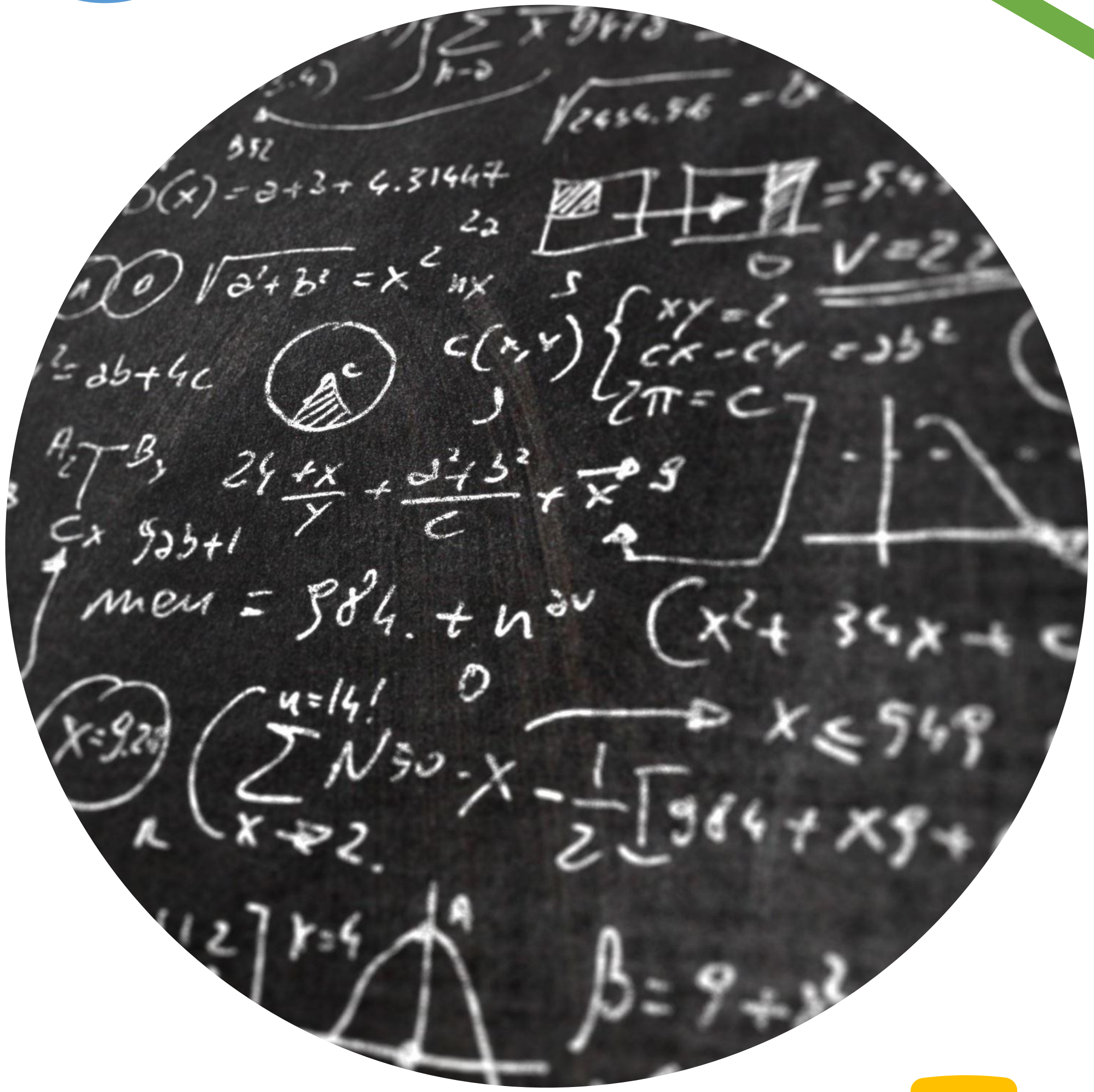


DSIR Project 3 : Comparing Models for Classification of Text Data Scraped From Reddit

Lavanya Acharya

4/1/2022

I'm a scientist studying machine learning models. I just started working with NLP and I'd like to pick the best model for my project. Here, I test a handful of different models on their efficiency in classifying human language.

