GA DSI Project 3: Splitting Clouds

A NLP project to classify text from subreddits r/AWS and r/Azure

Prepared by: Zheng Ting

On: 18 March 2023

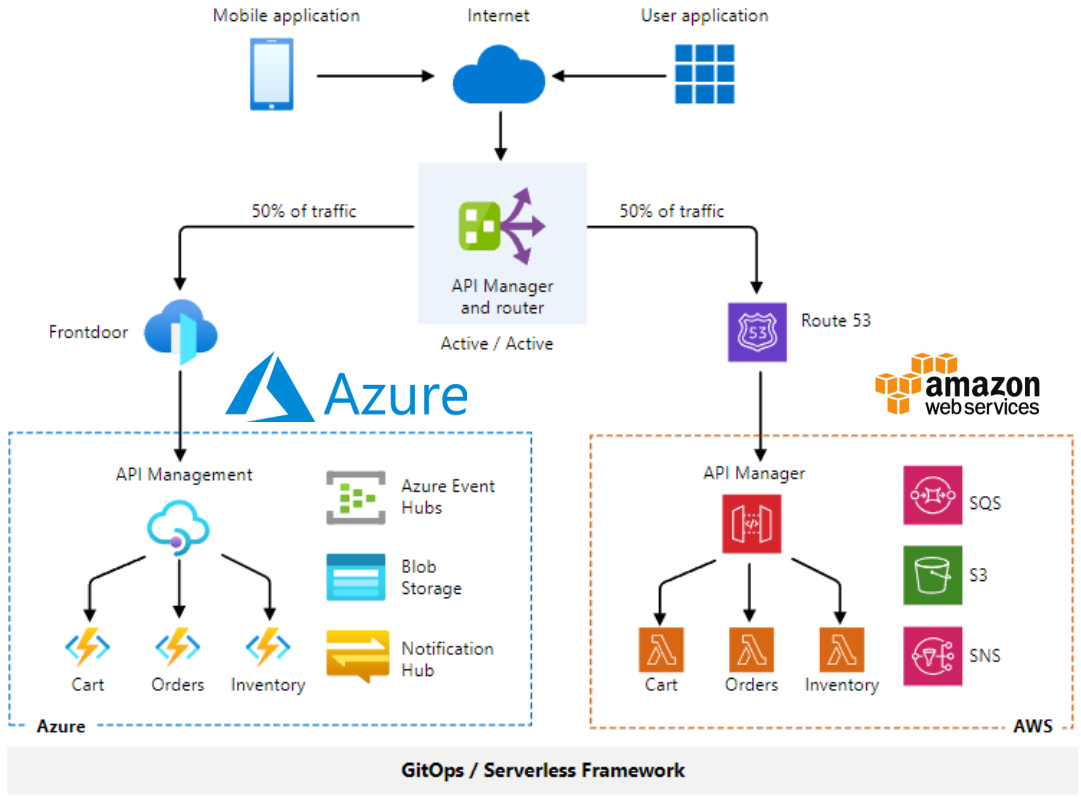
Source: In One Piece Chapter 1027, Luffy and Kaidou split the clouds across Onigashima.

Here, we observe the clash of 2 “emperors” in cloud services, AWS and Azure.

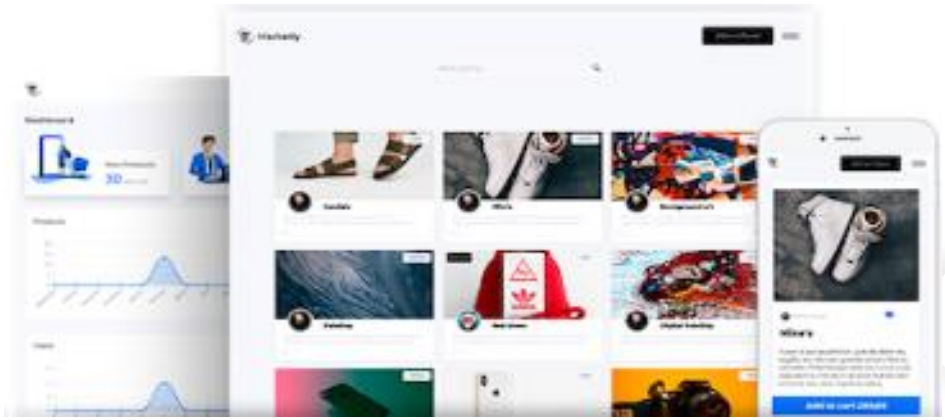
Contents

Scenario

+ Hypothetical E-Commerce company with a multi-cloud architecture.



Source: https://learn.microsoft.com/en-us/azure/architecture/example-scenario/serverless/serverless-multicloud



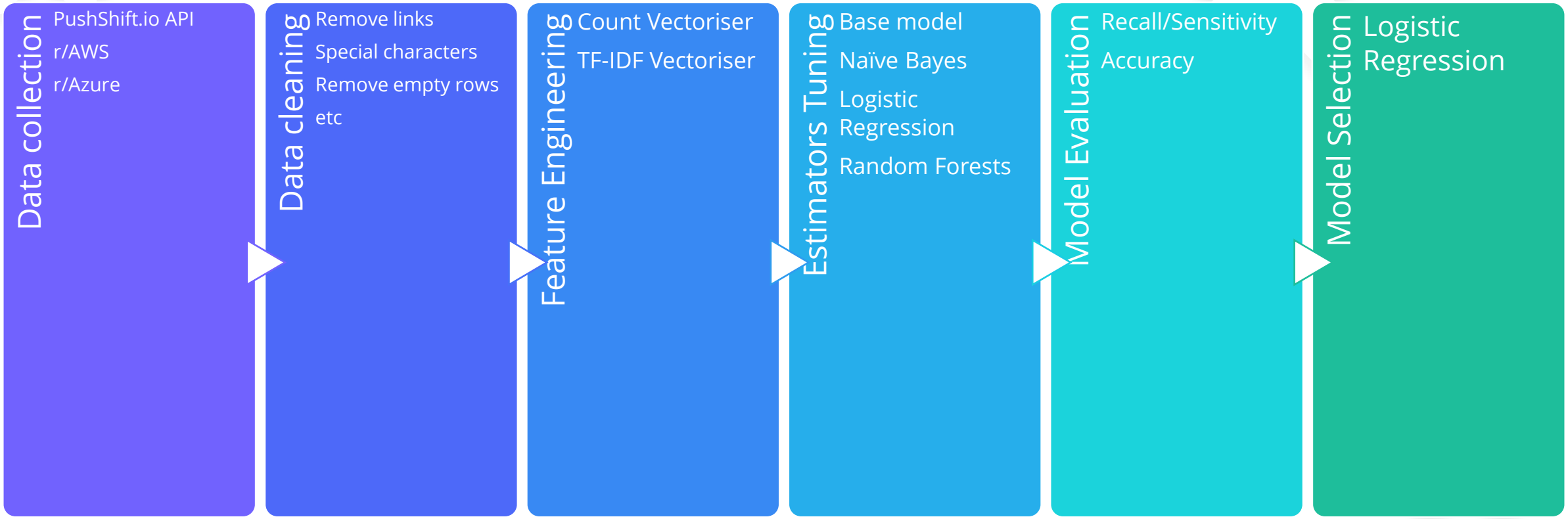
Source: https://zeroqode.com/ecommerce

Problem statement

Source: https://www.techtarget.com/searchitoperations/definition/IT-incident-management

- In this project, we face the cold start issue of creating a machine learning model to classify request tickets, by making use of publicly available data in subreddits.
- This model aims to automate the categorisation of IT requests raised for a company utilizing both AWS and Azure services.
- Reduce time for the relevant DevOps or support team to respond.

