STACKOVERFLOW DATA ANALYSIS

PROBLEM DEFINITION

Stackoverflow is a question-and-answer forum for technical enthusiasts where users are free to ask and answer questions. Several attributes are pinned around posts and users such as votes, tags, reputation, badges, etc., Although the forum is filled with professional experts the average response time for each question is around 4 days. In some cases, this delay is when the right person doesn't happen to come across a user's question making it hard to tap the full potential of an expert. If this turnaround time is reduced by directing the posts to the right person at the right time, then the questions will be answered quickly, thereby decreasing the time spent on solving issues.

The main objective of the project is to find insights that help reduce the response time for a question, thereby increasing the productivity of tech wizards. In addition, we explore other trends that can help with the same.

CHALLENGES

Huge Size of Data and Data Format

The data retrieved from stack overflow dump was massive in volume (~ 110+ GBs) and the file format was XML. Hence, we used a data bricks package coupled with spark to convert the xml data into spark data frame and process the entire operation.

Cost of computation in AWS

The stack overflow dataset was extremely huge and the pricing model of charge per usage on the AWS platform seemed overwhelming. The Google Cloud Platform provides an initial credit of $CAD 370 which is more than sufficient for our computational and storage needs.

Prediction Challenges

There was no straightforward dataset for training the models. A cross join was used to create positive and negative 'answered' tuples of users and posts to obtain features that help determine if a question is answered or not. The features were large as we must consider details about the question asked like tags and title and that requires a model like bag-of-words.
When predicting if the user would answer questions of a particular class, we noticed there was severe imbalance towards the not answered class. This made the model predict not answered to every input. To restrict the feature size, we restricted the final number of the bag of words features and under sampling the 'not answered' label helped reduce the model's bias towards it.

METHODOLOGY

With the help of various tools and technologies, we implemented the below pipeline to solve the problem and overcome the challenges.