Train

Train a handful of machine learning classifiers in their ability to classify text data into the appropriate class

Compare

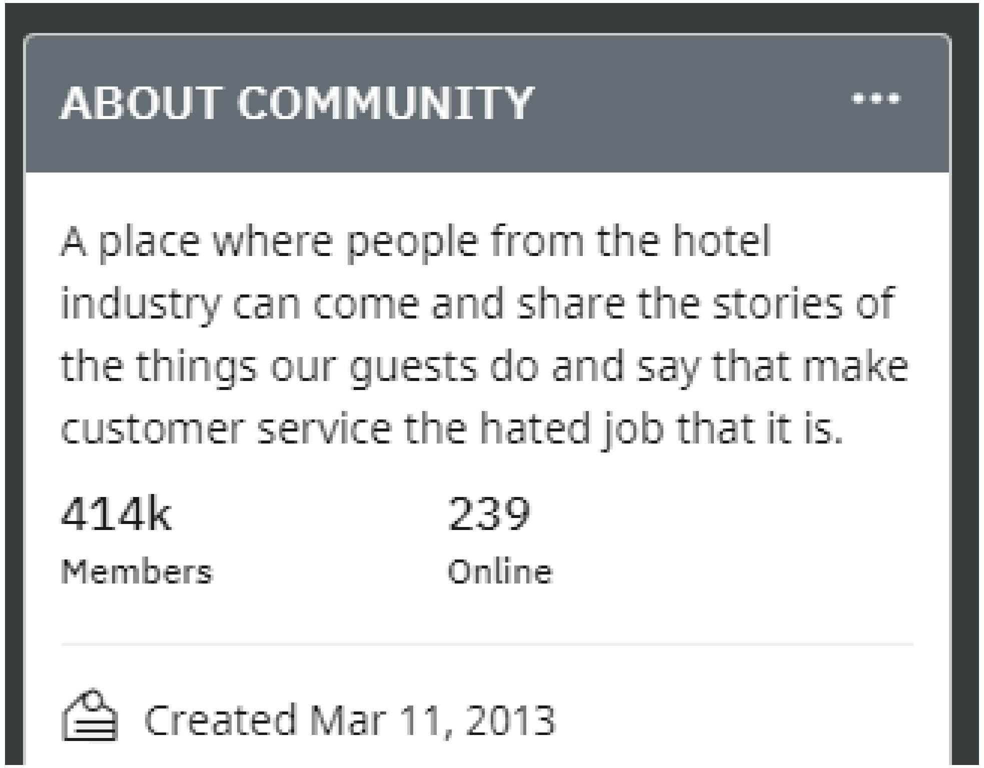
Compare performance metrics to find optimal models

Test and compare

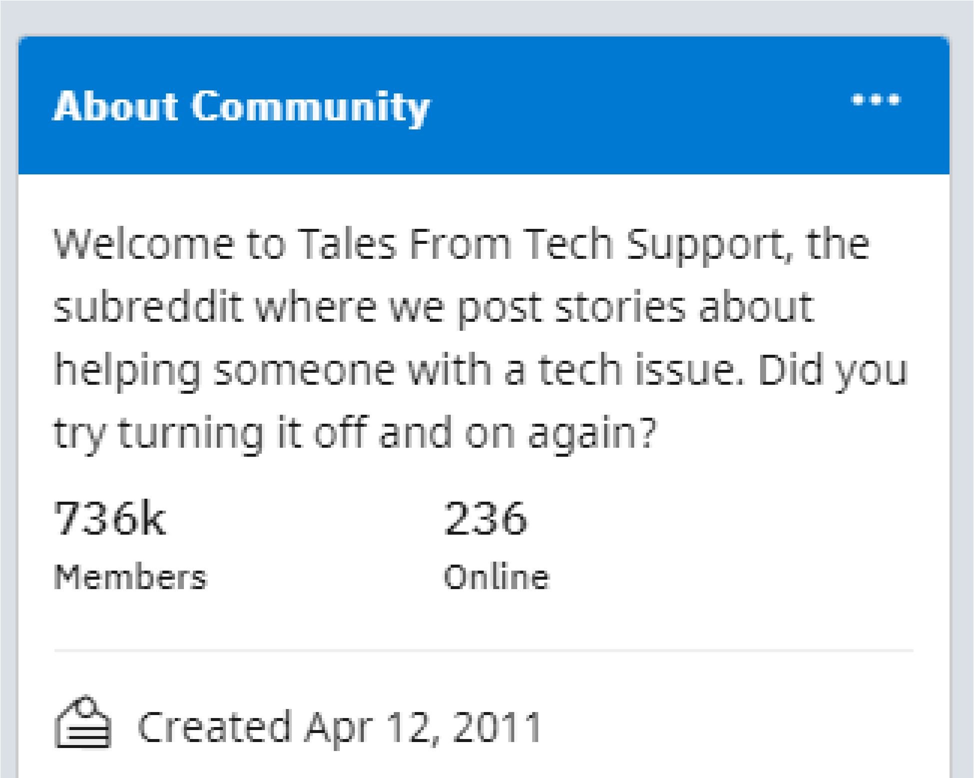
Test and compare optimal models with modified data