Overview


Data collection
PushShift.io API
r/AWS
r/Azure
Data cleaning
Remove links
Special characters
Remove empty rows
etc
Feature Engineering
Count Vectoriser
TF-IDF Vectoriser
Estimators Tuning
Base model
Naïve Bayes
Logistic Regression
Random Forests
Model Evaluation
Recall/Sensitivity
Accuracy
Model Selection
Logistic Regression

Data Collection

+ Web scraping done using pushshift.io API
+ subreddits are r/AWS and r/AZURE
+ API call iterated 5 times to get 5000 posts per subreddit

	subreddit	selftext	author	title	score	num_comments	utc_datetime_str	removed_by
0	aws	[removed]	BrianPRegan	AWS Pricing Add-on for Google Sheets	1	0	2023-03-10 16:28:11	None
1	aws	[removed]	Winter_Sucks_7868	Kansas AWS	1	0	2023-03-10 15:52:40	None
2	aws	We have a site to site VPN between our AWS and...	silicondt	VPN - dynamic - can we put one static also?	1	0	2023-03-10 15:20:34	None

Data Cleaning

- Drop
  - subreddit neither azure nor aws
  - Drop rows with **less than 2 words** in the posts
- **Combine**
  - Title
  - Description
- **Removes** special characters/symbols:
  - Url links
  - [Removed] or [Deleted]
  - newlines
  - quotes
  - bullet points
  - Strikethrough
  - Table
  - Heading
  - Spoilers
  - Code, inline and block

Data Cleaning Example

Title

CloudFormation Condition on CommaDelimitedList

Description

Hi all, I'm working on this cfn template and running into an issue setting up a Conditional for an optional CommaDelimitedList parameter. When I run the following though, the API reports an error: *every Fn::Equals object requires a list of 2 string parameters.* \n\n Parameters:\n SomeParam:\n Description: >-\n (Optional) Comma separated list of things.\n Type...

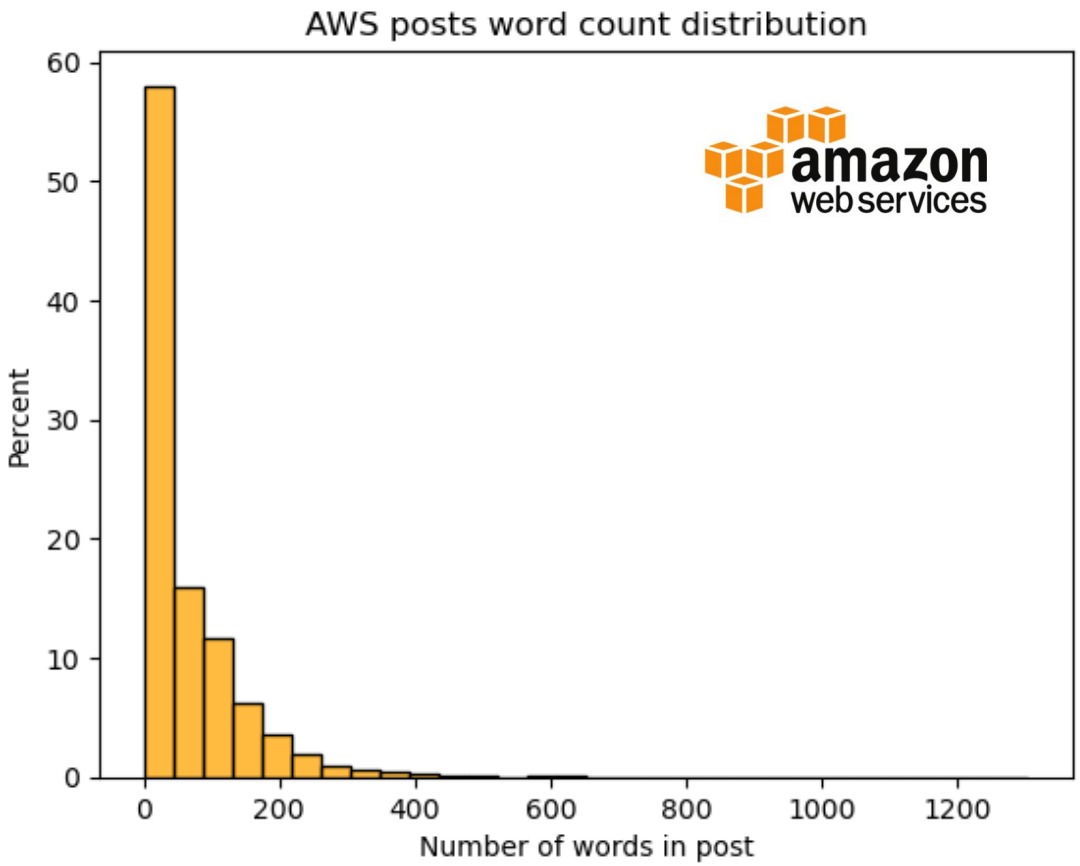
CloudFormation Condition on CommaDelimitedList Hi all, I'm working on this cfn template and running into an issue setting up a Conditional for an optional CommaDelimitedList parameter. When I run the following though, the API reports an error: every Fn::Equals object requires a list of 2 string parameters. Parameters: SomeParam: Description: - (Optional) Comma separated list of things. Type: C...

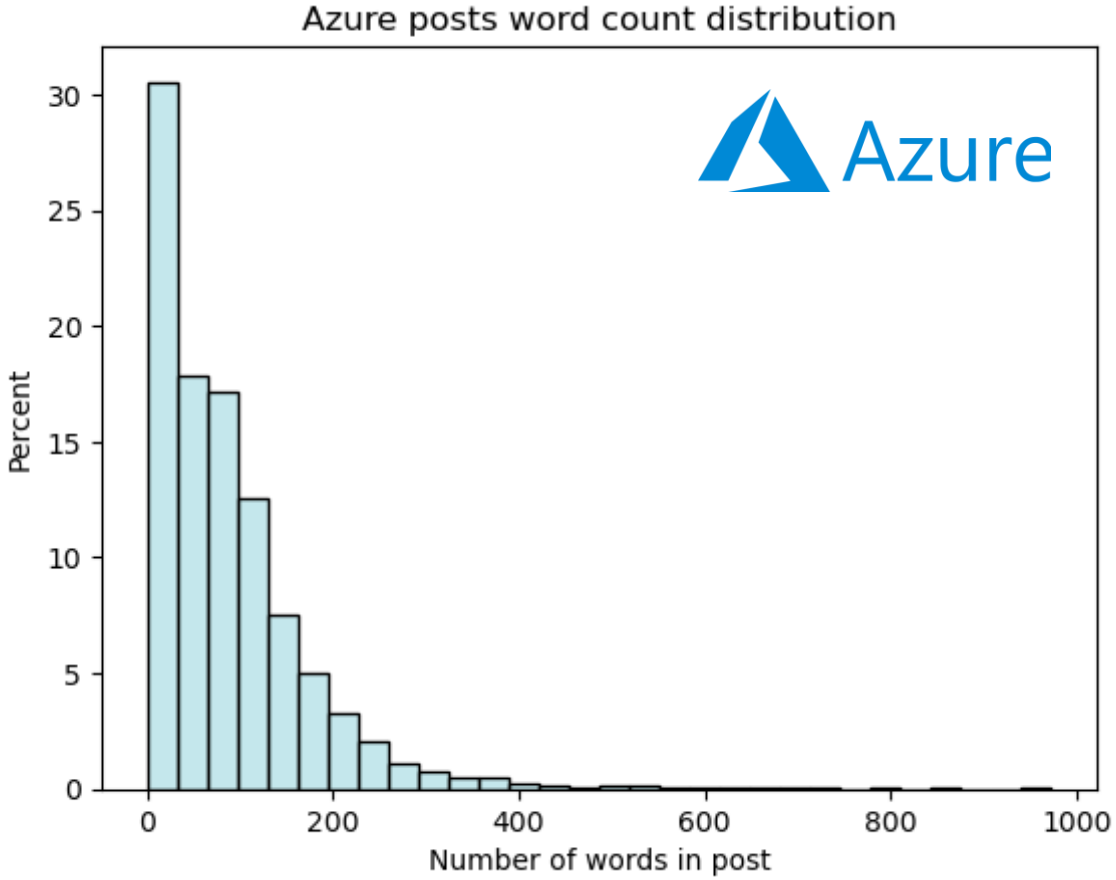
Anyone using Workload Discovery? We're building an alternative, what features do people actually use?

