PROJECT DISPLAY

# Customer Query Classification for Banking Customer Support Using NLP

A Business Pitch for a Next-Level Customer Support Solution

**Presented By: Daniel Muruthi**

# AGENDA

- Introduction
- Business Understanding
- Data Understanding & Preparation
- Exploratory Data Analysis
- Machine Learning Modelling
- Baseline Model Results
- Conclusions (Classic Pipelines)
- Transition To Bert Modelling
- Model Performance Comparison
- Key Conclusions
- Recommendations For Improvement

# Introduction

BUSINESS PITCH

**Problem Statement**

Customers interact via chat/email for simple tasks (balance checks, lost cards, PIN resets)

**The Challenge**

Manual routing → slow responses, higher support costs

**Goal**

Automatically classify incoming queries into 77 banking intents

# BUSINESS UNDERSTANDING

- Build a robust intent-classification model for BANKING77 (77 fine-grained intents)
- Automate message routing to the correct support team
- **Expected Impact**

  Faster resolution, reduced manual workload, improved customer satisfaction

# DATA UNDERSTANDING

**Source :** PolyAI/BANKING77 on Hugging Face

**Features :**

Text :

- Raw customer query (string)
- Used as the model's input

Category :

- Intent label (string) from 77 possible classes
- Used as the model's target

**Total Records :** 13 083 (10 003 train, 3 080 test)

# PROJECT AIM & SUCCESS METRICS

**AIM**

Accurately classify user queries into one of 77 intents

**METRICS**

- Accuracy: Overall percentage of correct predictions
- Macro F1 Score: Balances precision & recall equally across all classes (crucial for rare intents)

# DATA PREPARATION

01

Loading both Train and test CSVs datasets → pandas DataFrames

02

Verify that each entry has a text and category field, with no missing or null values.

03

Ensure the data has no duplicates and is clean for modeling.