Data – Scraped from reddit using Pushshift API

r/TalesFromTheFrontDesk



r/talesfromtechsupport



Models used

Logistic Regression	Support Vector Machine	Multinomial Naïve Bayes	Voting Classifier
• Count Vectorizer • TF-IDF Vectorizer	• Count Vectorizer • TF-IDF Vectorizer	• Count Vectorizer	• Ada Boost Classifier • Gradient Boosting Classifier • Logistic Regression

Methodology – Data Acquisition and Cleaning

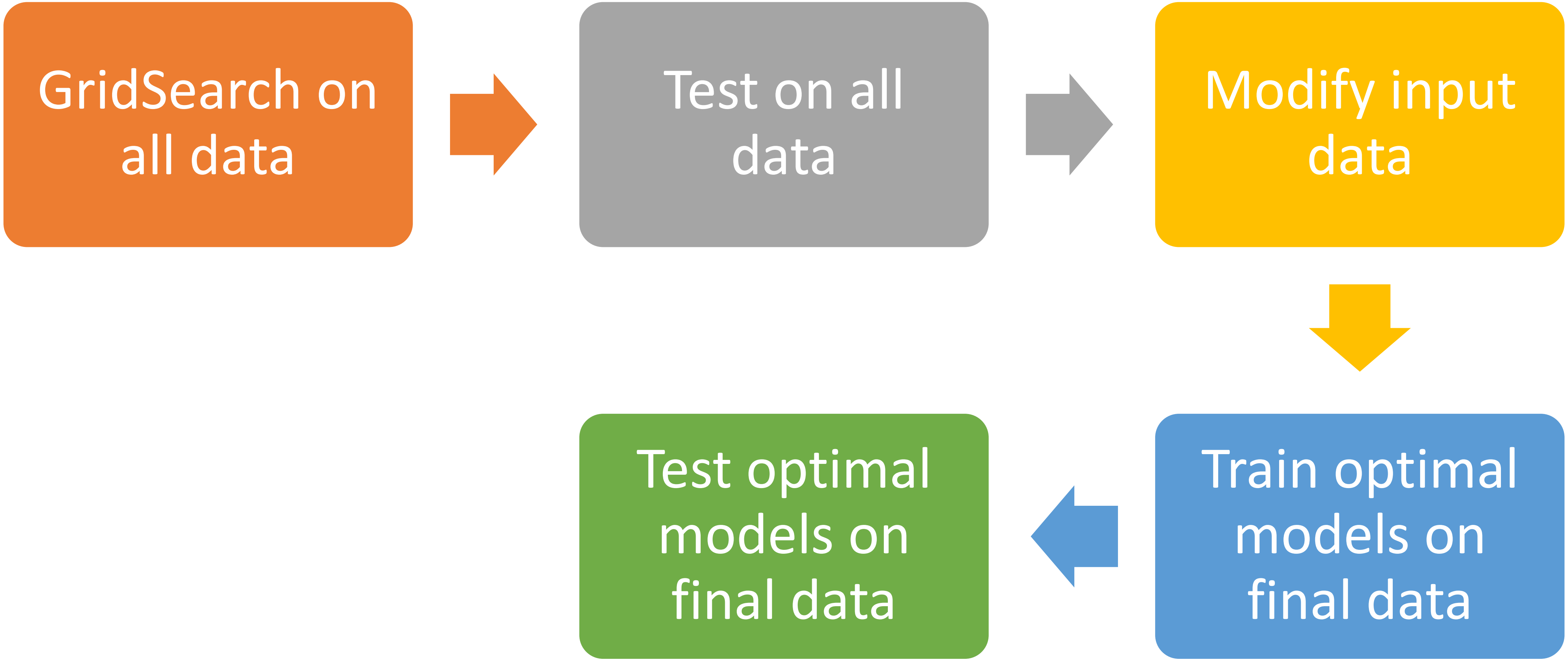
Set function to retrieve 5000 posts from each subreddit

