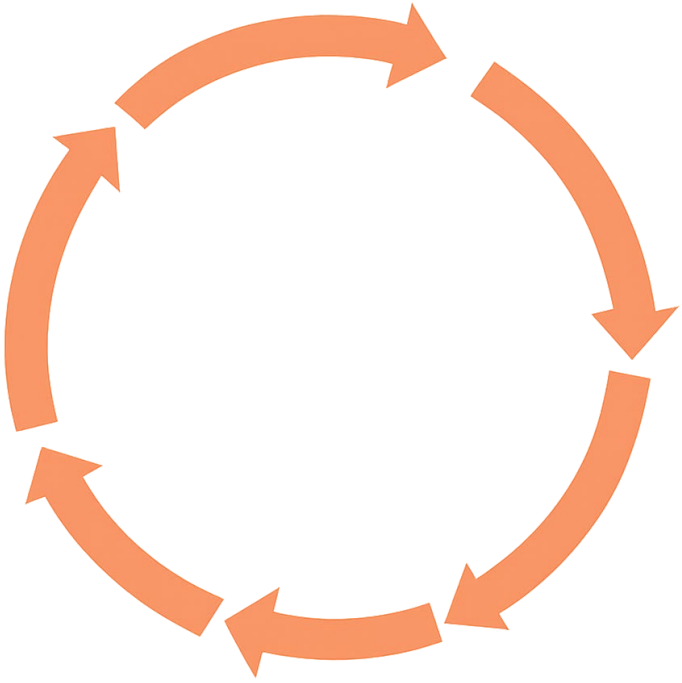
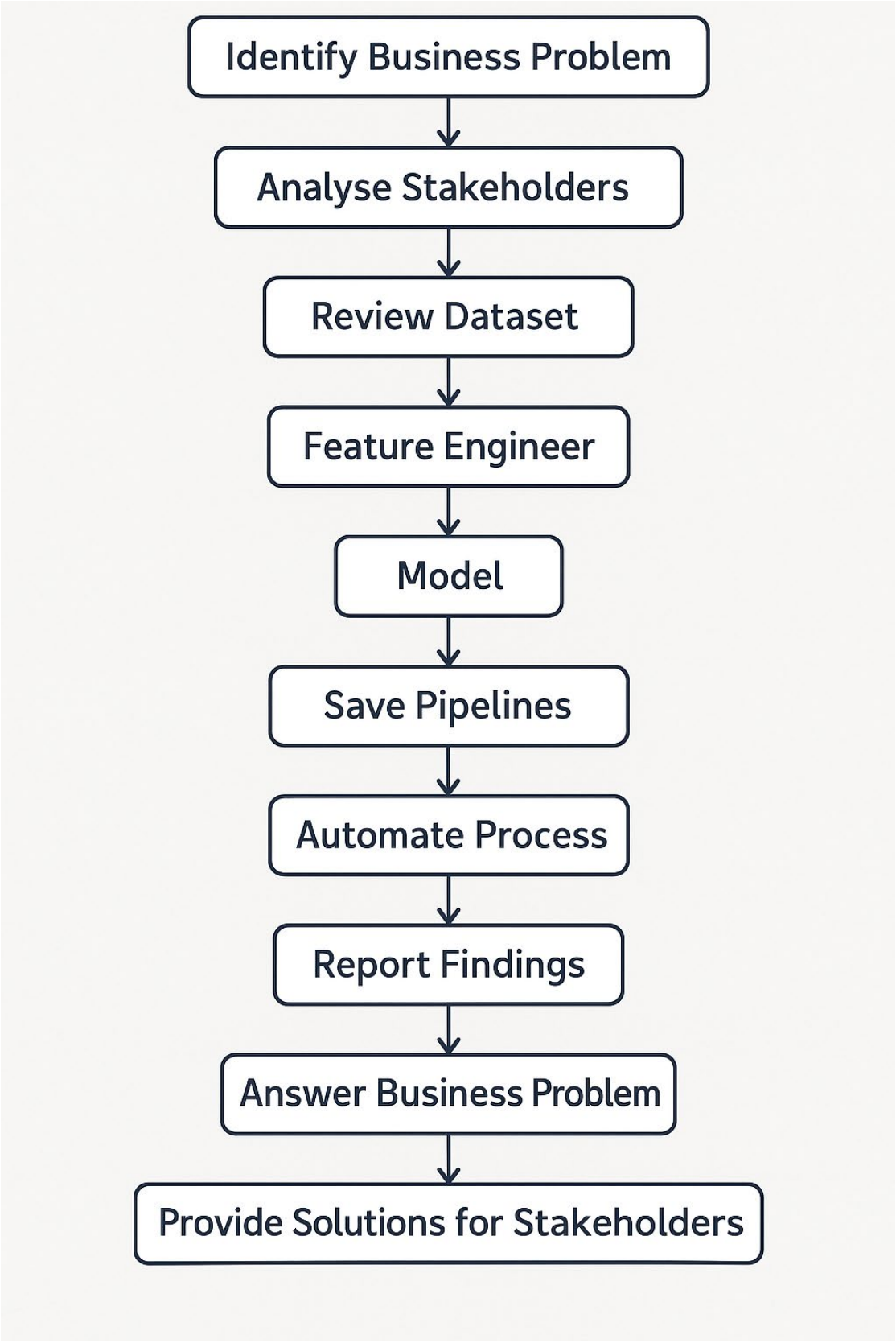
Connor Fordham – IOD Capstone Project 2025

Closing the Feedback Loop – Re-Claiming Churned App Customers



Process Overview



Index

Problem Statement

App customer retention is terrible, with high uninstall rates and high dissatisfaction. Companies are unable to effectively use feedback loops to understand their customers and retain them.

What is the problem or the opportunity this project is investigating?

The problem is app customers churn at an absurdly high rate, there is an opportunity to prevent this by implementing feedback loops. Which reportedly aren't being used by most companies.

Why is this problem valuable to address?

Reports show it costs 5-25 as much to acquire a new customer than it does to retain one. Companies with feedback loops make about 3 times as much as those who don't.

What is the current state?

49% of apps get installed within the first 30 days(customer churn) and 76% of companies don't use feedback loops.

What is the desired state?

To connect companies and customers. Companies want the business and customers want to be heard. They go hand in hand for success and customer satisfaction.

Has this problem been addressed by other research projects? What were the outcomes?

Yes. In 2016 Marketing Science dropped churning by 11% replying to customers within 48hours. In 2022 Stanford HCI

Group used BERT-based classifier to help collaborate with human to speed up loop processes. As well as many other data science/AI integration tactics deployed to attack this problem in multiple fields. McDonalds, AirBnB, Amazon and Duolingo are companies who also made huge strides in closing their feedback loops and found positive results.

Industry & Domain

What is the industry/domain?

App Economy, E-Commerce Sector & Marketing Sector.

What is the current state of this industry? (e.g. challenges from startups)

Massive volumes of reviews go unprocessed, 89% of all reviews on Google Play store ignored. Businesses lack systems to classify and act on feedback in real time. Dissatisfied app customers likely to churn, not have their issues fixed.

What is the overall industry value-chain?

$925 billion dollar industry, customer feedback is at the retention stage of this value chain.

What are the key concepts in the industry?

Churn rate, sentiment analysis, retention automation and AI-driven feedback management.

Is the project relevant to other industries?

Yes, any industry relying on feedback or online platforms can apply these solutions to improve engagement and overall success.

- 49% of apps uninstalled within 30 days
- 89% of reviews on play store un unanswered
- Companies who use feedback loops make 2-3x more than those who don't
- 76% of companies admit not knowing how to build effective feedback loops
- 76% of app customers feel brands ignore their feedback
- 88& of consumers more likely to purchase if brand listens to them
- 95% of customers are willing to give brands a 2nd chance
- 70% of customers will re-purchase if issues resolved
- Cost of acquiring a new customer costs 5-25x more than retaining an existing one

Stakeholders

Who are the stakeholders?

- **Business owners**
- **Business managers & marketers**
- **Customer support**
- **App developers**
- **Customers**

Why do they care?

Teams want to reduce churn and improve satisfaction. Owners want a thriving brand that maximises profits. Customers want timely and thoughtful responses to problems.

What are stakeholder expectations?

Internal: Actionable insights and reduced manual labour

External: Knowing their voice/brand matters and that actions lead to noticeable improvements in performance

Business Questions

What is the main business question that needs to be answered?

Can this dataset be used to build sustainable solutions to app customer churning? Can we recover them before they churn?

What is the business value of answering this question?

Up to 60% of increased revenue for a business succeeding in closing feedback loops vs not.