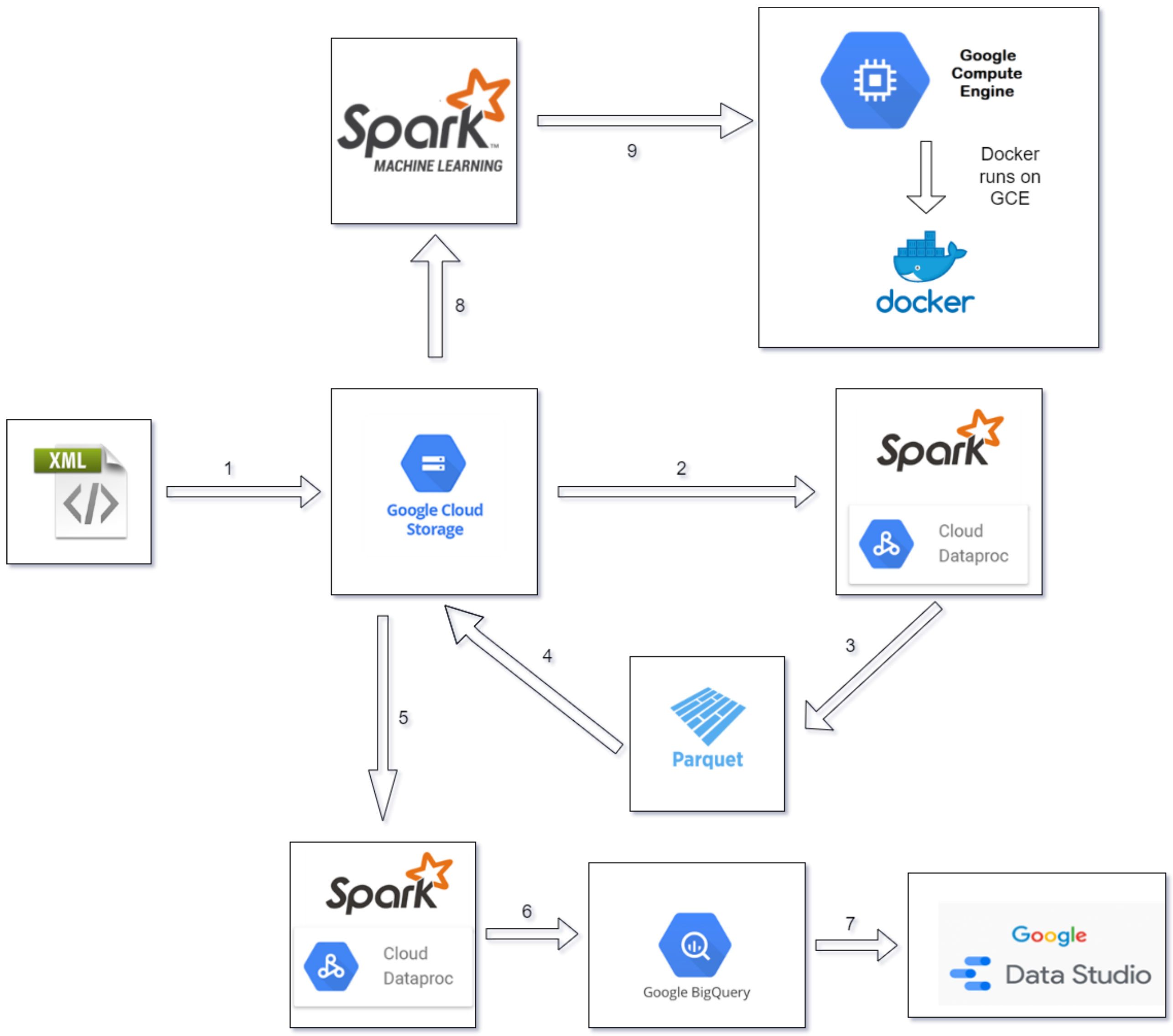


Fig: Technical flow diagram

TOOLS OVERVIEW

PySpark – Spark is an API to process large scale of data with extensive parallelism on a distributed cluster. Spark is used to read the large size data and convert it into useable data frames. It is also used to perform meaningful aggregations to help achieve our insights. The pyspark ML package was extensively used to extract features and for the actual training of the model.

Google Cloud Dataproc – Dataproc is an opensource cloud-based distributed processing platform. It provides user controllable on-demand clusters that can scale from few nodes to few hundreds. Users are payable only for the resources used for computation.

Google Big Query – Big query is a serverless data warehouse that empowers the analysis of big data by performing powerful SQL operations. In addition, big query charges only for operations performed on the data unlike other options like Big Table which charges for the duration of running.

Google Data Studio – Data studio is an online platform to visualize data via informative and interactive reports. Data Studio uses the BigQuery BI engine to make fast retrieval of the necessary data.

Streamlit – Streamlit is an open-source python framework to quickly build and deploy machine learning and data science web applications. It helped avoid a lot of code required for the actual UI implementation and focus only on the actual process to be delivered.

Docker – Docker was used as a containerization tool to host the web application on the Compute Engine VM on GCP. Docker helps us avoid the issues of dependencies for running the application. The entire application can be packaged as an image that can be run on any system that has docker engine.

STAGES

Data Collection

Stack Exchange is a network of question-and-answer websites on topics in various fields. It releases data dumps of all its publicly available content via archive.org and is periodically updated. The stack overflow data till September 2021 was available as XML format and downloaded as a compressed file(.zip). The downloaded data set includes Users, Posts, Votes, Comments, Badges and Tags. The obtained data was uncompressed and uploaded to the Google Cloud Storage bucket for the next process in the pipeline.