Anyone here using Workload Discovery to build diagrams? We've been building a free alternative and I'd be interested to see what sort of things people actually use workload discovery for.\n\nWhat we're doing: [https://overmind.tech/blog/cloud-infrastructure-diagrams] (https://overmind.tech/blog/cloud-infrastructure-diagrams)

Anyone using Workload Discovery? We're building an alternative, what features do people actually use? Anyone here using Workload Discovery to build diagrams? We've been building a free alternative and I'd be interested to see what sort of things people actually use workload discovery for. What we're doing:

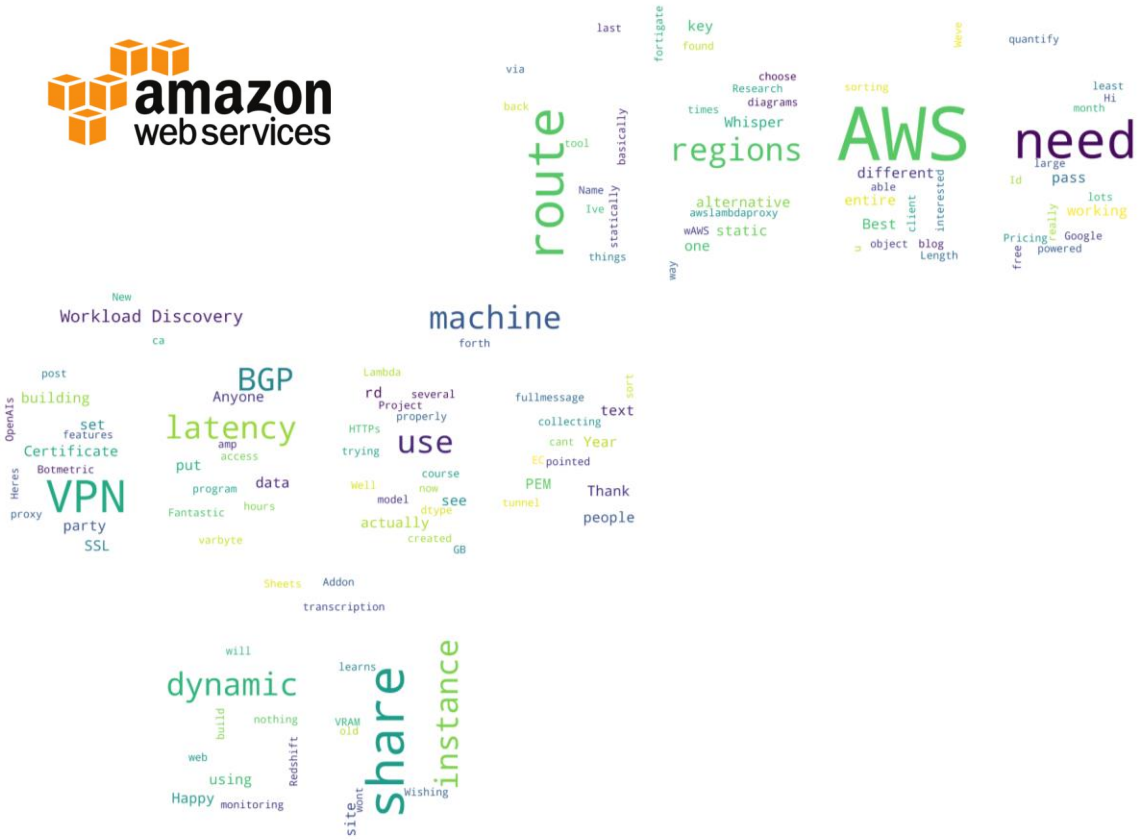
Distribution of word counts for subreddit posts

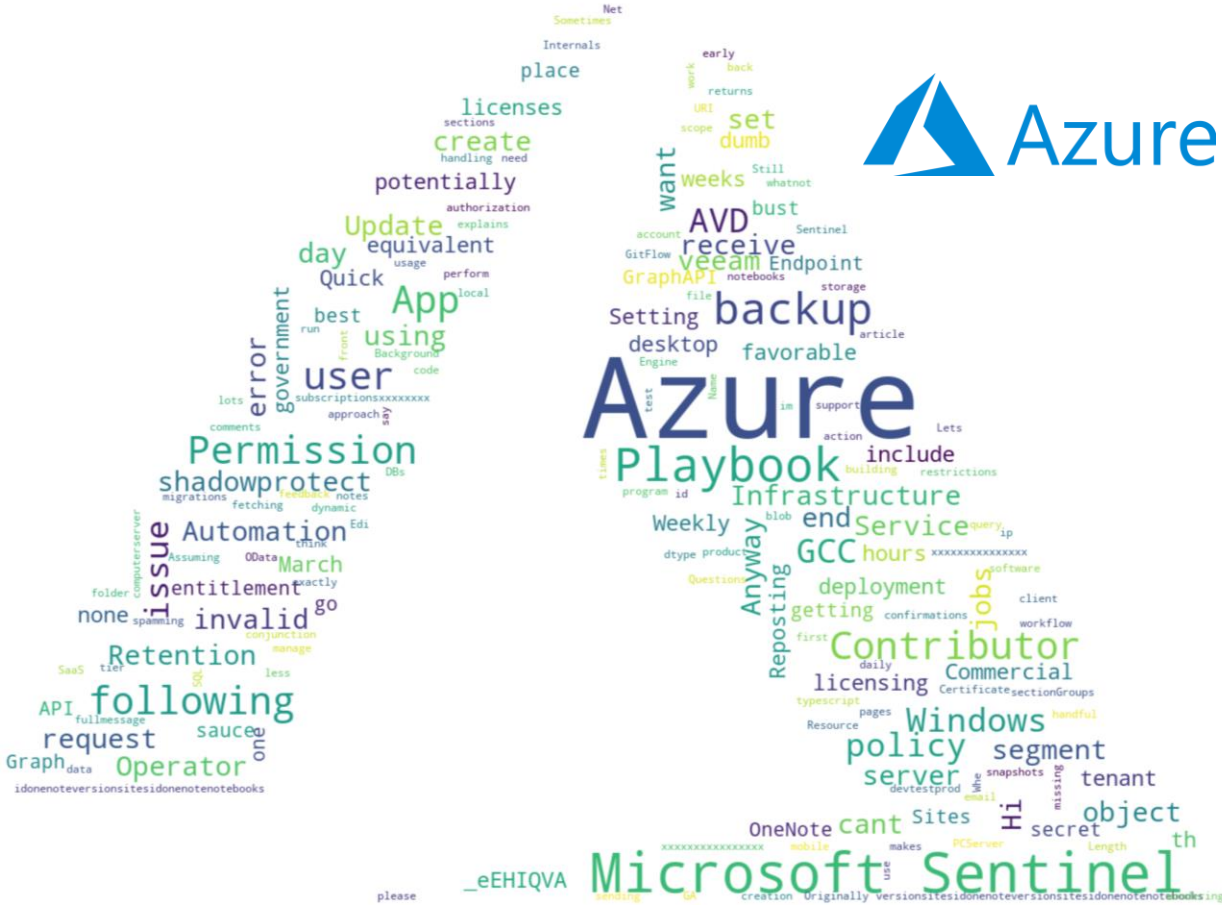




After data cleaning, there are around **60 - 80** words in each post on average.