What is the required accuracy? What are the implications of false positives or false negatives?

False positive – Minor, slight resource wastage

False negatives – High, customer churned and money lost

Can we clearly identify dissatisfied customers, understand them and automate the process of responding to them. Creating better customer understanding and long-term retention.

Data Questions

Is it structured appropriately?

Does it reveal trends?

Can we use it to detect dissatisfaction?

Can we use it to uncover underlying themes?

Can we automate detection?

Can we automate responses to customers?

My Data

Where was the data sourced?

Mendely, also available on Kaggle. Mendely:
https://data.mendeley.com/datasets/chr5b94c6y/2

What is the volume and attributes of the data?

(751500, 8), my primary set it (99000, 8), shopping app reviews.

How reliable is the data?

Reliable, Mendeley is part of Elsevier which is a reputable academic publisher. Unlike most datasets though, fake review information is not available.

What is the quality of the raw data?

Raw, good quality. Mostly English, some spam and duplicates. Fake review information not available.

How was this data generated?

Reviews written on app store or play store. Scraped or platform supplied.

Is this data available on an ongoing basis?

Yes. Customer reviews will always be available, Kafka can simulate ongoing ingestion of these if we automate the process.

Columns = Review ID, Content(review), Score/5, Thumbs up Counts per Comment, Timestamp(at), Replied Content from Company, Replied Timestamp & Application Name.

	reviewId	content	score	thumbsUpCount	at	replyContent	repliedAt	appName
0	3e4bd1fc-bbc0-4862-9a53-4674535263b3	I have been an Amazon customer for YEARS. I ne...	2	29	1723341060000	NaN	NaN	Amazon shopping
1	92144bc8-41a3-4625-a6a5-83b81b9277ab	For the last 30 days or more I can only naviga...	2	38	1723192685000	NaN	NaN	Amazon shopping
2	05e6e39e-c098-4747-9fe7-2092892676cd	Although I absolutely LOVE the company and the...	2	12	1723345924000	NaN	NaN	Amazon shopping
3	0478ac98-b4b1-4934-a6f3-edd0ecf2cf00	Experience had gotten better but lately, they ...	2	8	1723204284000	NaN	NaN	Amazon shopping
4	9b7aedaf-8676-467c-b0bf-21e4fb494c54	Too many sponsored irrelevant items. About 40-...	2	6	1723191781000	NaN	NaN	Amazon shopping

My Data Science Process

Exploring Data

Removing Null Values

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99000 entries, 0 to 98999
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   reviewId       99000 non-null  object 
 1   content        99000 non-null  object 
 2   score          99000 non-null  int64  
 3   thumbsUpCount  99000 non-null  int64  
 4   at             99000 non-null  int64  
 5   replyContent   2 non-null      object 
 6   repliedAt      2 non-null      float64
 7   appName        99000 non-null  object 
dtypes: float64(1), int64(3), object(4)
memory usage: 6.0+ MB
```

Column Dropping

```
df.drop(columns = ['replyContent', 'appName', 'reviewId', 'repliedAt'], inplace=True, errors = 'ignore')
```

Date-Time Conversion

```
# Convert 'at' from milliseconds to datetime
df['review_date'] = pd.to_datetime(df['at'], unit='ms')