Discarded [removed] or [deleted]

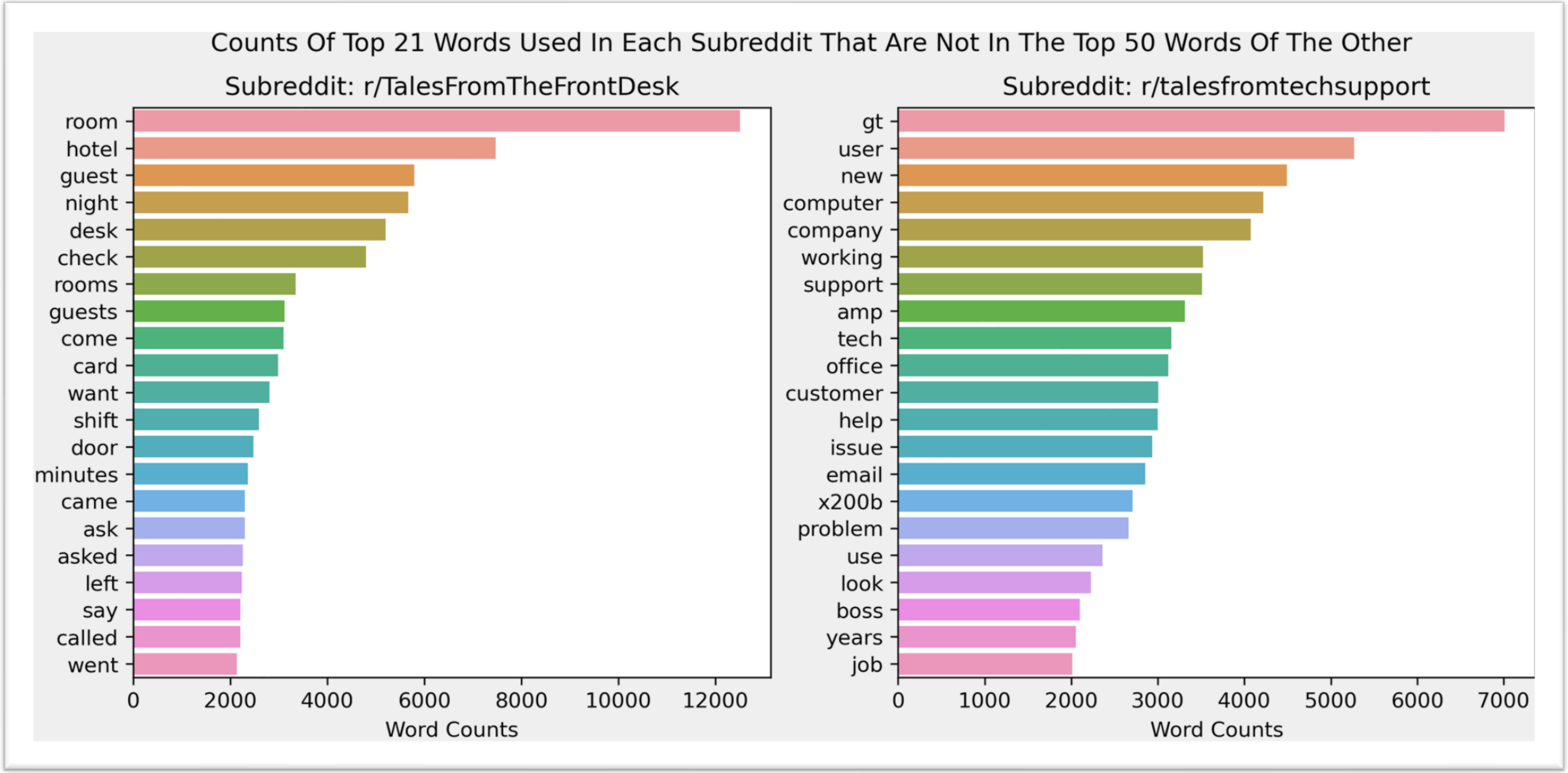
Discarded 1-word posts

4999 from FrontDesk, 4886 from TechSupport

Methodology – Training and Testing



Data modification – creating more overlap



Stop Words

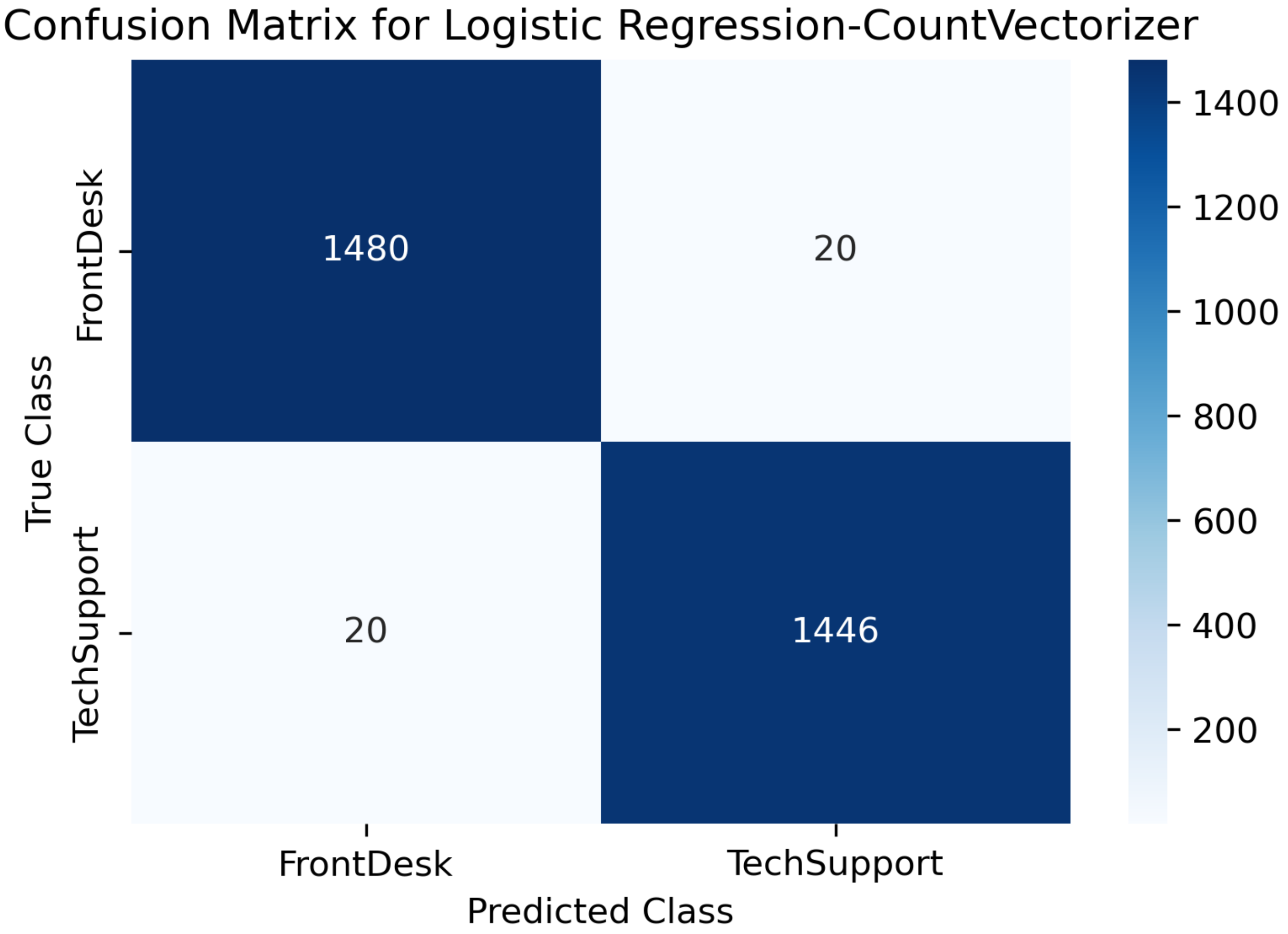
{'wherever', 'thin', 'hereby', 'another', 'every', 'indeed', 'though', 'formerly', 'you', 'out', 'guests', 'shift', 'because', 'is', 'has', 'nine', 'tech', 'afterwards', 'through', 'nobody', 'each', 'email', 'me', 'around', 'co', 'else', 'fifty', 'so', 'ask', 'whoever', 'get', 'in', 'gt', 'again', 'was', 'can', 'such', 'than', 'serious', 'myself', 'onto', 'been', 'more', 'already', 'give', 'working', 'here', 'an', 'meanwhile', 'ie', 'whereafter', 'card', 'they', 'un', 'whereupon', 'see', 'became', 'twelve', 'x200b', 'side', 'almost', 'few', 'eleven', 'amount', 'empty', 'seem', 'thru', 'my', 'forty', 'but', 'whether', 'years', 'becoming', 'toward', 'twenty', 'office', 'often', 'hereupon', 'into', 'either', 'always', 'might', 'new', 'less', 'boss', 'find', 'what', 'never', 'use', 'done', 'seemed', 'at', 'go', 'sometime', 'therefore', 'hotel', 'made', 'below', 'we', 'de', 'perhaps', 