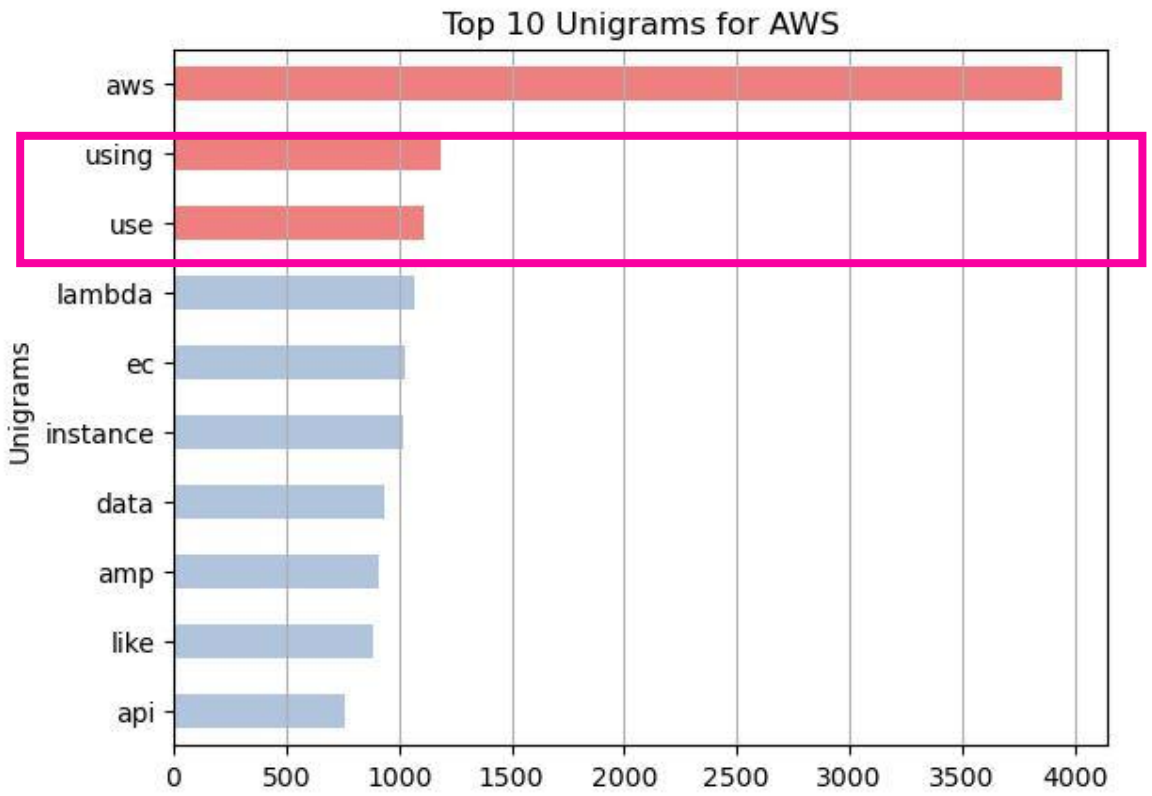
Top words from r/AWS and r/Azure side-by-side