# Preview result
df[['at', 'review_date']].head()
```

	at	review_date
0	1723341060000	2024-08-11 01:51:00
1	1723192685000	2024-08-09 08:38:05
2	1723345924000	2024-08-11 03:12:04
3	1723204284000	2024-08-09 11:51:24
4	1723191781000	2024-08-09 08:23:01

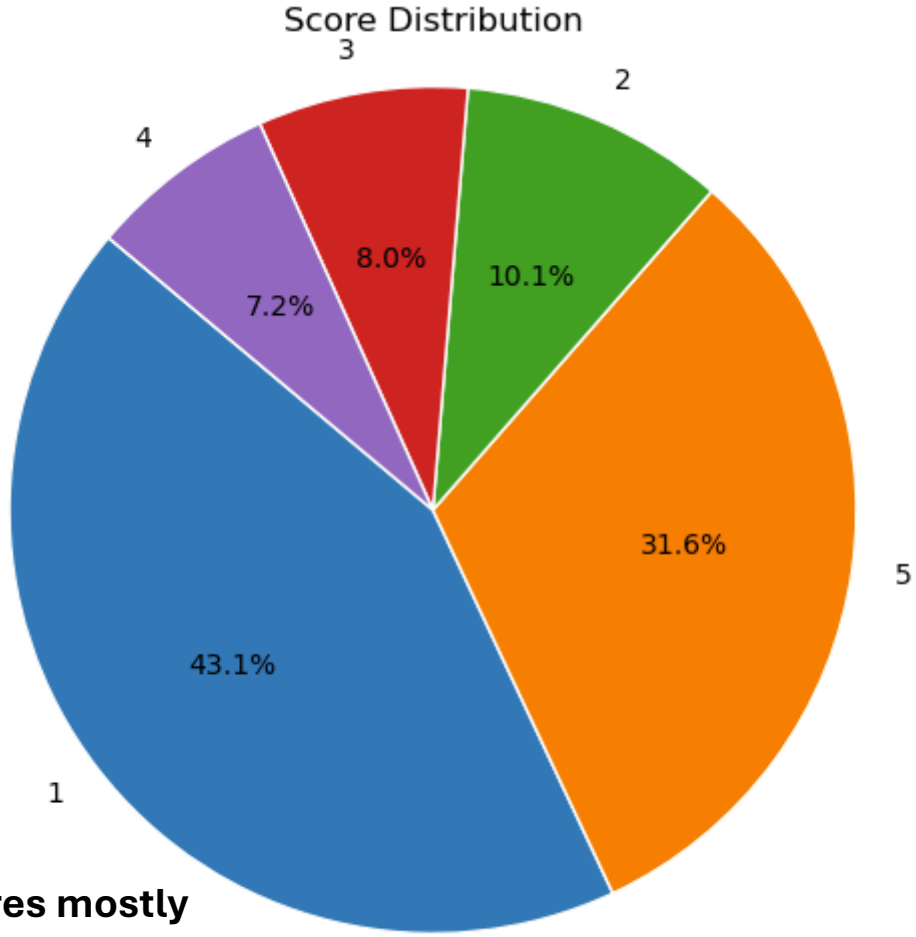
Language Preprocessing

NLTK, Spacy

- Incorporate slang terms
- Create domain-specific stop-words

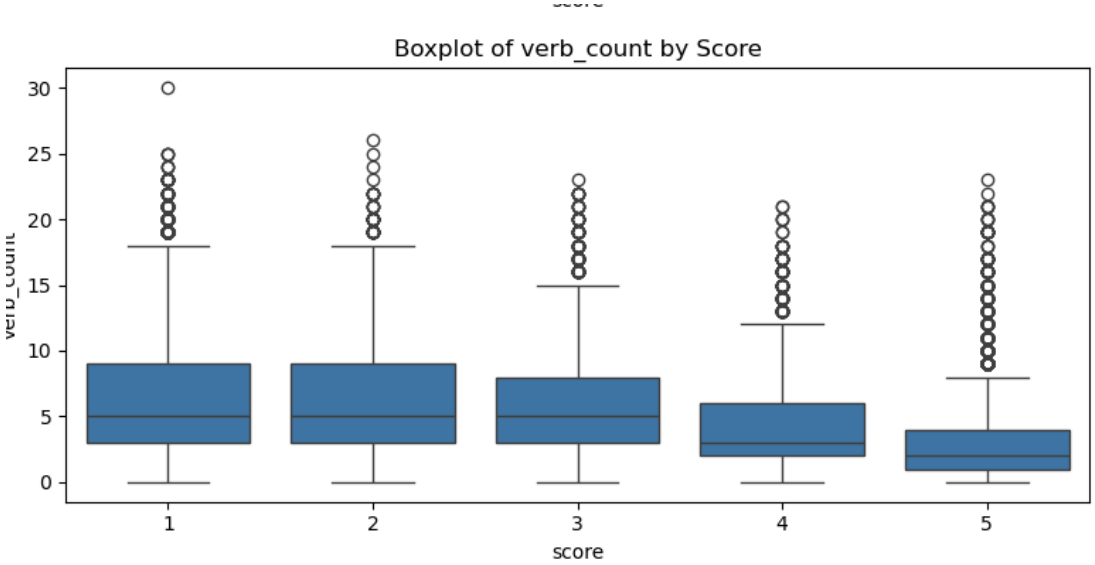
- Filter reviews to English
- Tokenise + Lemmatise using Spacy
- Remove domain stop-words
- Count POS tags (nouns, verbs, adjectives, adverbs)
- Compute text based features (word count, character length, negation words, exclamation/question marks, capital letter ratios)
- ***Exploring & Visualising Data***

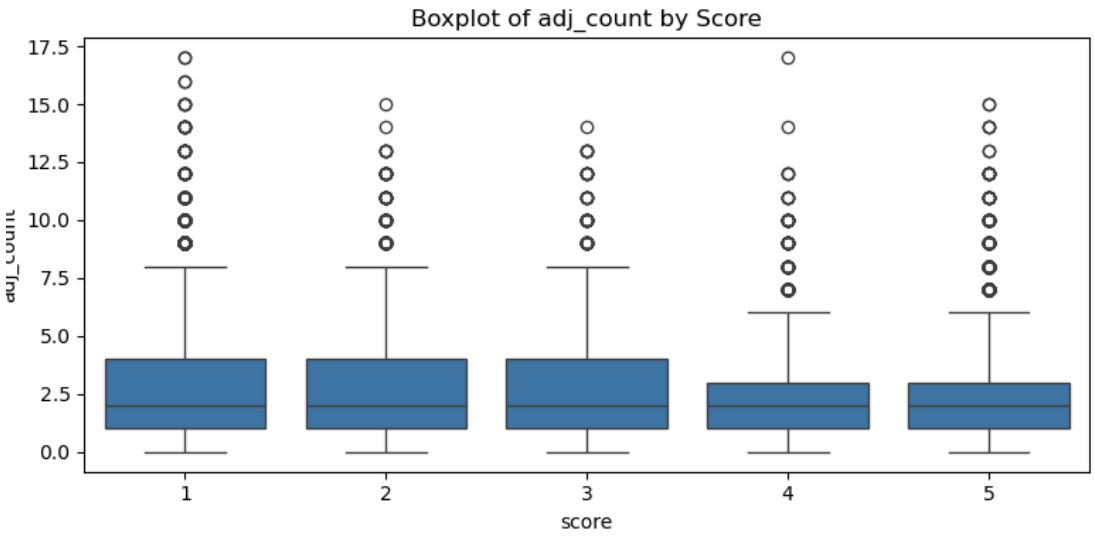
Pie Chart of Score Distribution



- **Negative scores mostly**

Boxplots of Scores vs Verbs/Adjective Counts





- **Lower scored reviews contain more descriptive text**
- **Test all relevant language columns vs score**
- **Negation and capitalisation also correlate with lower scores – keep stop words or customise ones to remove from text**

Feature Engineering Data

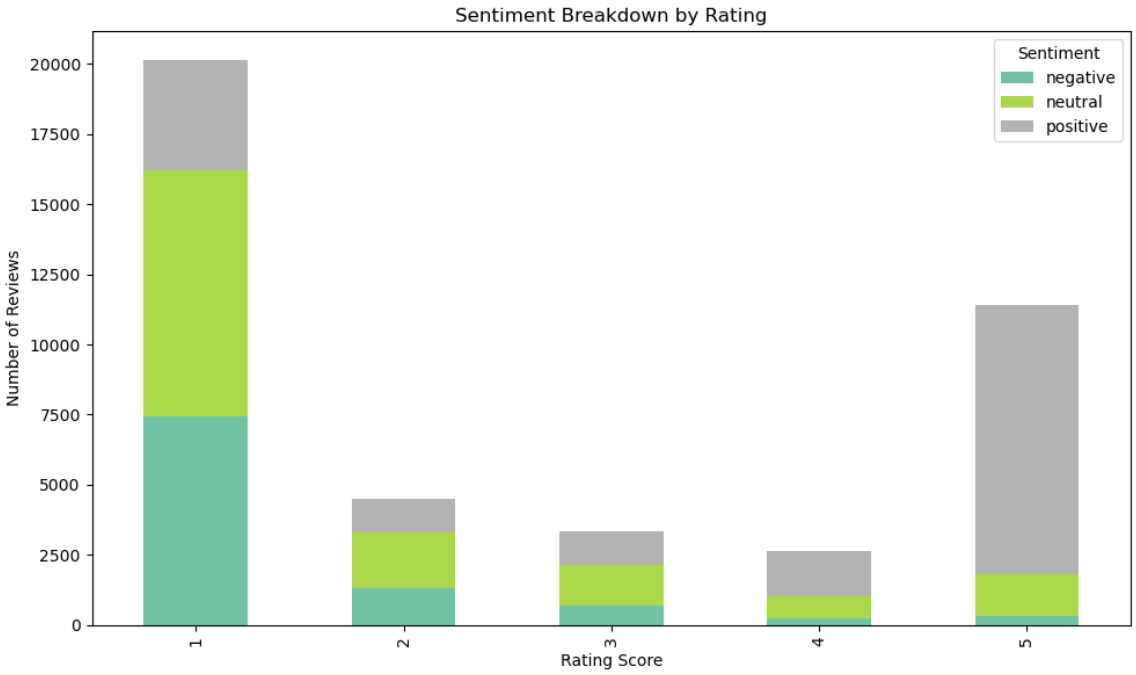
- **Sentiment score using TextBlob**
- **Label reviews based on sentiment**
- **BERT embeddings using SentenceTransformer() – delivers deep semantic language context**

Defining 'Unhappy' as Score<=3

Sentiment Polarity vs Unhappy Label

- **Negative sentiment more common in low scores**

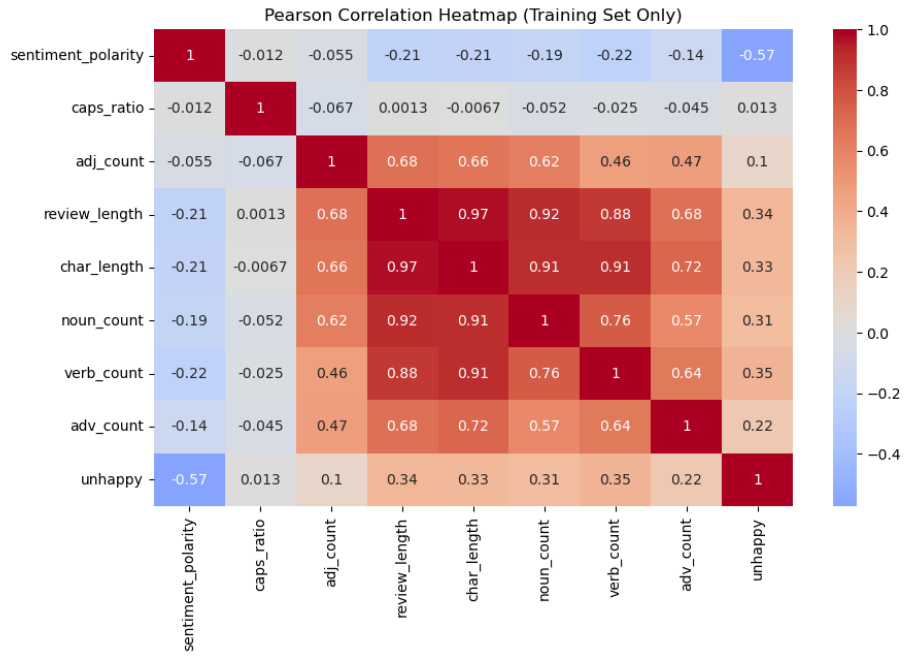
Sentiment Breakdown by Rating Scores



- **More positive/neutral sentiment then negative in 1 star reviews**
- **Rating does not equal sentiment**
- **Potentially customers who had a bad experience, but have good things to say about the company**

Split Train-Test now for Foolproof Testing – No Data Leakage

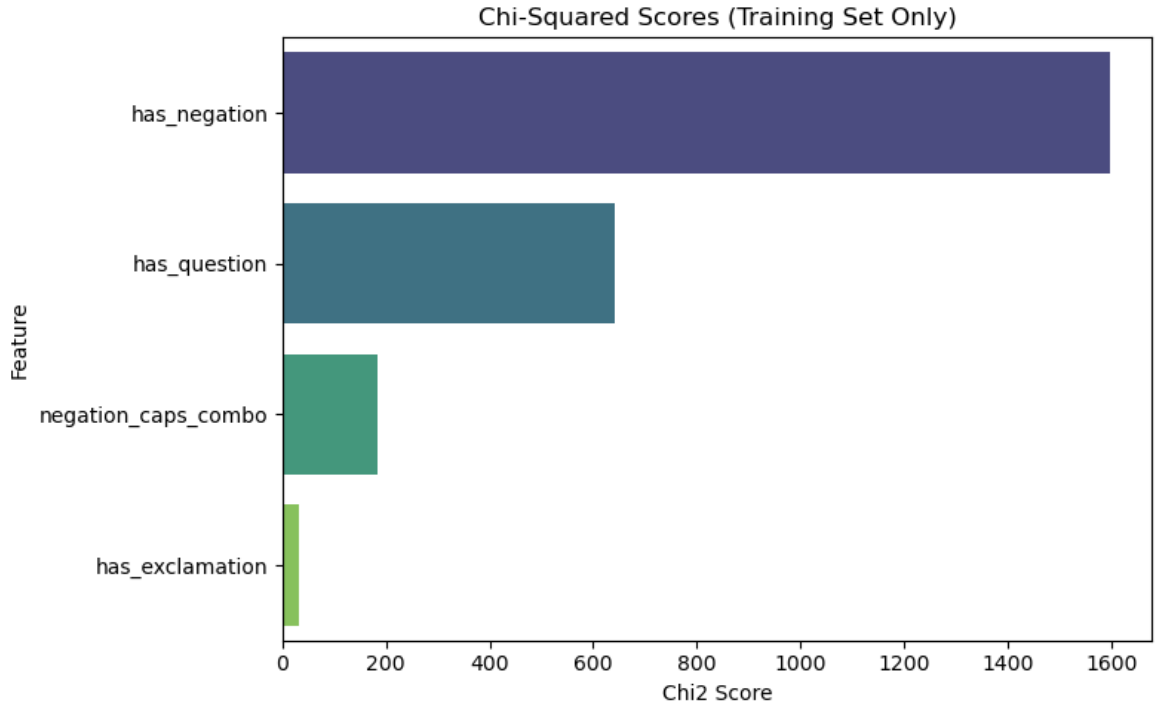
Numerical Feature Correlation – Pearson Correlation



Top correlated numeric features: ['sentiment_polarity', 'verb_count', 'review_length', 'char_length', 'noun_count', 'adv_count']

- **Top features for modelling: Sentiment polarity, review length, verb count, adjective count**
- **Features to remove due to multicollinearity(too similar to other features, model overfits on these): Noun count, character length, adverb count potentially.**

Categorical Feature Selection