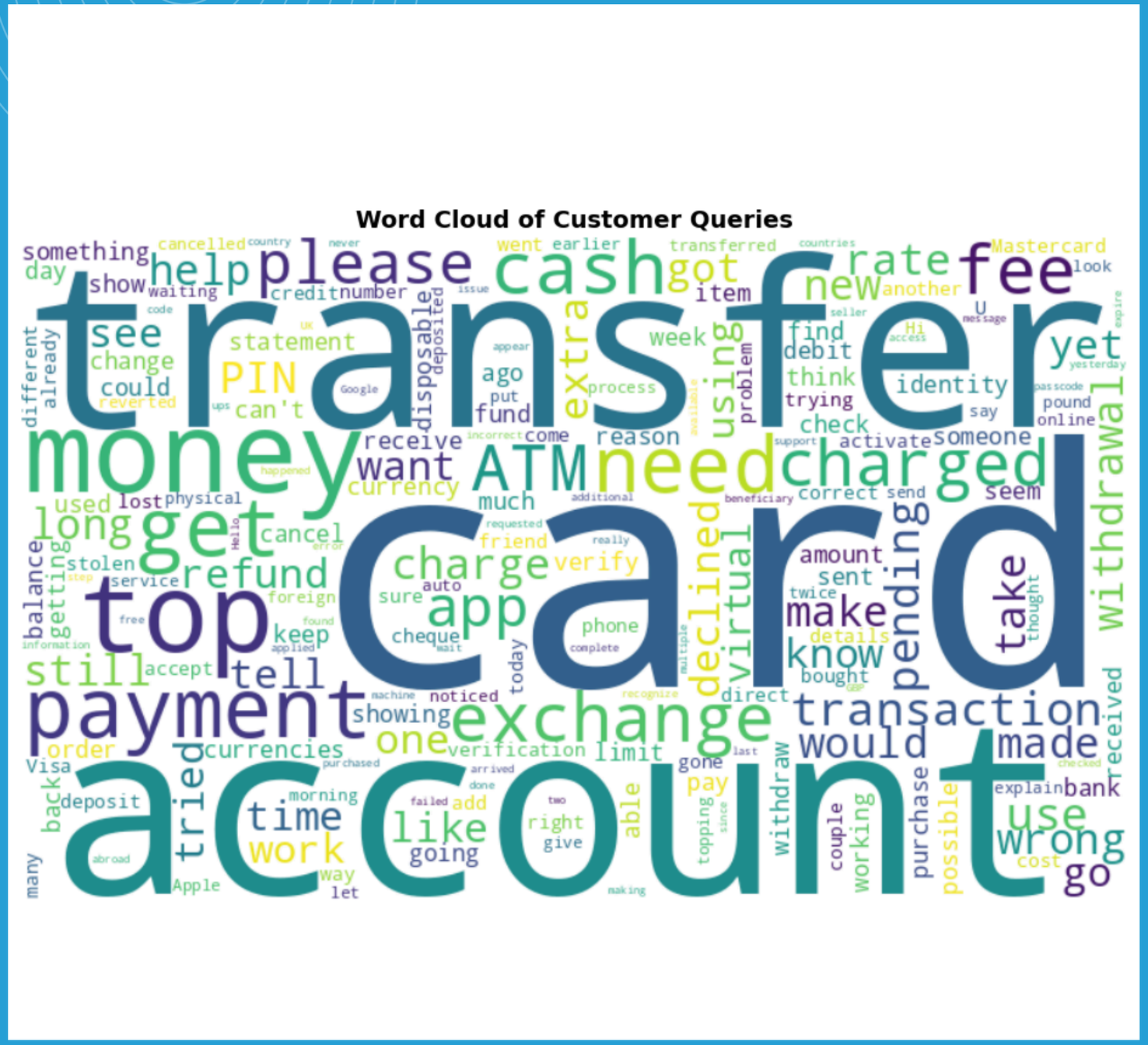
04

Confirm that the 77 intent categories are well-represented using a stratified train-test split.

# EXPLORATORY DATA ANALYSIS (EDA)

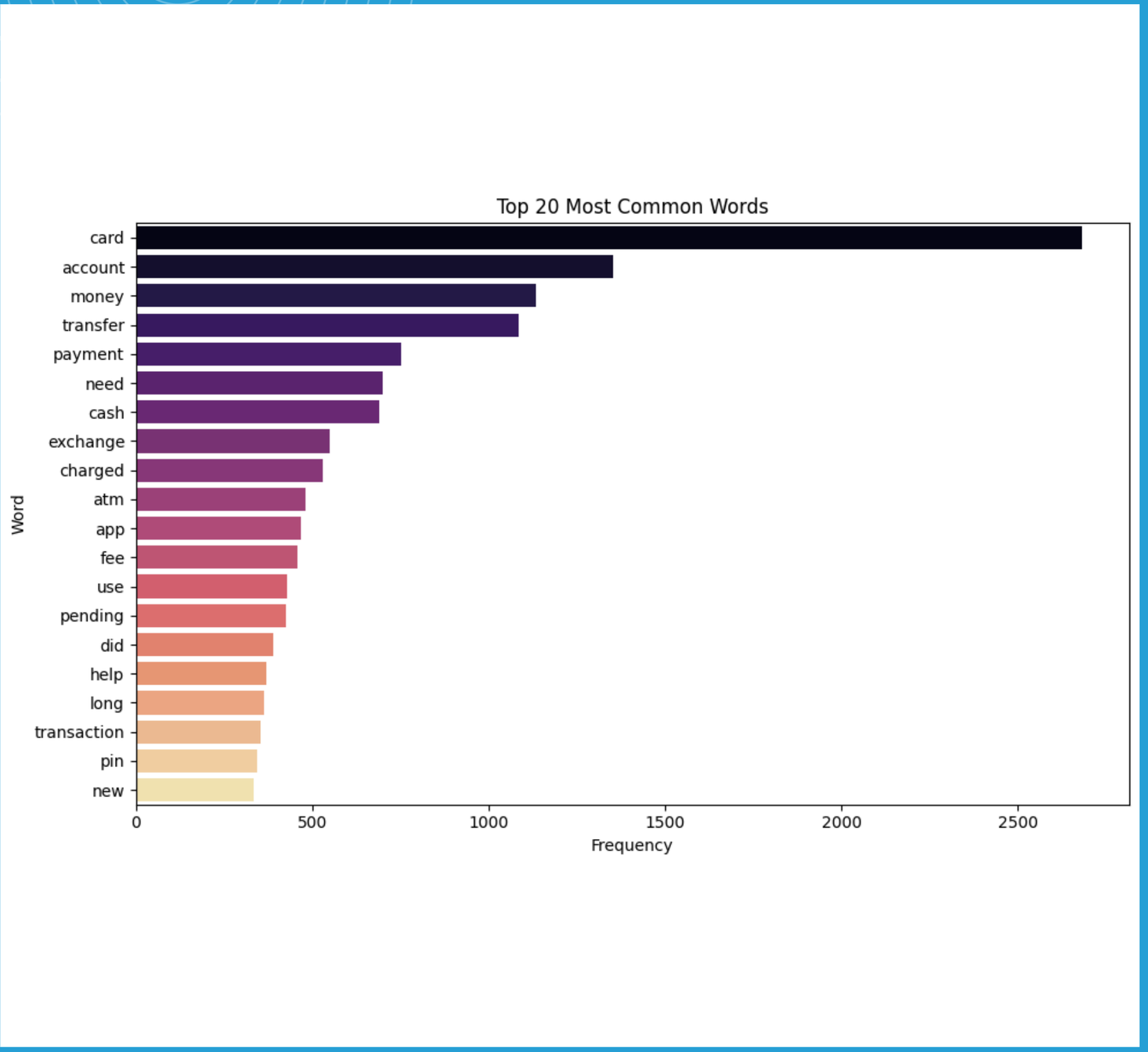
# Customer Banking Pain Points: What They're Asking About