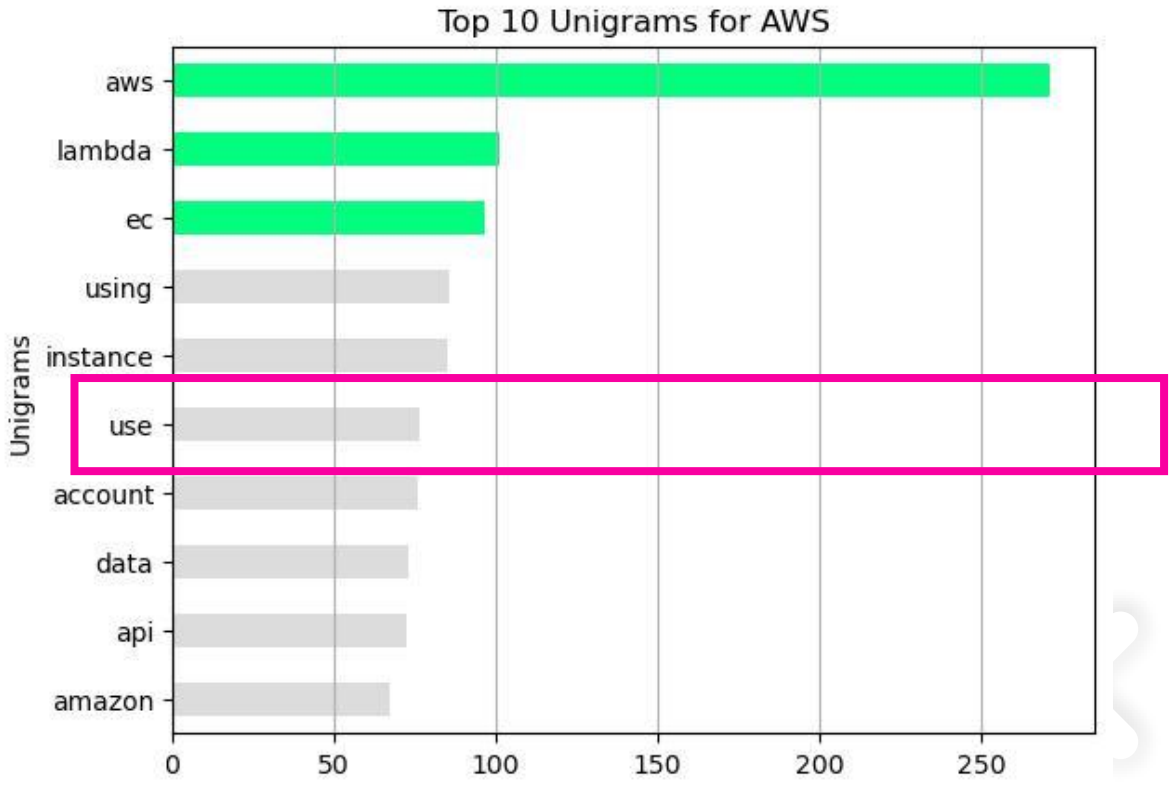


Top Words in r/AWS posts

Count Vectoriser

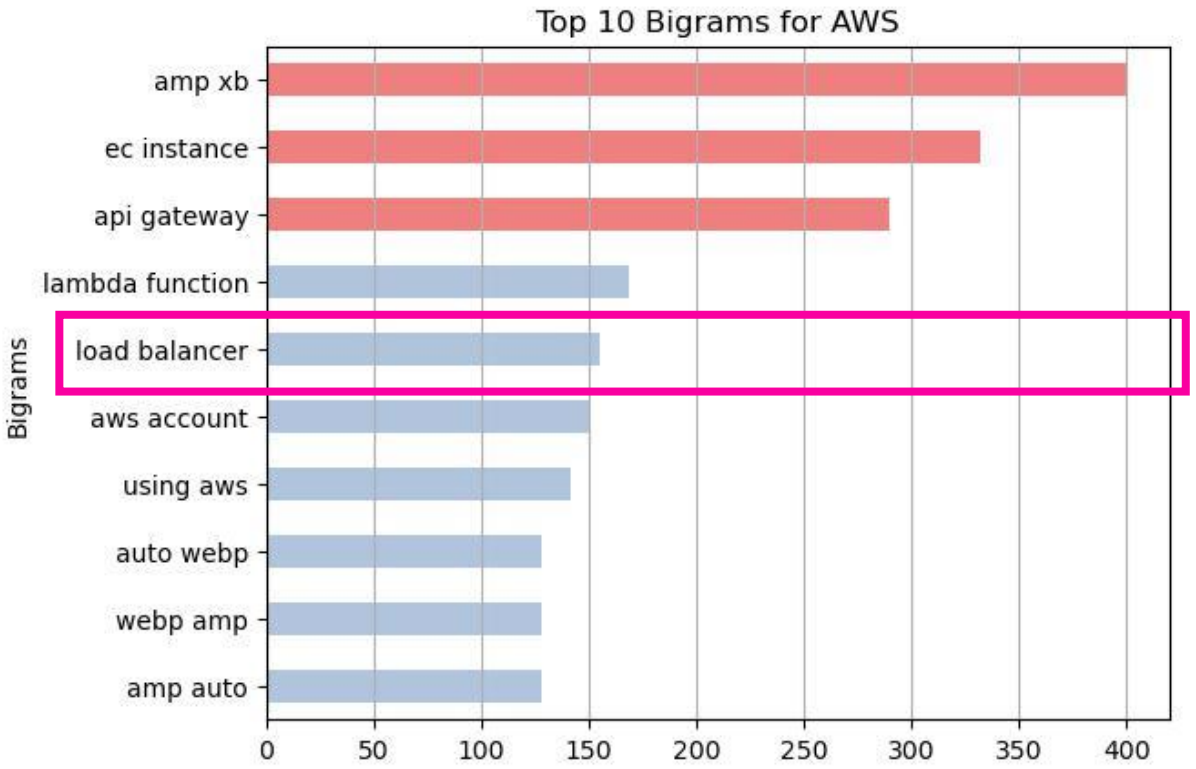


TF-IDF Vectoriser

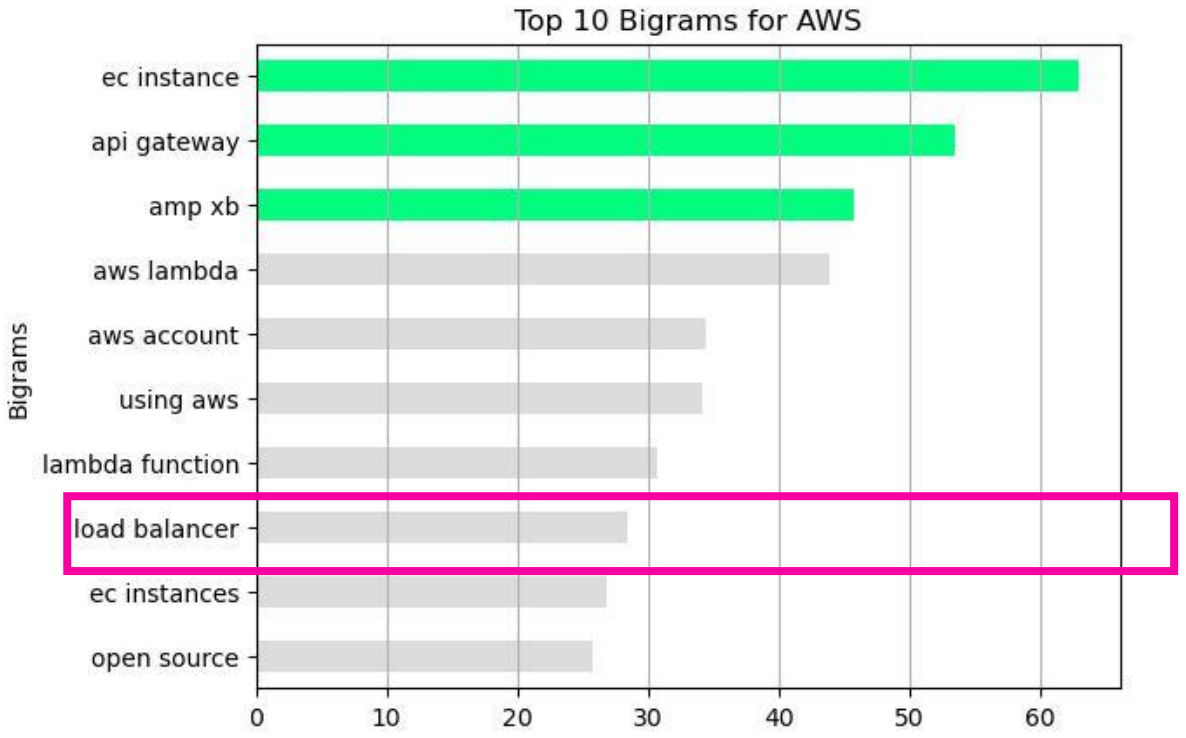


Top Bigrams in n r/AWS posts

Count Vectoriser

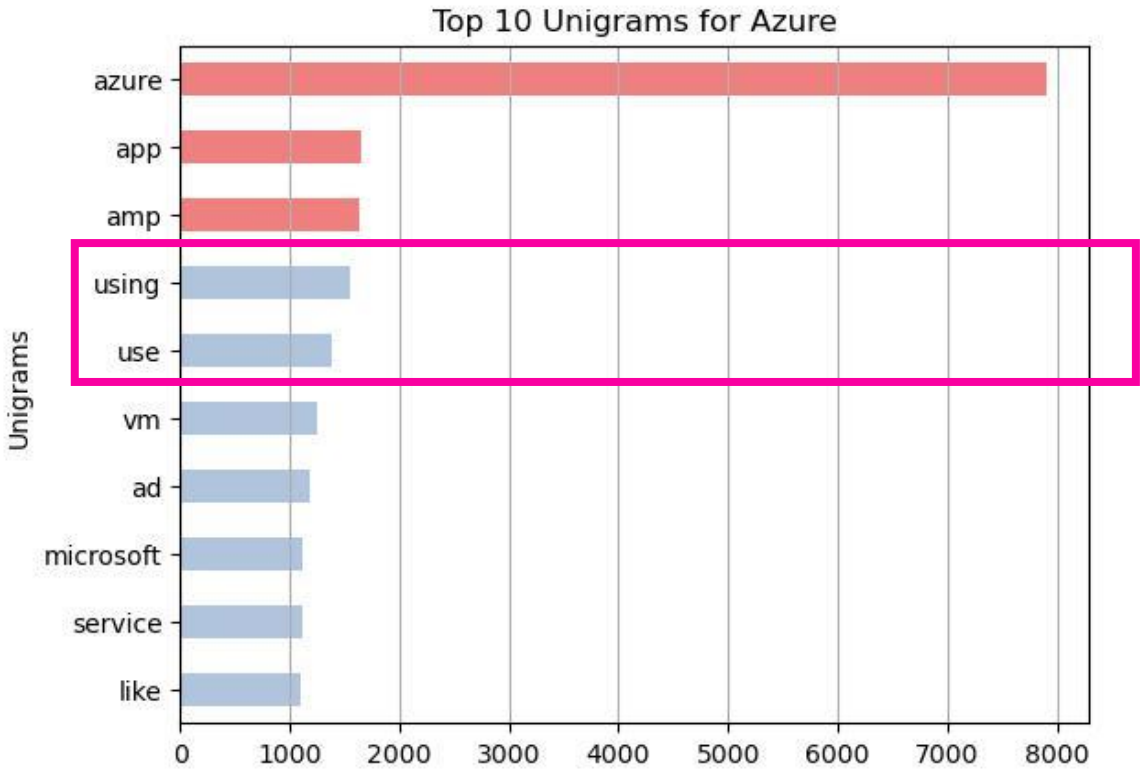


TF-IDF Vectoriser

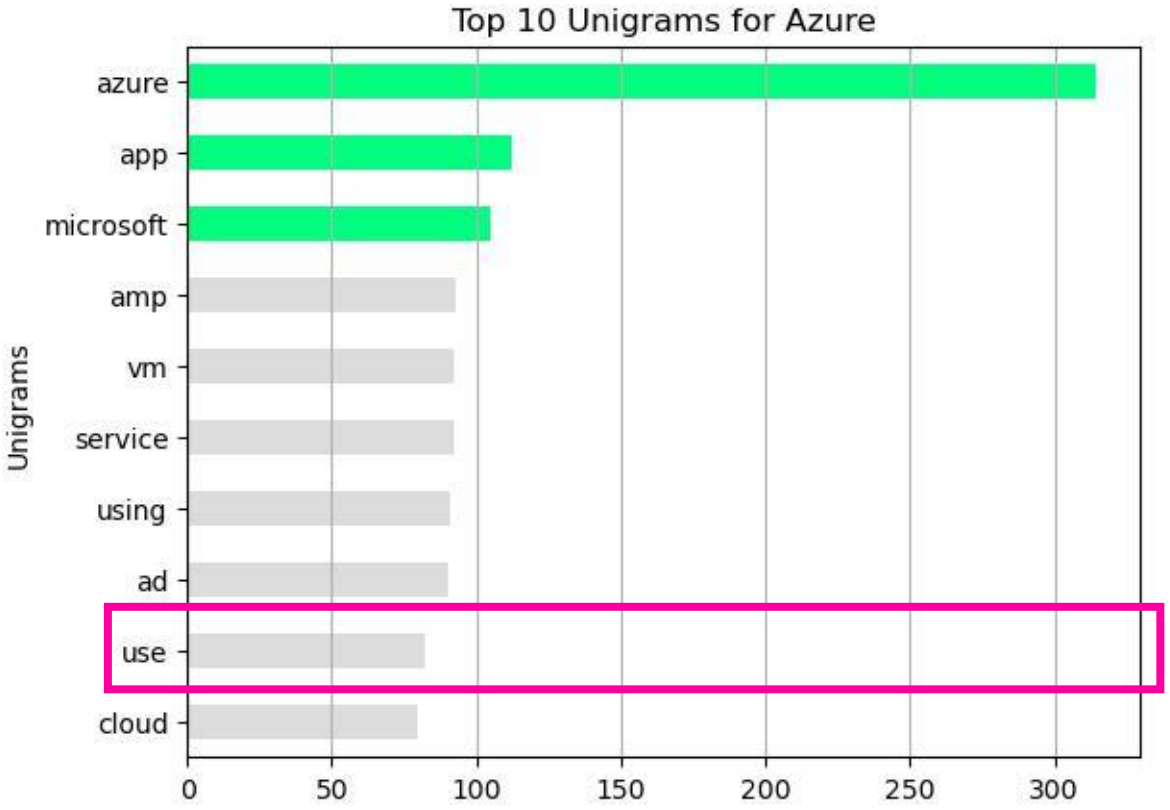


Top words in Azure

Count Vectoriser

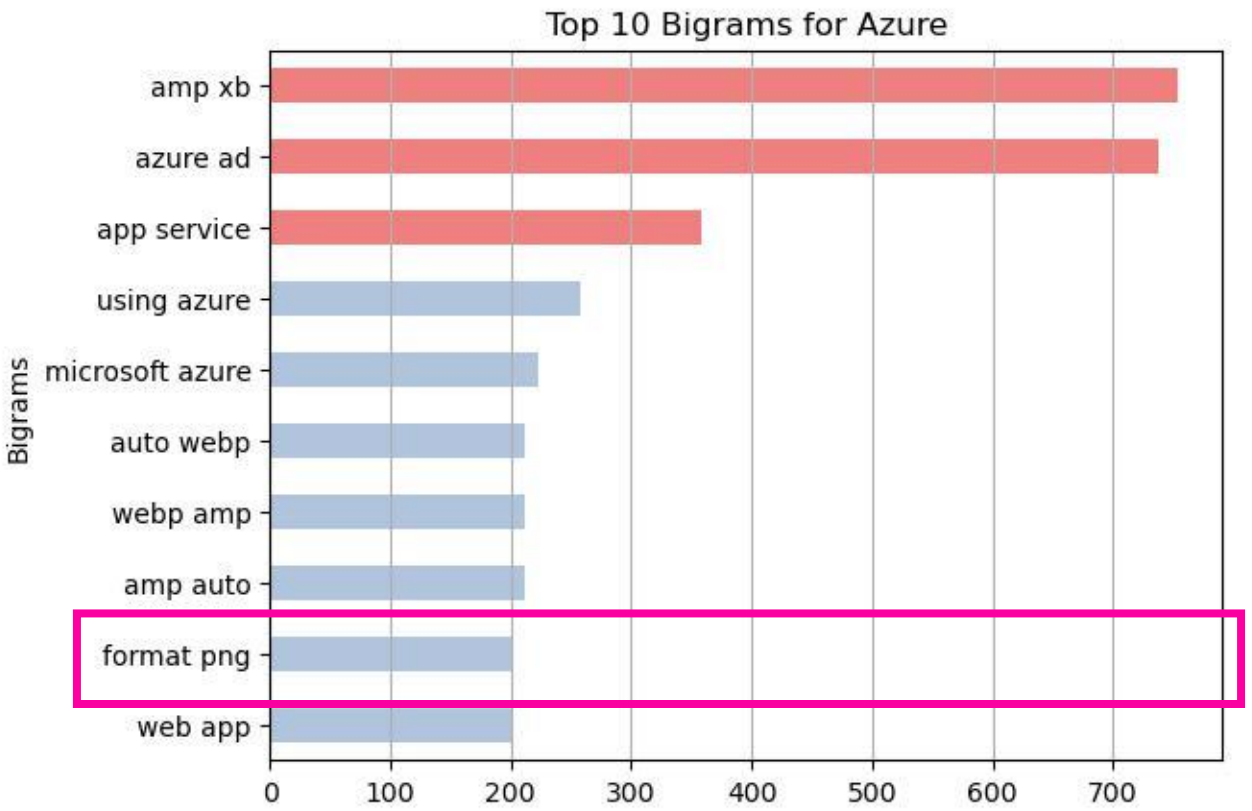


TF-IDF Vectoriser

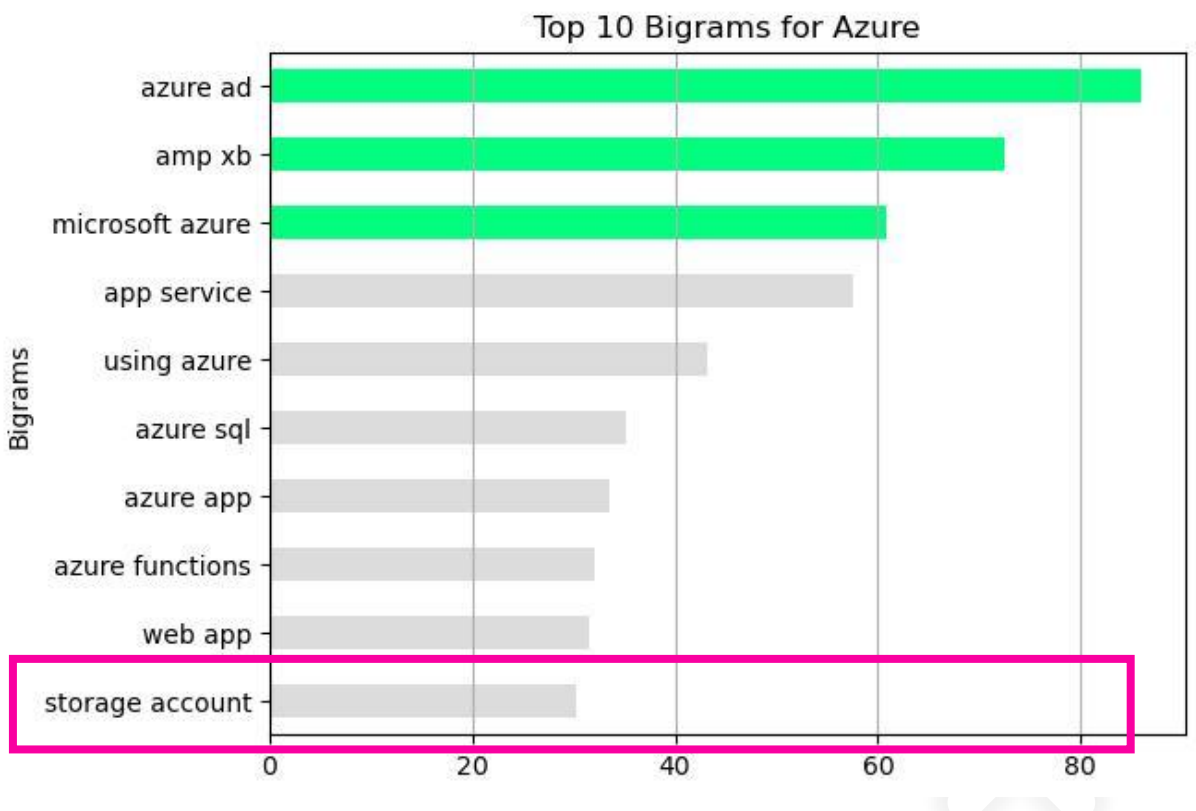


Top Bigrams in Azure

Count Vectoriser



TF-IDF Vectoriser