```
Top chi² categorical features: ['has_negation', 'has_question', 'negation_caps_combo']
```

- **Top features: Has negation & has question.**
- **Only take 2, don't overcomplicate model**

Final Features

Sentiment polarity, review length, capitalisation ratio, adjective count, has question and has negation.

Modelling – Supervised

BERT – Pre-trained deep learning model. Understands tone and context.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_bert, y, test_size=0.2, stratify=y, random_state=42)

clf = LogisticRegression(max_iter=1000)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.92      0.83      0.87      2809
           1       0.92      0.96      0.94      5600

    accuracy                           0.92      8409
   macro avg       0.92      0.90      0.91      8409
weighted avg       0.92      0.92      0.92      8409
```

- **Very strong performance**
- **Understands emotion in text**
- **Recall for 1: 96% of all unhappy customers captured**

Featured Engineer Modelling

Logistic Regression

RandomForest

GradientBoosting

XGBoost

StackingClassifier – Best Model Above for Final Estimator

GridSearchCV to find Optimal Parameters for Each

Search within these Parameters

```python
# Define models and parameter grids
models = {
    'LogisticRegression': {
        'model': LogisticRegression(max_iter=1000, solver='liblinear'),
        'params': {
            'C': [0.1, 1, 10],
            'penalty': ['l1', 'l2']
        }
    },
    'RandomForest': {
        'model': RandomForestClassifier(),
        'params': {
            'n_estimators': [100, 200],
            'max_depth': [10, 20, None],
            'min_samples_split': [2, 5]
        }
    },
    'GradientBoosting': {
        'model': GradientBoostingClassifier(),
        'params': {
            'n_estimators': [100, 150],
            'learning_rate': [0.05, 0.1],
            'max_depth': [3, 5]
        }
    },
    'XGBoost': {
        'model': XGBClassifier(eval_metric='logloss', use_label_encoder=False),
        'params': {
            'n_estimators': [100],
            'learning_rate': [0.1],
            'max_depth': [3, 5]
        }
    }
}
```

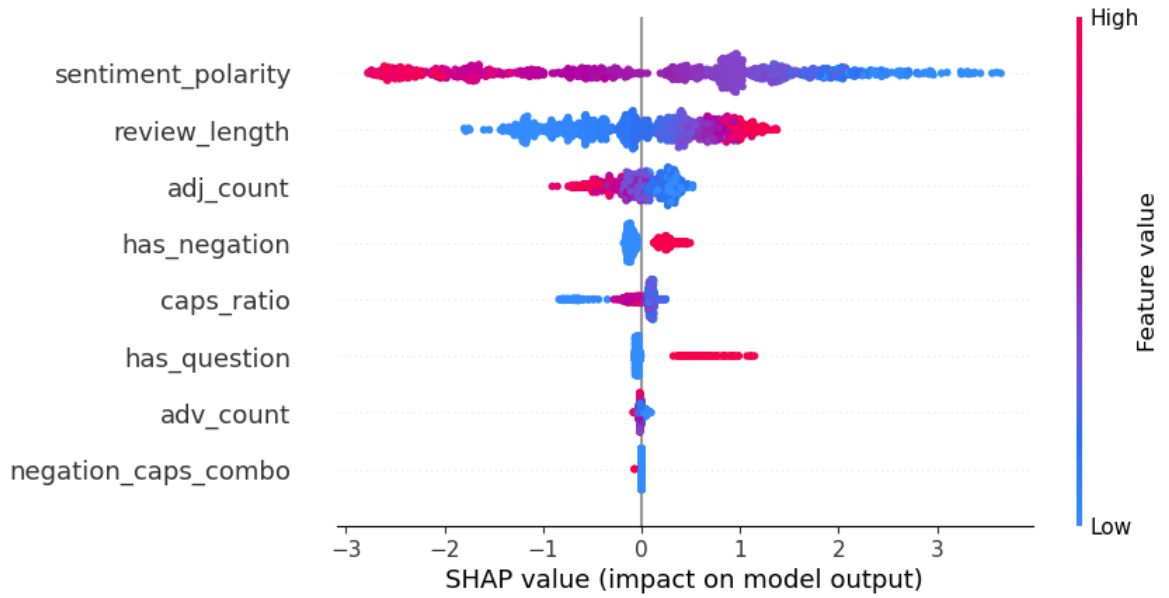
Final Stacking Ensemble Results Using XGBoost Final Estimator

Stacking Ensemble with XGBoost Results:

	precision	recall	f1-score	support
0	0.81	0.71	0.75	2808
1	0.86	0.91	0.89	5600
accuracy			0.85	8408
macro avg	0.83	0.81	0.82	8408
weighted avg	0.84	0.85	0.84	8408

- **Also good results**
- **Recall for 1: 91% of all unhappy customers captured**