'well', 'it', 'fifteen', 'whence', 'am', 'whatever', 'too', 'will', 'least', 'mine', 'since', 'amongst', 'per', 'him', 'this', 'room', 'a', 'where', 'cry', 'could', 'ours', 'both', 'although', 'top', 'problem', 'everywhere', 'anywhere', 'seeming', 'night', 'very', 'move', 'couldnt', 'upon', 'describe', 'not', 'any', 'who', 'with', 'whither', 'the', 'nevertheless', 'put', 'via', 'thereafter', 'ltd', 'someone', 'say', 'while', 'why', 'come', 'across', 'rather', 'by', 'seems', 'mostly', 'to', 'first', 'something', 'name', 'himself', 'five', 'us', 'on', 'due', 'back', 'beforehand', 'except', 'of', 'computer', 'up', 'neither', 'herself', 'until', 'wherein', 'thereby', 'con', 'from', 'same', 'are', 'customer', 'full', 'throughout', 'went', 'take', 'hers', 'would', 'amoungst', 'somehow', 'ten', 'found', 'keep', 'thence', 'nothing', 'front', 'further', 'door', 'sincere', 'other', 'nowhere', 'some', 'its', 'noone', 'ever', 'whose', 'eg', 'become', 'between', 'still', 'whom', 'show', 'under', 'as', 'only', 'that', 'system', 'her', 'latter', 'during', 'together', 'mill', 'above', 'hundred', 'whereas', 'enough', 'had', 'those', 'much', 'whenever', 'minutes', 'there', 