobs.industry_depth = ”Critical. Being able to understand terminology is a major determinant of success” 0.5 if ”Important. I’m willing to tolerate some amount of unfamiliarity.” 0.3 for all other cases
3	industry_performance	How well does this panda know an industry?	1 if jobs.industry appears in pandas.specialities 0.5 if jobs.industry appears in pandas.capabilities 0.1 for all other cases

```
match_percentage = industry_weight * industry_performance * location_match
```

Slack webhook

Webhook URL: `https://hooks.slack.com/services/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`
This will automatically post to a channel called #sandbox in SeekPanda’s Slack