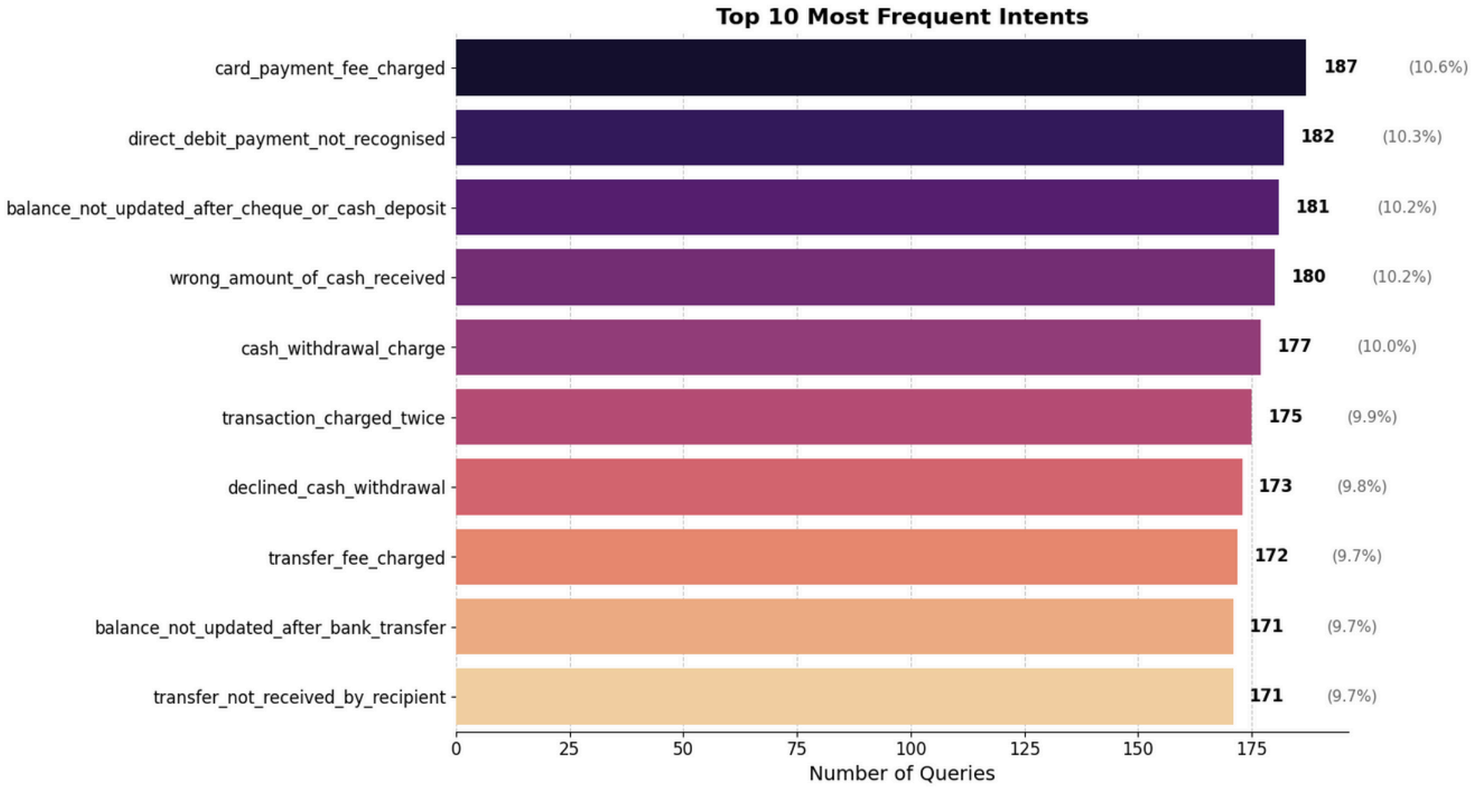
- This word cloud visualization reveals the most frequent terms in our customer query dataset. Key themes emerging include concerns about money transfers, card transactions, cash handling, fees, and account management issues. The prominence of terms like 'help,' 'please,' and 'need' highlights the urgency customers feel when reaching out about banking problems.