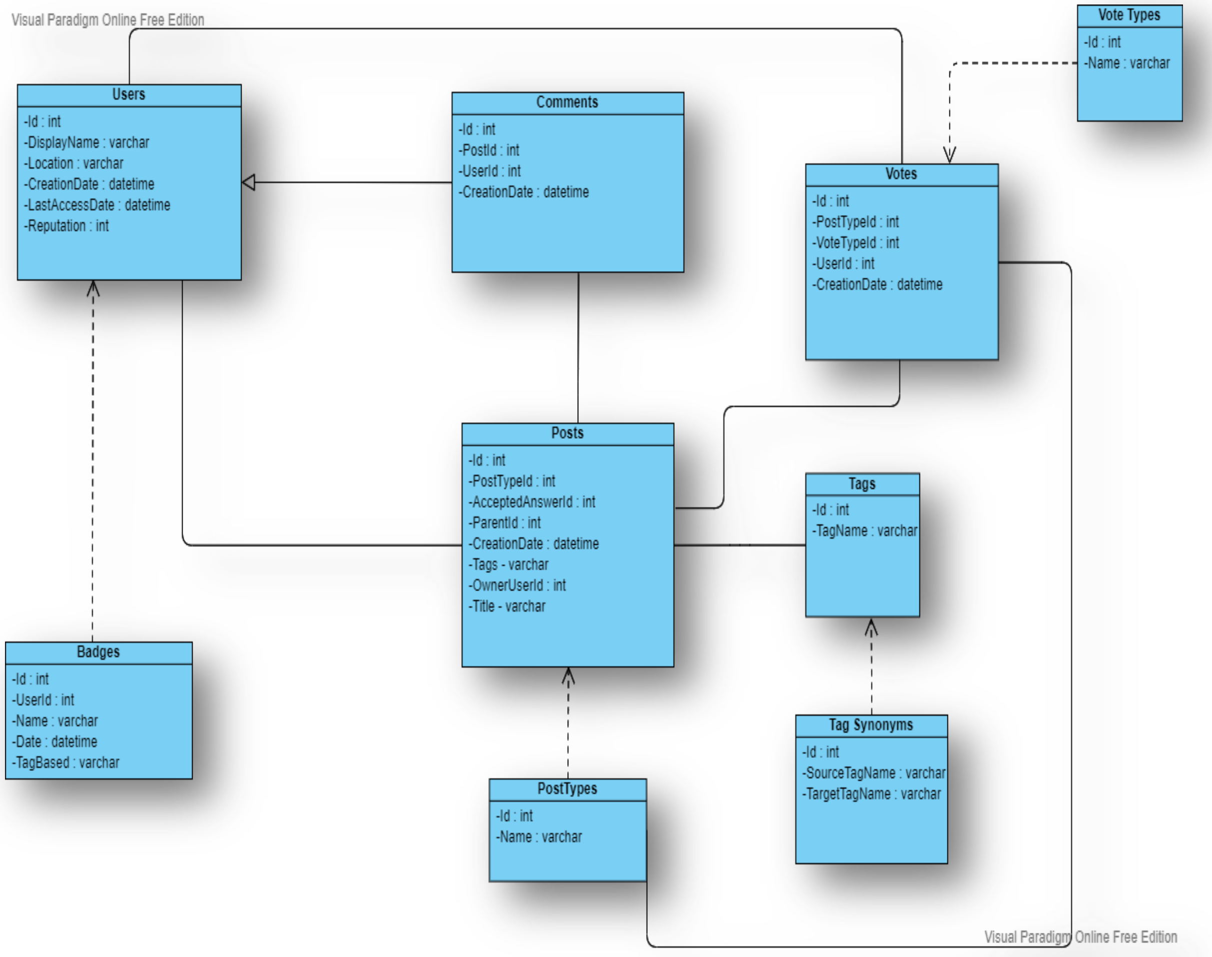


Fig: Schema diagram of the data

ETL

The XML data was read and converted to data frames using the Databricks API. In this stage, the unnecessary columns were dropped, NULL values encountered were cleaned and the raw data was transformed into usable format. The Transformed data was stored as parquet files back to the storage bucket.

Data Analysis

The pre-processed and cleaned data is analyzed and aggregated using the queries through the Spark Dataframe API in pyspark.sql module based on the different scenarios that are being addressed, and the obtained results are stored in a data warehouse (BigQuery).

Data Storage

Google Big Query is used to store the analyzed results which can be queried efficiently through the BigQuery BI engine later for the visualization process in Data Studio.

Data Visualization

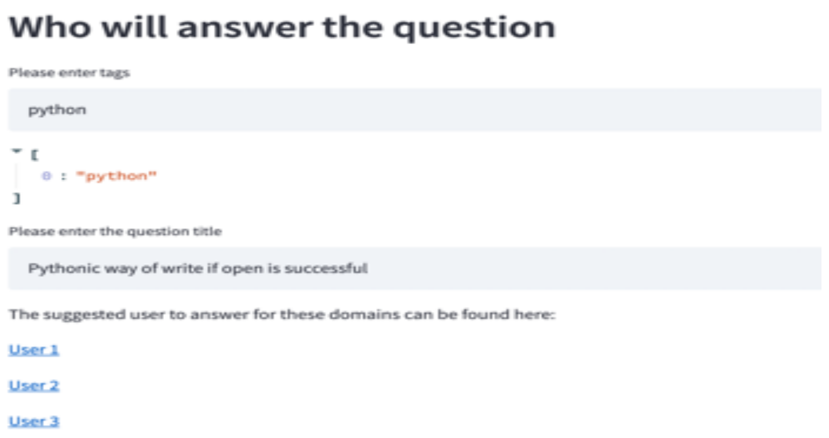
The dataset stored in the BigQuery is visualized using Google Data studio through customizable charts and reports. We have created dashboards that displays the results of analysis which includes:

- Active Users from 2015 based on the number of Q/A posted.
- Top 10 expert users in each tag based on upvotes.
- Average response time per Tag.
- Trend analysis of Badges and Tags.
- Topographical visuals of Tags and Badges.
- Unanswered Question in each Tag.
- Number of duplicate posts per Tag.

Prediction

The features are extracted using PySpark scripts and predictive models are trained on the derived data**.** The model's pipeline and stages are stored in a Google Cloud storage bucket. This is then used by a web application to predict the following:

- If a user will answer a question based on the tags, title and his/her reputation and activity.
- Who are the most likely to answer a question related to a particular domain? The top 3 users are recommended.



Who will answer the question

Please enter tags

python

```
[
  0 : "python"
]
```

Please enter the question title

Pythonic way of write if open is successful

The suggested user to answer for these domains can be found here:

User 1

User 2

User 3

The web application is built in Python using the Streamlit framework and is hosted in a VM on GCP (Compute Engine) as a docker container.

RESULTS

Outcomes

- Successfully implemented the pipeline to process the data and derive actionable insights using data analysis.
- Created a dashboard with informative and interactive charts for viewing the results of data analysis.
- A web application to show which user is likely to answer the question based on the input parameters and to predict if the user answer will be accepted or not.

Learnings from data analysis

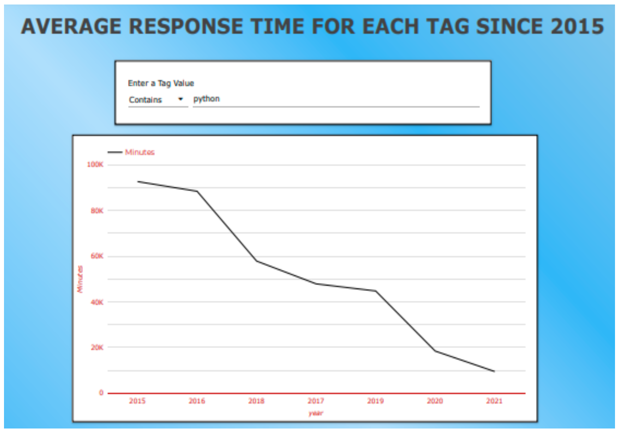
- Python is the most trending language in the years 2020 and 2021 but JavaScript is the all-time highest used language with an overall post of 2265418, followed by Java with 1796493 posts. A new project or team can take advantage of recent popularity of Python in their applications if it suits their needs. Insights like these can help teams starting on new projects by choosing technologies with greater responsiveness and a better community.
- Gordon Linoff is the most active user since 2015 with an overall of 84584 posts and his area of expertise is SQL. Any unanswered SQL question which is pending for a long time can be recommended to Gordon.
- We have identified the top users in each tag based on the number of upvotes they receive for answers and their overall profile reputation. We noticed that there are more that 1 million unanswered questions in the stack overflow forum for the period between 2015-2021. Likewise, all the top users in each tag can be suggested with the unanswered questions relating to their expertise.
- Figured out the location of users who have answered questions on specific tags. This can be potentially used to direct the newly posted questions to specific location depending on the time zone to reduce the turnaround time for answering the questions.
- We extracted a feature set for Badges data which can be used to further improve the process of predicting a user’s response time and acceptance of a user’s answer.