Feature Evaluation – SHAP Value



- **Validates features used**

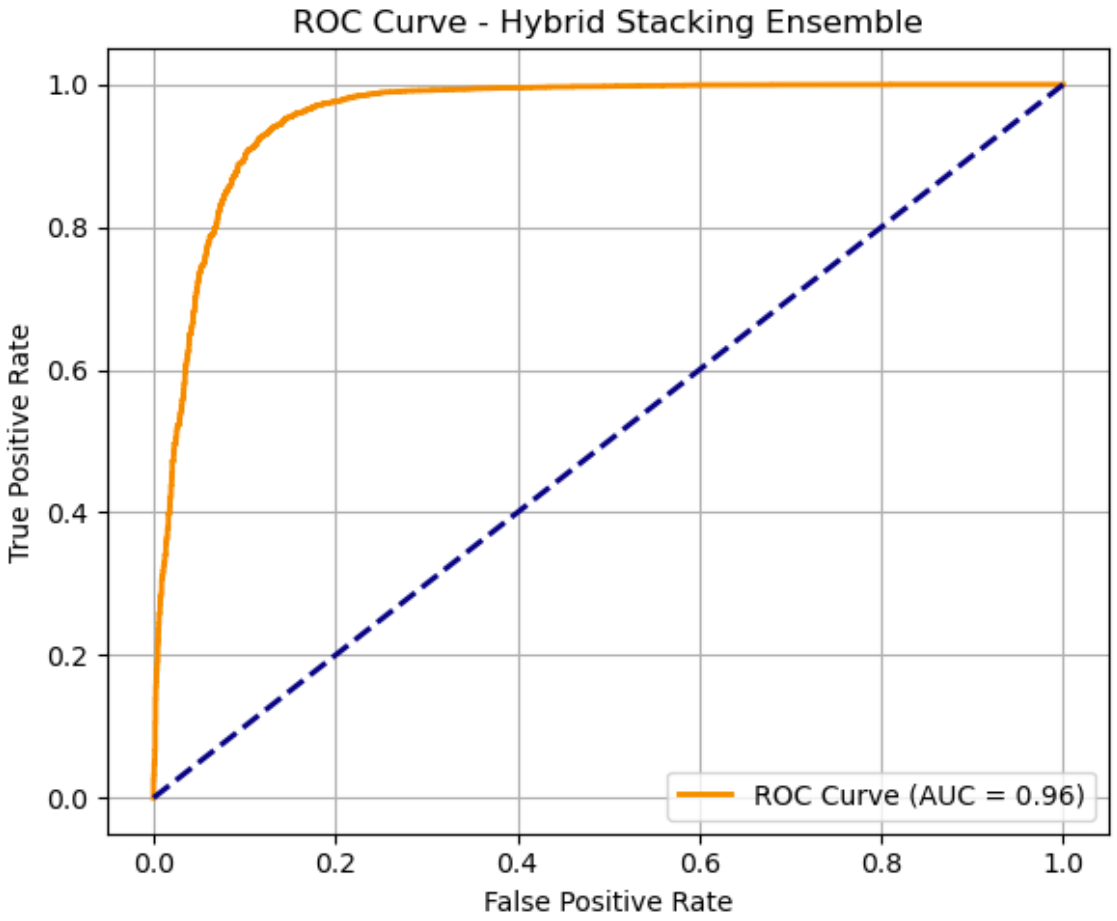
Hybrid Model – Stacking Ensemble + BERT

```
Hybrid Stacking Ensemble Results:
              precision    recall  f1-score   support

           0       0.91      0.84      0.87      2808
           1       0.92      0.96      0.94      5600

    accuracy                           0.92      8408
   macro avg       0.92      0.90      0.91      8408
weighted avg       0.92      0.92      0.92      8408
```

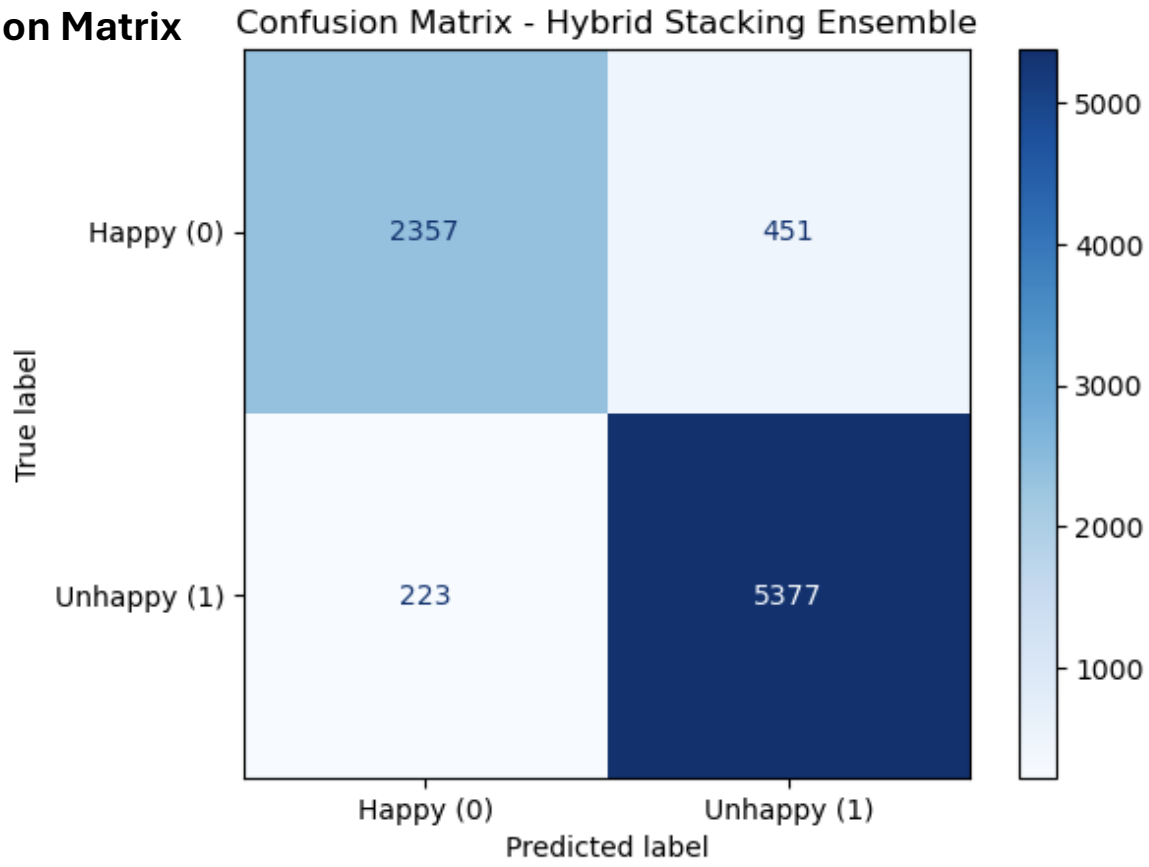
- **Small improvements**
- **More robust, use this**
- **BERT can miss obvious indicators at times, needs other features as support for success**

Result Graphs

ROC Curve



Confusion Matrix



- **Both strong**
- **ROC Curve showing good generalisation**
- **Confusion matrix showing minimal errors. False positives and false negatives low.**

Save Model Pipelines and Test on New Company Data

Preprocessing saved as: 'run_full_processing.pkl'

Model saved as: 'stacked_hybrid_model.pkl'

Test on AliBaba, AliExpress and Shein – As Results Hold True, Increase Test Sizes for Robustness Test of Models

Final Results on 10,000 Test Size from Shein

Hybrid Stacking Ensemble Results on SH Company Data:

	precision	recall	f1-score	support
0	0.95	0.92	0.94	7500
1	0.79	0.86	0.82	2483
accuracy			0.91	9983
macro avg	0.87	0.89	0.88	9983
weighted avg	0.91	0.91	0.91	9983

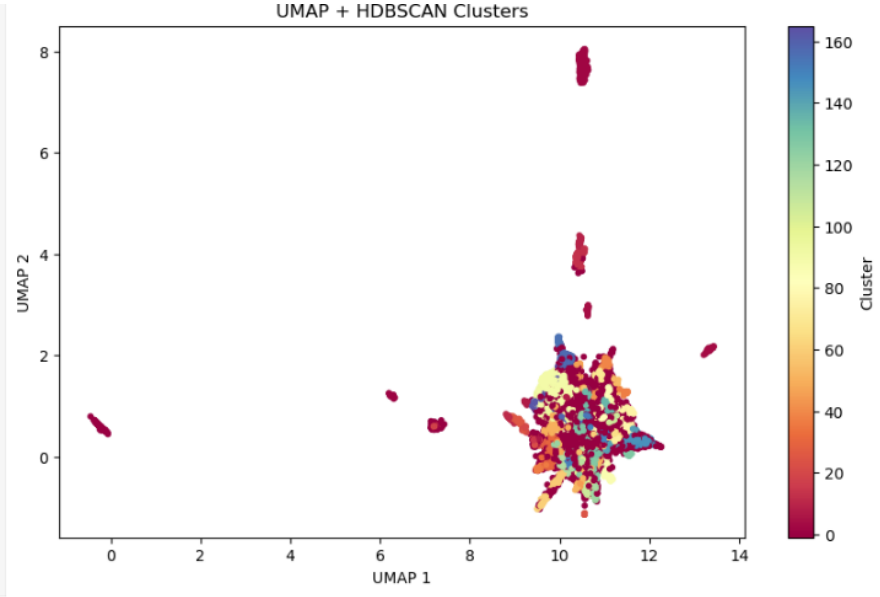
- **Still performing well**
- **Recall for 1: 86% of all unhappy customers captured**
- **Generalisation good and ready for deployment**

Unsupervised Learning

Dimensionality Reduction using UMAP – Preserves structure well

HDBScan – Density based, handles noise well

Initial Results



How Dense is Each Cluster

```
df2['cluster'].value_counts().head(20)
```

```
cluster
-1      15928
 90      3274
 50      3028
 52      1216
 101      756
 2        658
 157      562
 30       534
 152      490
 145      406
 135      405
 75       401
 149      398
 36       389
 0        384
 141      372
 146      361
 159      359
 93       315
 26       296
Name: count, dtype: int64
```

- **-1 is noise**
- **Lots of noise**
- **Consistent cluster sizes other than that**
- **Key words in clusters insightful, not enough**