# Banking Customer Vocabulary: Word Frequency Breakdown

- This chart quantifies the exact terminology customers use when discussing their banking concerns. 'Card' dominates the conversation, appearing nearly twice as frequently as any other term, while 'account,' 'money,' and 'transfer' follow as key focus areas. The prominence of operational terms like 'ATM,' 'app,' and 'PIN' reflects customers' engagement with various banking touchpoints, highlighting where interface improvements could enhance customer experience.






# Top 10 Banking Issues Requiring Immediate Attention

- This analysis identifies the most common customer service intents driving support requests. Payment and transaction issues dominate customer concerns, with card fees, unrecognized payments, and balance update problems each accounting for over 10% of queries. These top issues represent critical areas where improved systems and customer communication could significantly enhance the banking experience.

# DATA PREPROCESSING

01 Train/Validation Split: Stratified 80/20

02 Text Cleaning: Lowercase, remove punctuation, regex filters

03 Vectorization: TF-IDF (unigrams to trigrams, min_df=2, max_df=0.95, sublinear_tf=True)

# MACHINE LEARNING MODELLING

**Pipeline Steps:**

- TF-IDF Vectorizer
- Classifier

**Baseline Classifiers:**

- Multinomial Naive Bayes
- Logistic Regression
- Linear SVC

# BASELINE MODEL RESULTS

| MODEL | ACCURACY | F1-SCORE |
|---|---|---|
| Multinomial Naive Bayes | 78.81% | 76.82% |
| Logistic Regression | 83.31% | 83.60% |
| Linear SVC (baseline) | 86.91% | 86.97% |
| Tuned Linear SVC (GridSearch) - TestSet | 88.57% | 88.56% |

- Linear SVC achieved the highest Macro F1 among classic pipelines
- LinearSVC Mode was further tuned

# CONCLUSIONS (CLASSIC PIPELINES)

TF-IDF + Linear SVC achieved 88% macro F1, accurately classifying 77 intents from short customer queries

Sublinear TF scaling and n-gram capture (1–3) helped extract meaningful phrases while reducing noise

Macro F1 tuning improved rare-intent prediction, ensuring balanced performance across all categories

# TRANSITION TO BERT MODELING

# WHY TRANSITION TO BERT?

01

After deploying the best GridSearch-optimized LinearSVC model, real-world performance was underwhelming despite strong validation metrics.

02

This led to experimenting with BERT, aiming to leverage its contextual language understanding for better intent classification.

03

However while BERT showed potential, limited data (10,003 rows) constrained its effectiveness — highlighting the need for more data to unlock its full power.

# MODEL PERFORMANCE COMPARISON

| MODEL | ACCURACY | F1-SCORE | COMMENTS |
|---|---|---|---|
| Multinomial Naive Bayes | 78.81% | 76.82% | Underperforms on infrequent classes |
| Logistic Regression | 83.31% | 83.60% | Faster inference with similar F1 to SVC |
| Linear SVC (baseline) | 86.91% | 86.97% | Strong classic baseline with TF-IDF features |
| Tuned Linear SVC (GridSearch) - TestSet | 88.57% | 88.56% | Performance improved through hyperparameter tuning |
| BERT (fine-tuned) | 90.55% | 90.47% | Highest overall performance; constrained by data size |

# KEY CONCLUSIONS

- Fine-tuning BERT yielded the highest performance, confirming that contextual language models can capture nuanced banking intents better.

- However, 10 003 training texts is quite small for fine-tuning a model of BERT's size, so results are still not "production-perfect." Scaling to at least 50 K – 100 K labeled examples (or augmenting via back-translation/weak labeling) should unlock BERT's full potential.

# RECOMMENDATIONS FOR IMPROVEMENT

## Increase the Training Corpus

- Augment the labeled dataset to at least 50 K – 100 K.

## Domain-Adaptive Pre-Training

- Pre-train on a large, banking-specific corpus to better align BERT's representations with domain terminology.

## Hyperparameter Tuning

- Experiment with different learning rates, batch sizes, dropout factors, and layer-freezing strategies to stabilize fine-tuning on limited data.

## Model Ensembling

- Combine predictions from complementary models (e.g., BERT + LinearSVC) using voting or stacking to leverage diverse feature representations.

## Robust Evaluation

- Employ stratified k-fold cross-validation to obtain more reliable performance estimates and mitigate variance on small datasets.

# Thank You

## Connect with me.

project repository

adinomuruthi1@gmail.com