TF-IDF Vectoriser improves on Count Vectoriser

+ **Similar** to Count Vectoriser, in taking **frequency** of words.
+ But, places **less emphasis on** words that are common.

Emphasize

- EC Instance
- Storage Account

De-Emphasize

- Using
- Load Balancer
- Format Png

1. Baseline Model

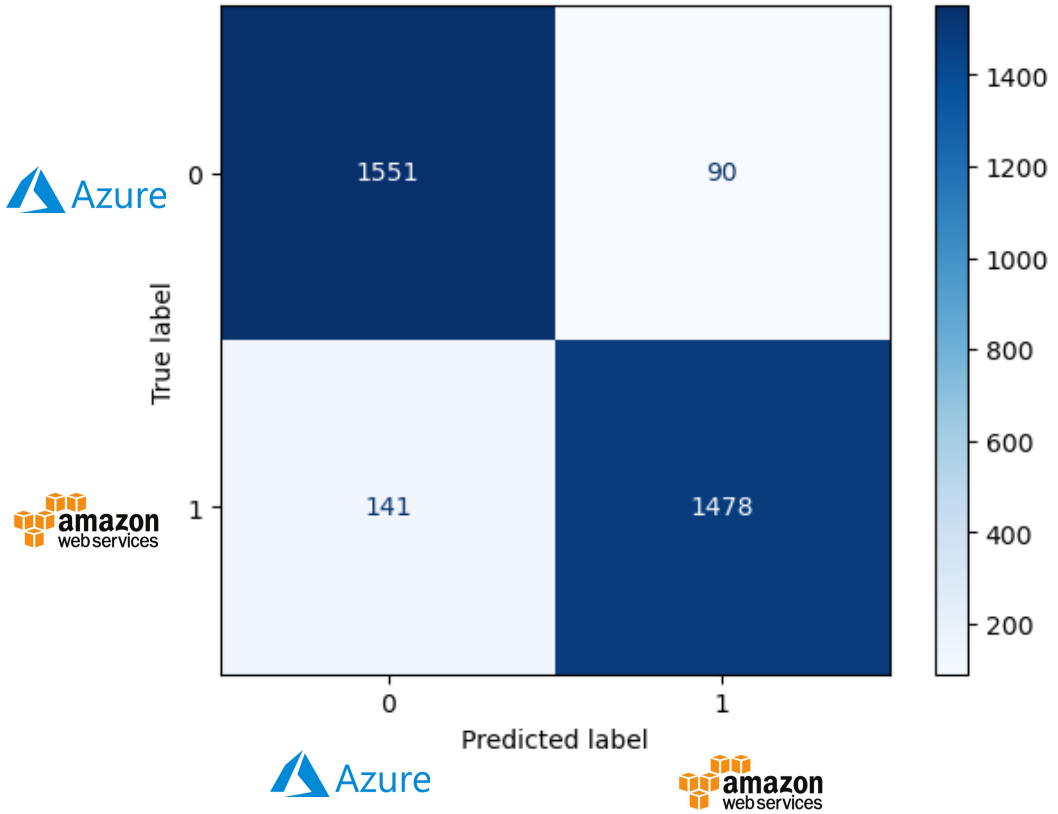
+TPR/Recall/Sensitivity is **0.766** and accuracy is **0.811** Model

Metric for Model evaluation

- **Can we improve the baseline model to be better than 81%?**