Negative Topic Clustering

- **Focusing on negative sentiment**
- **Using UMAP and Kmeans this time**

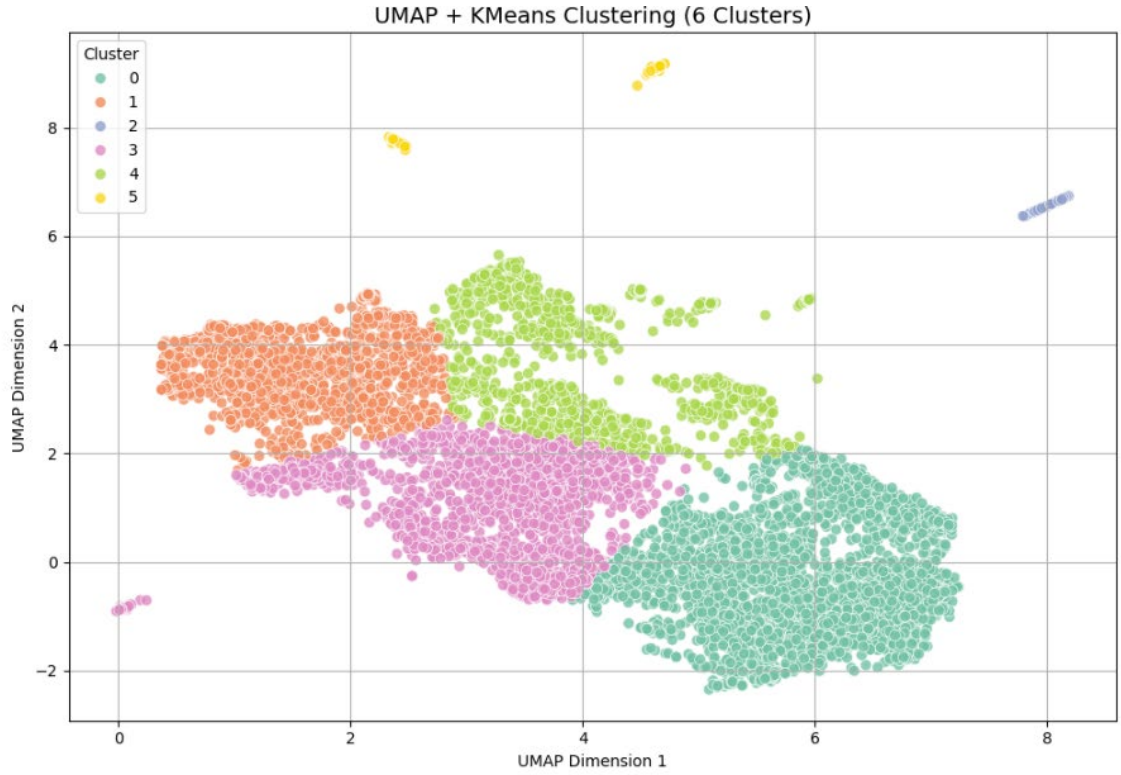
Test Scores – for different parameters

Silhouette Score – Measures how well each point fits within its own cluster vs how far it is from the next. Higher = Better

Davis-Bouldin Score – How similar the clusters are. Lower = Better

Use dimensionality reduction, cluster amount experimentation, amount of neighbours in each cluster and distance required between points to cluster to find best scores.

End Result



TF-IDF used to get key words from clusters

TF-IDF – Term Frequency – Inverse Document Frequency. Numerical statistic used to reflect how important a word is in a collection.

We used TF-IDF to get bigrams of key comments in clusters and score them. One example below:

```
📍 Cluster 2 — Top 10 Unique Bigrams (TF-IDF):
rufus ai (score: 9.070)
rufus annoying (score: 6.266)
hate rufus (score: 4.200)
way disable (score: 3.894)
ai rufus (score: 3.859)
disable rufus (score: 3.847)
option disable (score: 3.460)
option turn (score: 3.446)
stupid rufus (score: 3.356)
don want (score: 3.231)
```

We had 6 clusters in the end, we summarised their topics and made them into actionable insights below:

Clustering Issue Topics and Solutions

Cluster 0 - Customer Service + Prime Delivery Issues : Improve Support & Delivery SLAs

Cluster 1 - Buggy App Performance : Fix App Stability & Regression Testing

Cluster 2 - AI Assistant Rufus Backlash : Add Opt-Out or Rework Rufus Option

Cluster 3 - App UX Frustration + Past Orders/Kindle Issues : Review User Flow UX and Errors

Cluster 4 - Search Feature Frustration : Rebuild or Refine Search Experience

Cluster 5 - Dark Mode & Language Settings : Prioritise Dark Mode & Localisation

Pipeline Saving

Preprocessing and modelling pipelines saved, ready for importation to streaming services such as Kafka. Pkl & py files

CrewAI Testing

Automated AI agents, trained as experts in roles and given specific tasks. Train them on the company's issues we have discovered and have them work together to compose email replies to each complaint.

Roles in AI team when they read a review, all working together passing knowledge to the next:

Tone Profiler: Understand emotional tone, is this customer open to a recovery email?

Recoverability Checker: Is the customer too angry or trolling, specialises judgement on reply worth. Gives yes or no.

Issue Classifier: Is it one of the issues we identified while clustering, if so be extra sensitive. We know our company is making mistakes here. If not, mark as general dissatisfaction and pass context on.

Response Strategist: Define brief recovery plan based on information derived so far.

Email Writer: Write personalised human email, referencing issue identified and invite the customer to engage if we deem them recoverable.

Example Email and Evaluation below:

```
Review:
This is absolutely awful. I'm deleting this app today.
```

```
# Agent: Recoverability Evaluator
## Final Answer:
no: The customer's angry, frustrated, and disappointed tone indicates a low likelihood of receptiveness to a recovery email at this time.
```

- **Agents successfully derive this customer is not recoverable**
- **Based on our trained settings, will reply with an apology or not at all**

Kafka Stream Launch

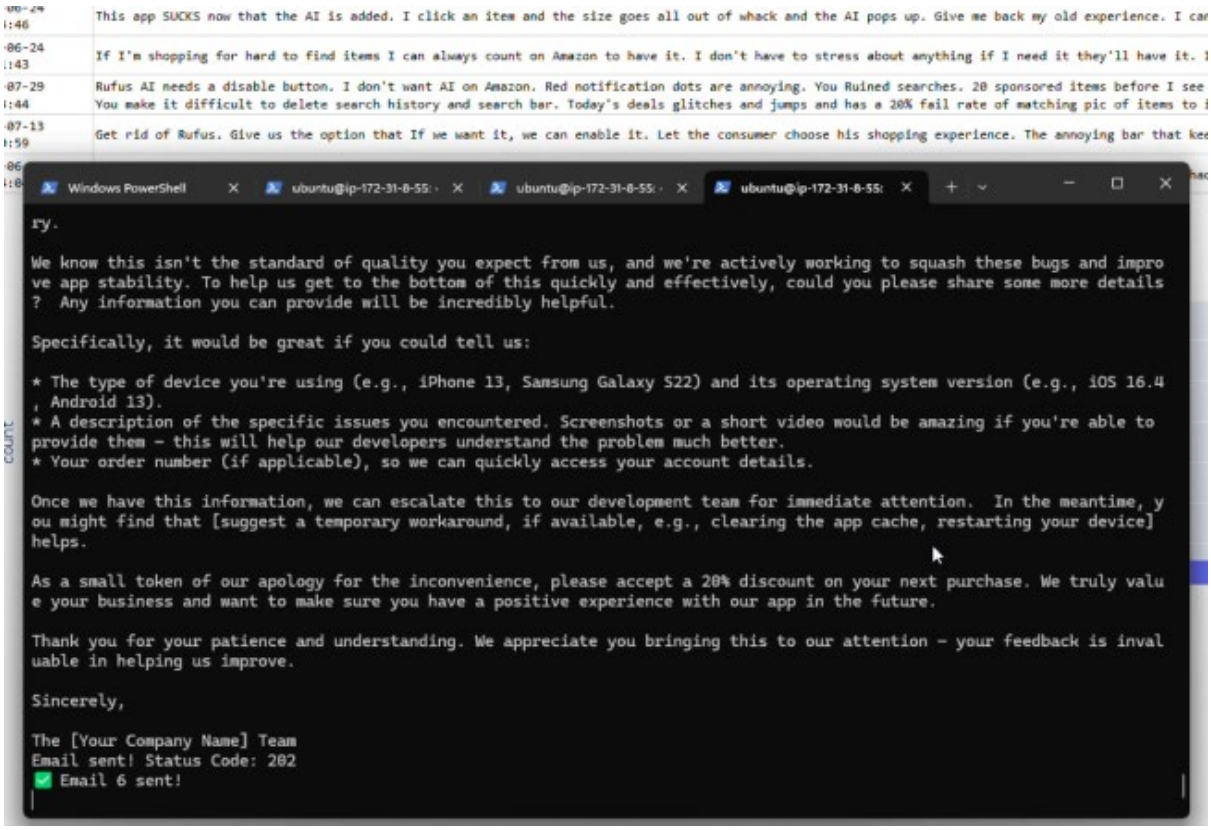
- **Setup AWS & EC2, port access for cloud use and streaming**
- **Open multiple ports if you want multiple dashboards**
- **Ensure using appropriate GBU, 2 CPU's optimal and 35-45gb of space for setup on AWS + set budget limit**
- **Download all relevant python programs to Kafka**
- **Upload csv review files(e.g. Amazon shopping.csv), pipelines(preprocessing and modelling)**
- **Create in-built nano files, saved in Kafka to read and process these together like in your lab**
- **Create dashboard code to run with saved pipeline, this defines the dashboard setup and how is fed your pipeline**
- **Files are consumer_dashboard.py ad kafka_csv_consumer.py**
- **Run both and the dashboard will run live modelling detection results, visualising**
- **Exports unhappy customers to new file on cloud**