'alone', 'about', 'then', 'former', 'also', 'he', 'called', 'yet', 'amp', 'beyond', 'whole', 'ourselves', 'may', 'asked', 're', 'left', 'job', 'now', 'beside', 'most', 'she', 'sometimes', 'next', 'how', 'third', 'came', 'support', 'sixty', 'guest', 'bill', 'others', 'hence', 'four', 'nor', 'detail', 'moreover', 'off', 'interest', 'six', 'inc', 'hasnt', 'before', 'for', 'all', 'should', 'etc', 'yourself', 'were', 'everything', 'desk', 'last', 'yourselves', 'user', 'part', 'check', 'whereby', 'behind', 'after', 'three', 'them', 'becomes', 'these', 'i', 'therein', 'many', 'none', 'look', 'against', 'themselves', 'issue', 'help', 'when', 'anyone', 'towards', 'being', 'his', 'their', 'namely', 'besides', 'everyone', 'even', 'fire', 'several', 'thus', 'along', 'otherwise', 'however', 'one', 'down', 'cannot', 'be', 'rooms', 'anyway', 'company', 'no', 'thick', 'itself', 'somewhere', 'anything', 'and', 'without', 'want', 'please', 'your', 'elsewhere', 'yours', 'own', 'within', 'have', 'herein', 'anyhow', 'over', 'cant', 'must', 'two', 'fill', 'which', 'or', 'do', 'once', 'latterly', 'among', 'call', 'eight', 'thereupon', 'our', 'hereafter', 'if', 'bottom'}