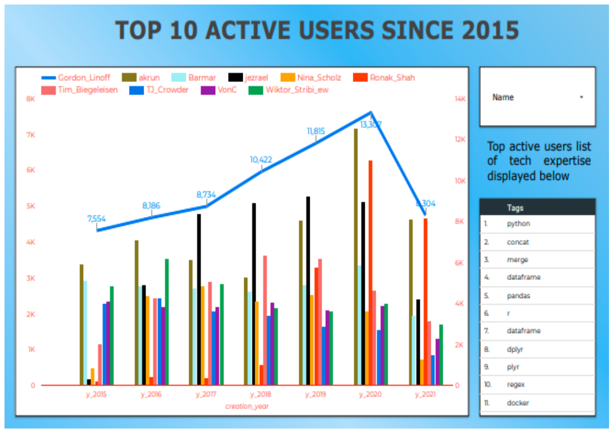
Learnings from implementation

- Efficient transformation of raw data from xml format to parquet which stores data in columnar fashion and preserves the data frame schema through its metadata.
- Creating and maintaining Google Cloud Platform Dataproc clusters to process the huge chunk of data.
- Visualized the analyzed data in the Google Data Studio with several interactive charts to effectively deliver insights that can help with the objective.
- Extracted features and trained predictive machine learning models using Pyspark ML module.

- Build a web application using Streamlit (an opensource python framework) to display the prediction results on a user interface.
- Containerization of web applications using Docker to deliver the entire applications as a package that can be run on any docker engine.

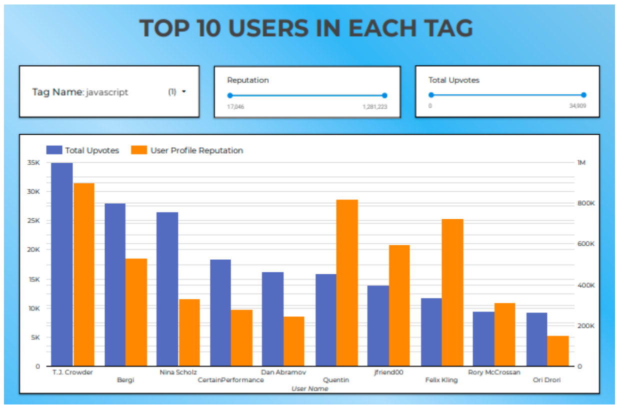
Some of the sample visualizations have been showed below,

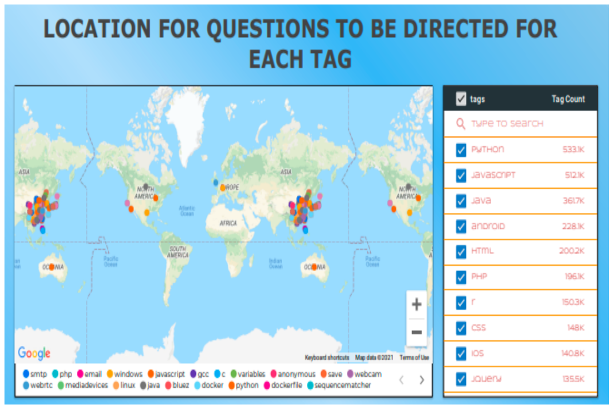




Top active users list of tech expertise displayed below

	Tags
1.	python
2.	concat
3.	merge
4.	dataframe
5.	pandas
6.	r
7.	dataframe
8.	dplyr
9.	plyr
10.	regex
11.	docker





tags	Tag Count
PYTHON	533.1K
JavaScript	512.1K
Java	361.7K
android	228.1K
HTML	200.2K
PHP	196.1K
r	150.3K
CSS	148K
iOS	140.8K
JQuery	135.5K

VISUALIZATION LINKS

Dashboard: https://datastudio.google.com/reporting/f4717110-30f4-4cbb-9df9-e84015941ad9
Web App UI: http://34.145.119.210:8501/

REFERENCES

- https://archive.org/download/stackexchange
- https://console.cloud.google.com/storage/browser/test-732
- https://cloud.google.com/dataproc/docs/quickstarts
- https://cloud.google.com/bigquery
- https://marketingplatform.google.com/about/data-studio/
- https://stackoverflow.com/help/badges

PROJECT SUMMARY

Topic	Marks
Getting the data	1
ETL	2
Problem	3
Algorithmic work	3
Bigness / Parallelization	3
UI	1
Visualization	3
Technologies	4
Total	20

TEAM MEMBERS

Hariish Nandakumar	[301465931]
Niranjan Ramesh	[301437301]
Hiren Bangani	[301442581]
Elavarasan Murthy	[301434125]