Live Dashboard after 70,000 Reviews Processed

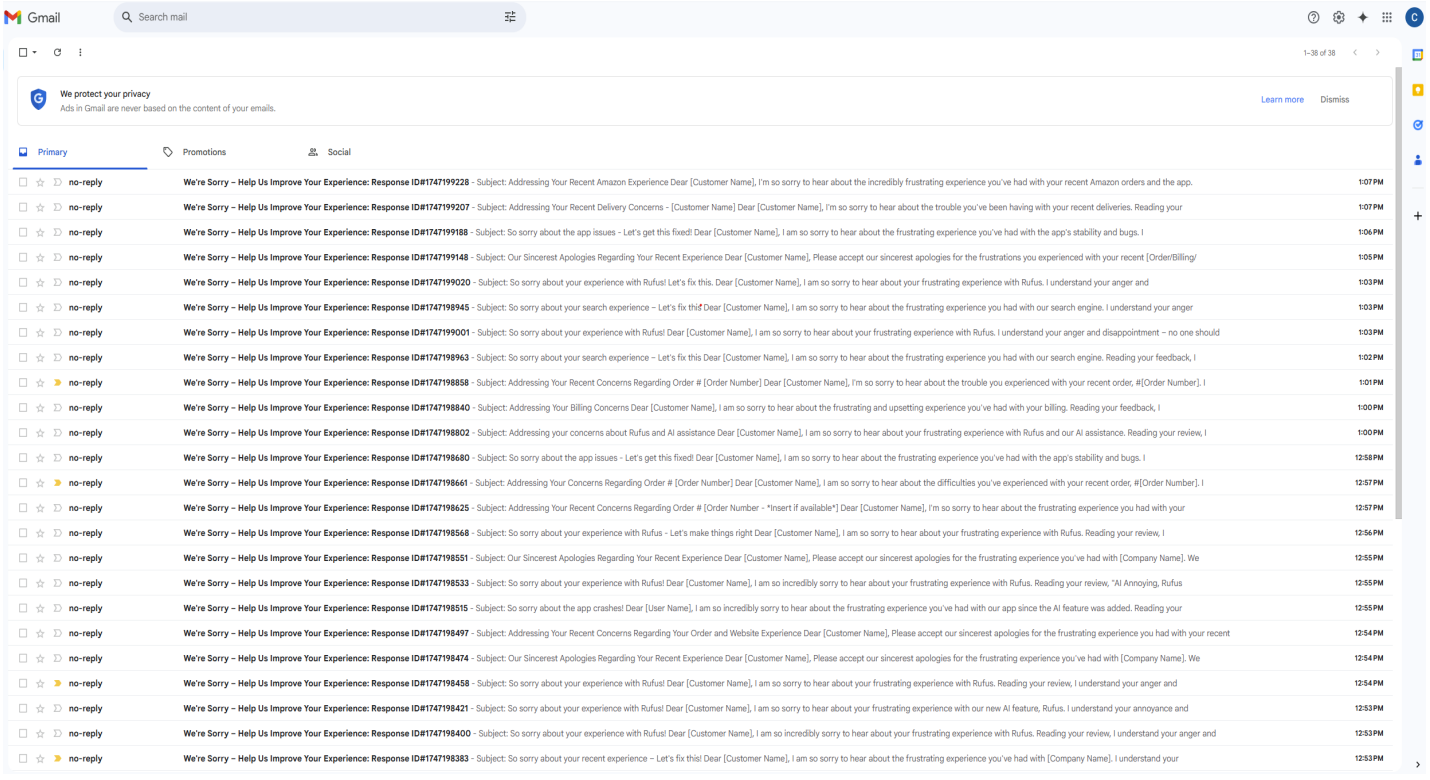
CrewAI Email Automation in Kafka

- **Input CrewAI setup into Kafka nano now**
- **Edit coding to read from unhappy review file being funnelled from dashboard**
- **Have it save those emails to another csv on cloud**
- **Setup auto email system and alert you when sent**
- **Setup auto email API with SendGrid, authenticate with domain and CNAME code verification**
- **Input dummy email address as placeholder for email receiving**
- **Estimate costs per Gemini 1.5 LLM request, code in live cost updates to track**
- **Nano file is now crew_emails.py**
- **Run and see results**

Successful Email Sending



Inbox Receiving Live Email Thread for each ReviewID



- **Emails read well, CrewAI can be refine for each customer for future use**
- **Sending successful, will need to increase Google Cloud quota if intending to send more than 1,000 emails a day**
- **SendGrid will also require subscription for more than 100 free emails. $21 a month allows 100,000**

Data Solutions

Is it structured appropriately?

Yes, easy to work with structure. Rich enough for NLP and informative enough for modelling.

Does it reveal trends?

Yes, throughout all parts of the data process. Early analysis reveals clear indications of unhappiness in app customers and modelling proves it.

Can we use it to detect dissatisfaction?

Yes, with great accuracy. Using BERT and various ensemble machine learning models. Over 90% on in-domain data.

Can we use it to uncover underlying themes?

Yes, unsupervised learning helps us do this. UMAP, HDBSCAN and KMeans. Clusters and topics are revealed, this can be done for any company's data.

Can we automate detection?

Yes, using Kafka and AWS as demonstrated. We can streamline the entire process and refine as we go.

Can we automate responses to customers?

Yes, using CrewAI and Kafka integration. We can train the program to send contextually accurate feedback responses and bulk send using SendGrid.

Business Solutions

Can this dataset be used to build sustainable solutions to app customer churning?

Yes, we produced a solution pipeline through 5 internal solutions.

1. **Real-time Customer Dissatisfaction Detection**
2. **Topic-Based Complaint Clustering**
3. **Feedback Dashboard for Stakeholders**
4. **Automated Feedback Loop with CrewAI Agents**
5. **Scalable Pipeline for Other Apps**

Can we recover them before they churn?

Yes and effectively. This ensure it's done in a timely manner before they are completely lost and does so with thoughtful intent. Which is what customers are looking for.