Model 1b – *Logistic Regression with Count Vectorizer*

Grid Search Parameters :
- Count Vectorizer:
    - max_features': [None, 2500, 3000],
    - max_df' : [.8, .7, 1.0],
    - stop_words':['english', None],
- LogReg estimator:
    - C' : [1, 0.1, 0.01],
    - solver': ['liblinear'],
    - penalty': ['l1', 'l2’]

Best Parameters:
- 'cvec__max_df': 0.7,
- 'cvec__max_features': None,
- 'cvec__stop_words': None,
- 'model__C': 0.1,
- 'model__penalty': 'l2’,
- 'model__solver': 'liblinear’

Recall Score: 0.986



Model 1a – *Logistic Regression with Tfidf Vectorizer*

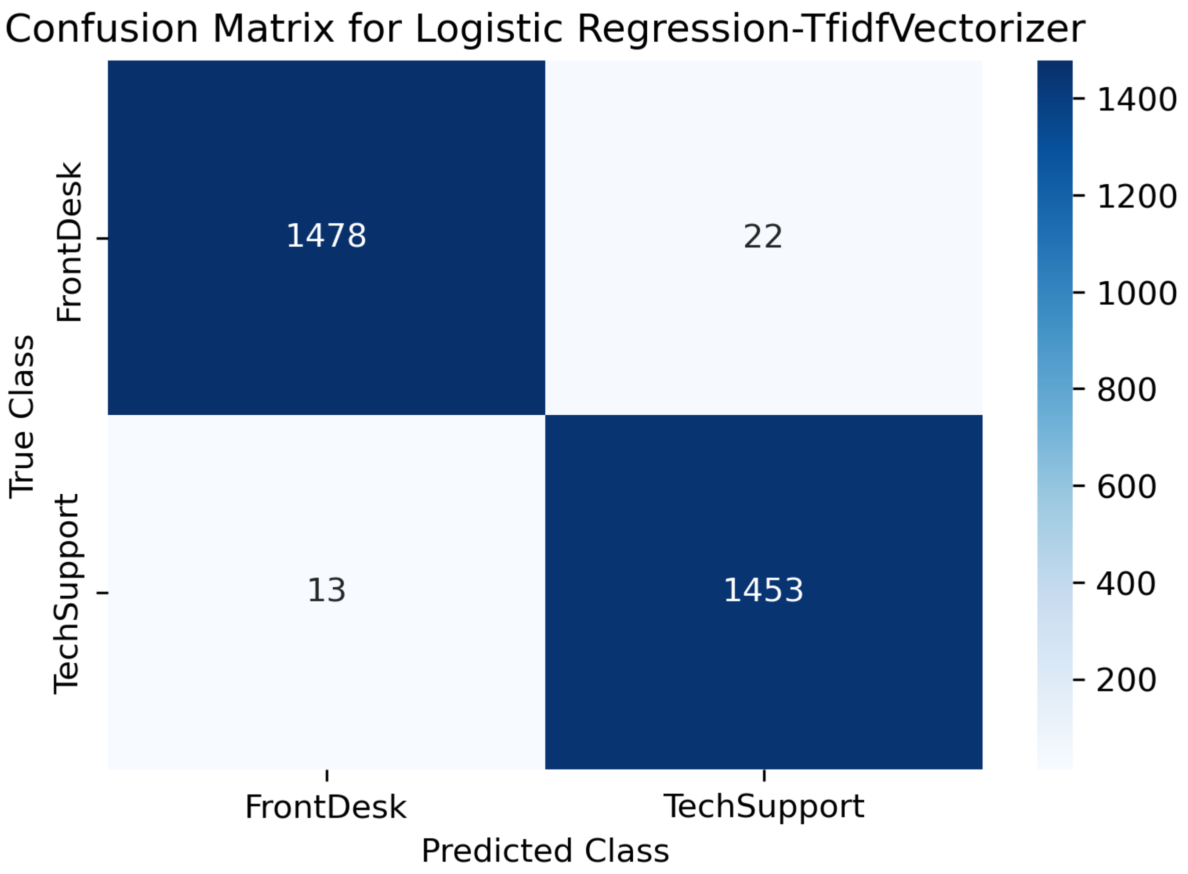
GridSearch Parameters

- 'tvec__max_features': [None, 2500, 3000],
- 'tvec__max_df': [.8, .7, 1.0],
- 'tvec__stop_words':['english', None],
- 'model__C' : [1, 0.1, 0.01],
- 'model__solver': ['liblinear'],
- 'model__penalty': ['l1', 'l2’]