- Company has a larger team and reliance on AWS cloud services. Hence, more urgency for production team to respond to AWS related events.
- **TPR/Recall/Sensitivity**: Ability to correctly identify all AWS events:
  - Out of 100 AWS events, to get TPR of 90%, 90 AWS events needs to be correctly predicted.
  - This score will not be affected by amount of Azure events misclassified.

2. Naive Bayes Evaluation

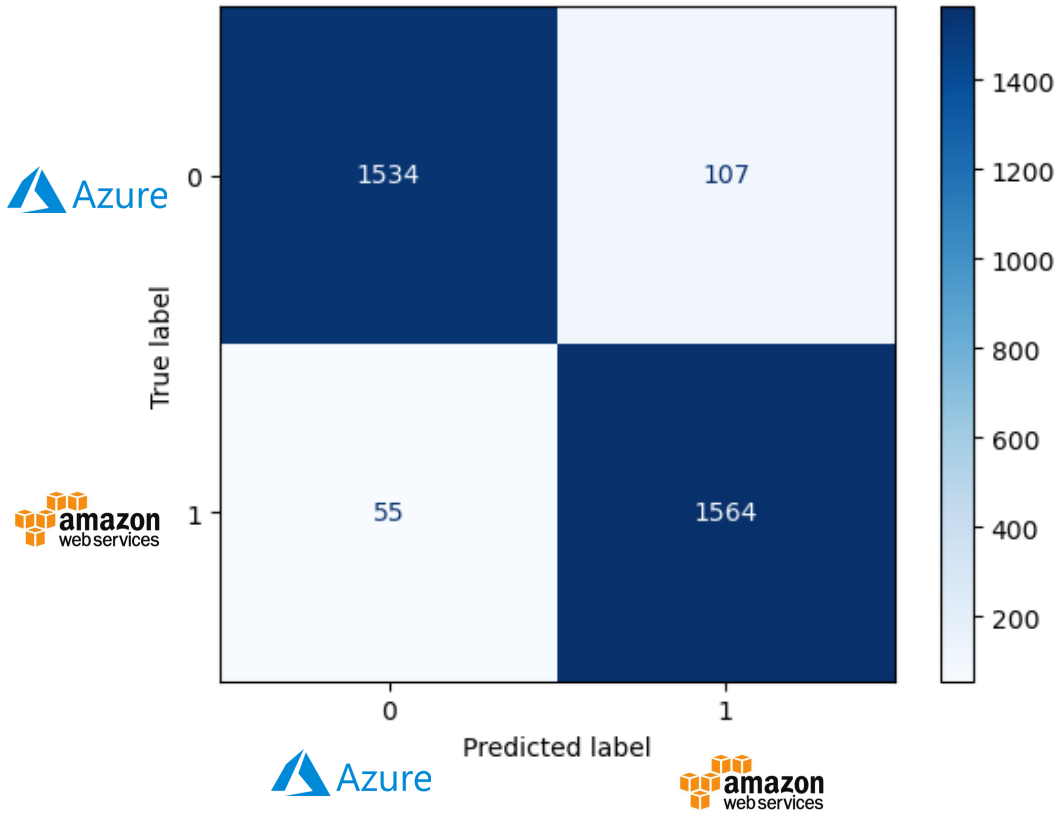
+ TPR/Recall/Sensitivity is **0.913** and accuracy is **0.929**



+ Multinomial Naive Bayes classifier is based on words being **independent** of one another.