Stakeholder Response

- **These steps are implemented for your business**
- **Data analysis is initially broken down manually for ensured refined and reproduction**

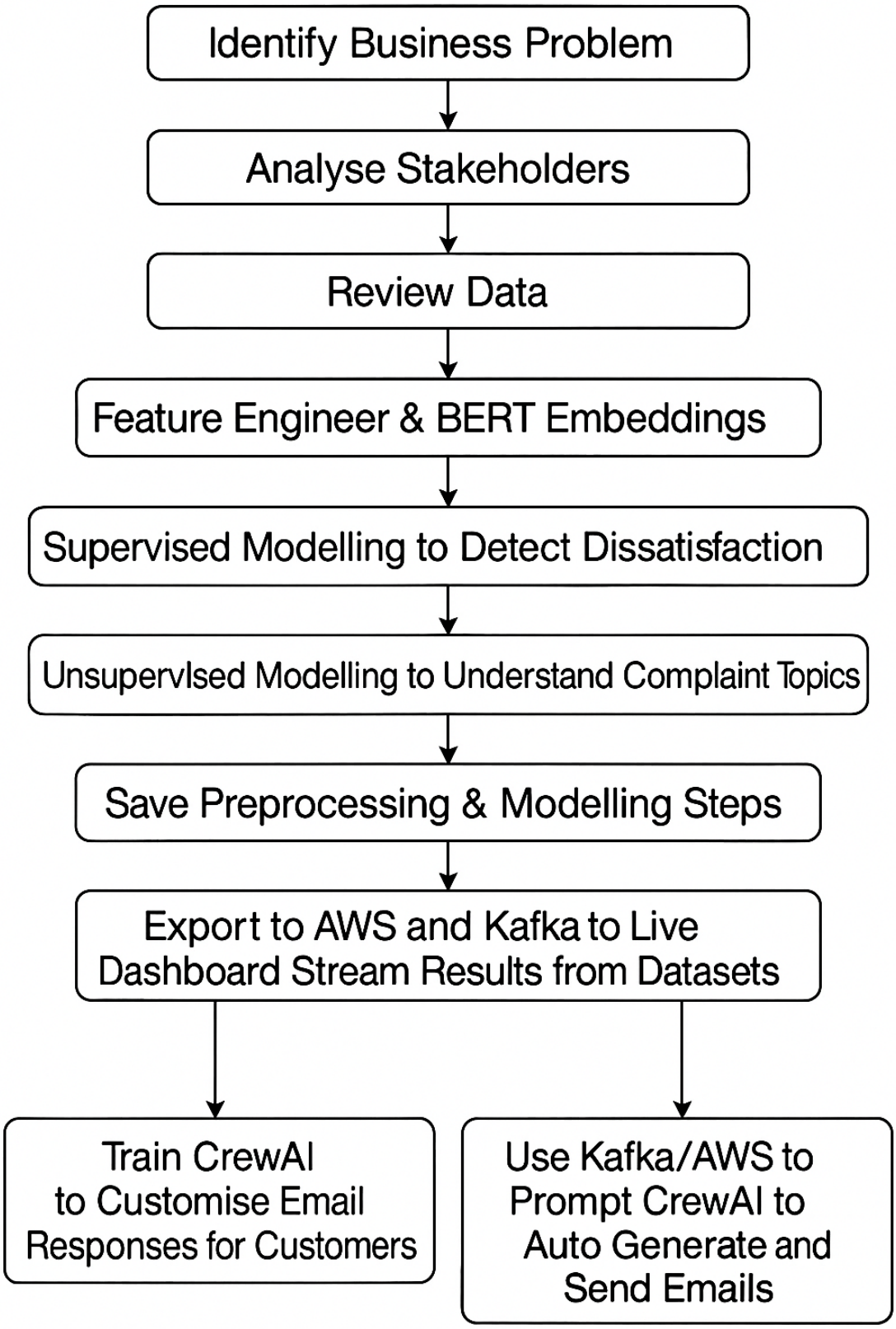
- **Detection modelling and dashboards are used**
- **Topic clustering is used to understand customers**
- **Email automation used such as mine**
- **Results closely monitored and edited until refined to satisfactory levels for your company**
- **Businesses then implement this pipeline system with the help of a data expert**
- **Customer service teams and managers are built appropriate information pipelines to ensure this enhances their work**
- **Progress is tracked and adjustments are made as required**

Summary

This project addresses the widespread issue of app customer churn and feedback neglect, highlighting how failed feedback loops are costing businesses substantial revenue. Drawing on real-world datasets, it delivers an end-to-end solution combining automated data insights, intelligent pipelines, and actionable responses.

The system is designed not just for the example company used in this case study, but is scalable and adaptable to any business operating in similar digital environments. It offers a replicable approach for companies looking to close the feedback loop, retain at-risk users, and turn dissatisfaction into recovery opportunities.

End to End Solution



References

Data Sourced from Mendeley: ShoppingAppReviews Dataset - Mendeley Data - https://data.mendeley.com/datasets/chr5b94c6y/2

Notebook Created through JupyterLab

Models Exported: run_full_processing.pkl, stacked_hybrid_model.pkl, CrewAI architecture from my JupLab

Statistical References:

🔁 **"Companies with strong feedback loops make 2–3× more revenue"**

Growett. (2023). *10 Customer Feedback Loops to Improve Revenue Generation*. Growett Blog. Retrieved from https://growett.com/blogs/10-Customer-Feedback-Loops-to-Improve-Revenue-Generation.html

---

📉 **"76% of businesses admit they don't know how to build effective feedback loops"**

CustomerThink. (2022). *How Continuous Improvement and Customer Feedback Drive Business Success*. Retrieved from https://customerthink.com/how-continuous-improvement-and-customer-feedback-drive-business-success/

🛍️ **"88% of consumers are more likely to buy from a brand that listens and responds"**

MarketingCharts. (2021). *Brand Metrics: Trust and Responsiveness*. Retrieved from https://www.marketingcharts.com/brand-related/brand-metrics-117111

🐶 **“Only 24% of consumers feel that brands actually listen”**

FinancesOnline. (2024). *Customer Loyalty Statistics*. Retrieved from https://financesonline.com/customer-loyalty-statistics/

♻️ **“95% of customers will give a second chance if complaints are resolved”**

University of Nottingham. (n.d.). *Complaint Handling Workbook*. Retrieved from https://training.nottingham.ac.uk/Public/Complaint-Handling-Workbook.pdf

---

🔁 **“70% of customers will return if issues are resolved in their favor”**

Short-Fact. (2021). *What Percentage of Dissatisfied Customers Will Return if Their Complaints Are Resolved?* Retrieved from https://short-fact.com/what-percentage-of-dissatisfied-customers-will-return-if-their-complaints-are-resolved/

📩 **“51% of marketers admit they target the wrong audience”**

RockContent. (2023). *Data-Driven Targeting and Personalization Challenges*. Retrieved from https://rockcontent.com/blog/data-wrong-targeting/

🚫 **“49% of consumers ignore brands that send irrelevant ads”**

Forbes. (2016). *Almost Half of Consumers Will Reject Brands Sending Irrelevant Ads*. Retrieved from https://www.forbes.com/sites/fionabriggs/2016/04/18/almost-half-of-consumers-will-reject-brands-sending-irrelevant-or-too-many-ads-study-shows

---

📲 **“46% of apps are uninstalled within 30 days”**

AppsFlyer. (2024). *App Uninstall Benchmarks Report*. Retrieved from https://www.appsflyer.com/resources/reports/app-uninstall-benchmarks/

📃 **“89% of app reviews go unanswered”**

Appbot. (2023). *Why Replying to Reviews Matters*. Retrieved from https://appbot.co/features/replies/

💰 **“Acquiring a new customer can cost 5 to 25 times more than retaining an existing one”**

Gallo, A. (2014). *The Value of Keeping the Right Customers*. Harvard Business Review. Retrieved from https://hbr.org/2014/10/the-value-of-keeping-the-right-customers

🧰 *Libraries & Tools*

- **pandas**
- **numpy**
- **matplotlib**
- **seaborn**
- **plotly**
- **dash**
- **re (regular expressions)**
- **os**
- **json**
- **datetime**
- **spacy**

- **nltk**
- **textblob**
- **langdetect**
- **sentence-transformers**
- **scikit-learn**
- **xgboost**
- **umap-learn**
- **hdbscan**
- **joblib**
- **kafka-python**
- **flask**
- **dotenv**
- **CrewAI**
- **langchain**
- **sendgrid**
- **smtplib**
- **Kafka**

🤖 *Algorithms & Models*

- **Logistic Regression**
- **Random Forest Classifier**
- **XGBoost Classifier**
- **Stacking Classifier (Ensemble Learning)**
- **Decision Trees (used internally by Random Forest)**
- **TF-IDF Vectorization**
- **BERT Embeddings (e.g., MiniLM or similar)**
- **UMAP (Uniform Manifold Approximation and Projection)**
- **KMeans Clustering**

- **HDBSCAN (Hierarchical Density-Based Spatial Clustering)**
- **Truncated SVD (for dimensionality reduction on sparse vectors)**
- **NMF (Non-negative Matrix Factorization) [if used]**
- **LDA (Latent Dirichlet Allocation) [if used]**

💉 *Evaluation & Metrics*

- **Accuracy**
- **Precision**
- **Recall**
- **F1 Score**
- **ROC-AUC**
- **Confusion Matrix**
- **Silhouette Score**
- **Davies-Bouldin Score**

🏵️ *Custom Feature Engineering*

- **Sentiment Polarity (via TextBlob)**
- **Review Length**
- **Capital Letter Ratio**
- **Adjective Count (POS tagging via spaCy)**
- **Presence of Negation Words**
- **Presence of Question Marks**
- **Word Count / Sentence Count**
- **Custom Sentiment Thresholding**