3. Logistic Regression Evaluation

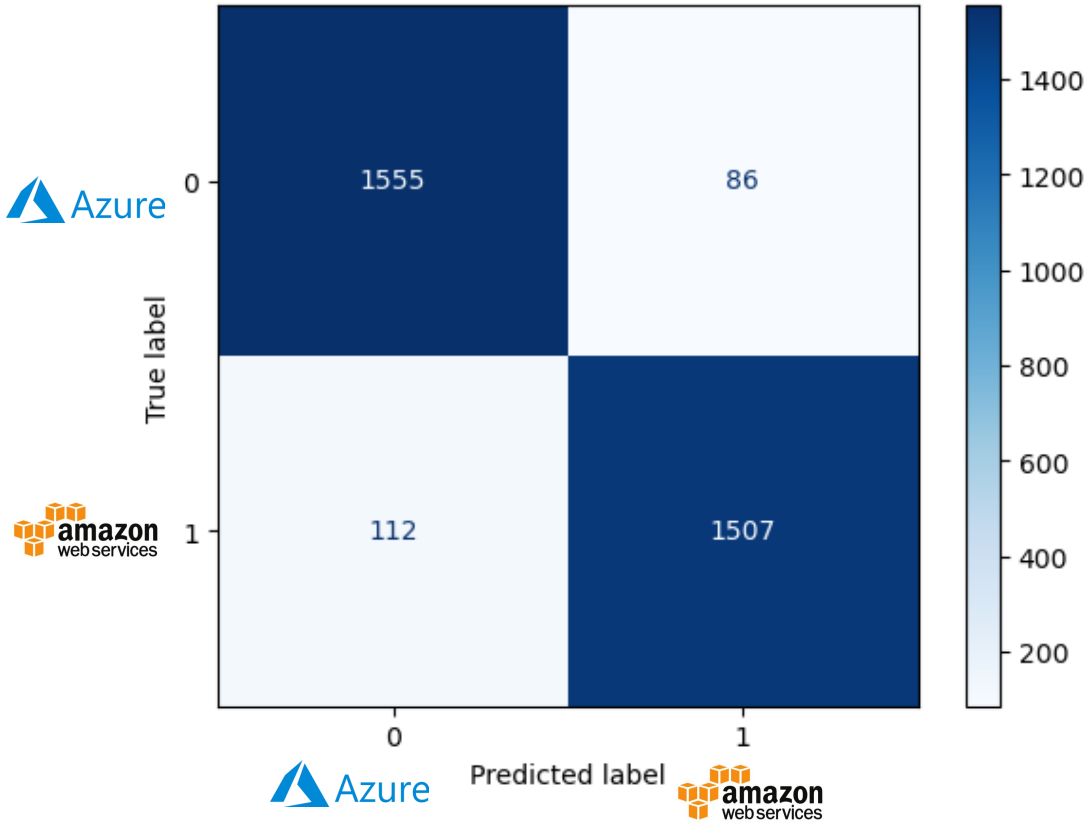
- TPR/Recall/Sensitivity is **0.966** and accuracy is **0.950**



- Naive bayes expects all words to be independent, Logistic Regression accounts for **correlation** between the words.

4. Random Forest Model Evaluation

- TPR/Recall/Sensitivity is **0.930** and accuracy is **0.939**



- A random forest is made up of multiple decision trees.
  - The predicted class will be **majority vote** based on most **important words.**
  - Reduced overfitting risk due to averaging of multiple decisions trees with **less correlated features**.

Model Selection

	model	vectoriser	hyperparameters	recall/sensitivity	accuracy
0	baseline	N/A	N/A	0.765749	0.811260
1	Naive Bayes	Count Vectoriser	[(cvec, CountVectorizer(max_features=3000, min_df=3, ngram_range=(1, 2),\n stop_words='english')), (nb, MultinomialNB())]	0.908586	0.930368
2	Naive Bayes	TF IDF Vectoriser	[(tvec, TfidfVectorizer(max_features=5000, min_df=3, ngram_range=(1, 2),\n stop_words='english')), (nb, MultinomialNB())]	0.912909	0.929141
3	Logistic Regression	TF IDF Vectoriser	[(tvec, TfidfVectorizer(max_features=5000, min_df=3, ngram_range=(1, 3),\n stop_words='english')), (lr, LogisticRegression(max_iter=500, random_state=42))]	0.966028	0.950307
4	Random Forest	TF IDF Vectoriser	{'rf__max_depth': None, 'rf__n_estimators': 300, 'tvec__max_features': 5000, 'tvec__min_df': 3, 'tvec__ngram_range': (1, 3), 'tvec__stop_words': 'english'}	0.930821	0.939264

- The priority is to classify AWS services related requests to be routed to company's support and DevOps team for triaging and response.
- Logistic Regression model is selected as it has the highest TPR and accuracy.

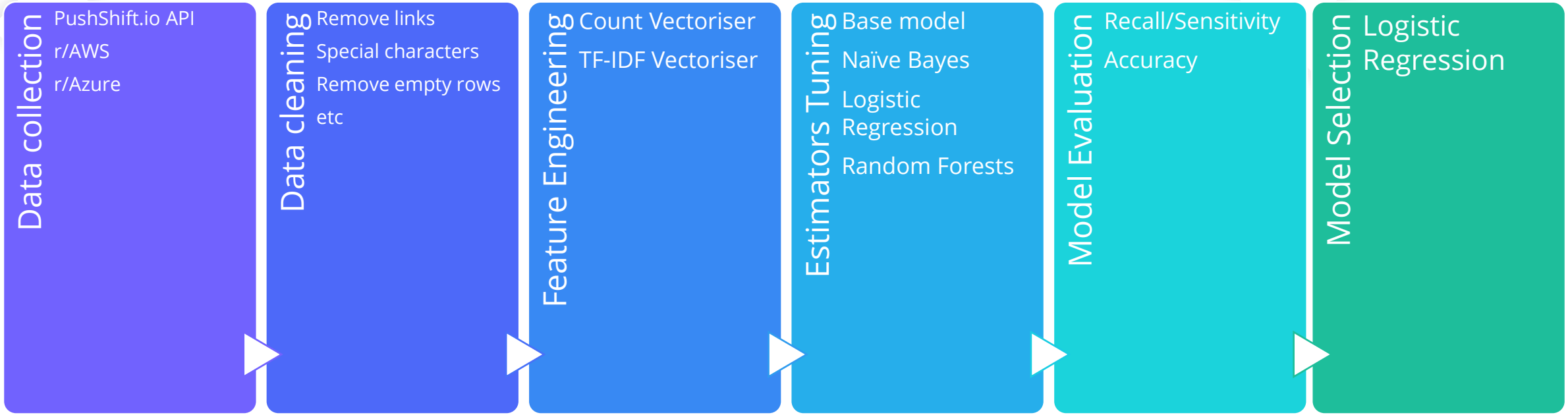
Findings and conclusion

- By using Logistic Regression, we can improve classification accuracy of AWS related IT request to the correct support team from **81%** to **96%**.
- **Time saved** in handshaking IT requests between different teams.

Future enhancements

+ Adding a more granularity to the category to predict for more fine tuned requests routing to support, DevOps, MLOps team.
+ Feeding AWS and Azure documentation corpus into the machine learning model for a more comprehensive coverage.
+ Additional derived features might give some insights to priority of the content.

Q&A and Summary



Problem statement:
Using publicly available subreddits data, this model aims to automate the classification process for IT requests raised for a company utilizing both AWS and Azure services, to reduce turn-around time for the relevant DevOps or support team to respond.