Best Parameters

- {'model__C': 1,
- 'model__penalty': 'l2',
- 'model__solver': 'liblinear',
- 'tvec__max_df': 0.8,
- 'tvec__max_features': None,
- 'tvec__stop_words': 'english’}

Recall Score: 0.991



Model 2a – *Support Vector Machine with CountVectorizer*

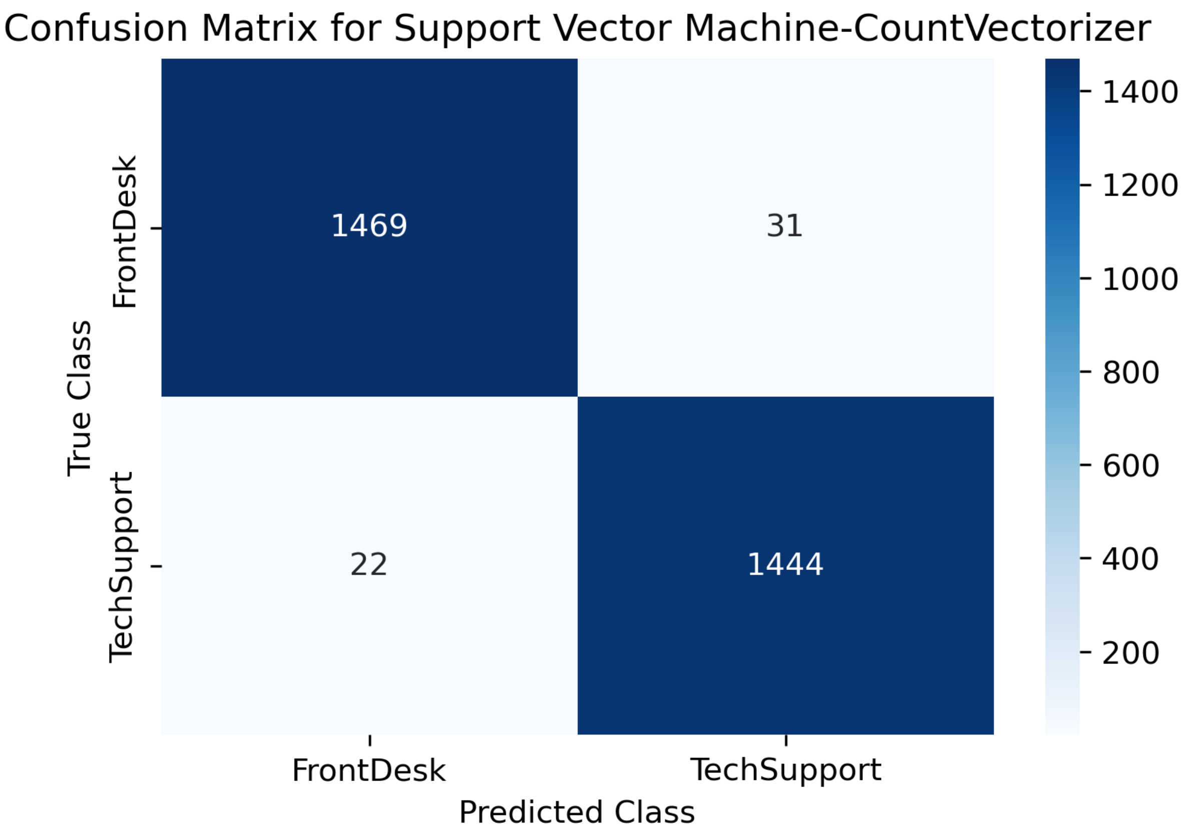
GridSearch Parameters

'cvec__max_features': [None, 2500, 3000, 3500],
'cvec__max_df': [.8, .7, 1.0],
'cvec__stop_words':['english', None],
'svm__kernel': ['poly', 'rbf', 'sigmoid']

Best Parameters

'cvec__max_df': 0.8,
'cvec__max_features': 2500,
'cvec__stop_words': None,
'svm__kernel': 'rbf'

Recall Score: 0.982



Model 2b – *Support Vector Machine with Tfidf Vectorizer*

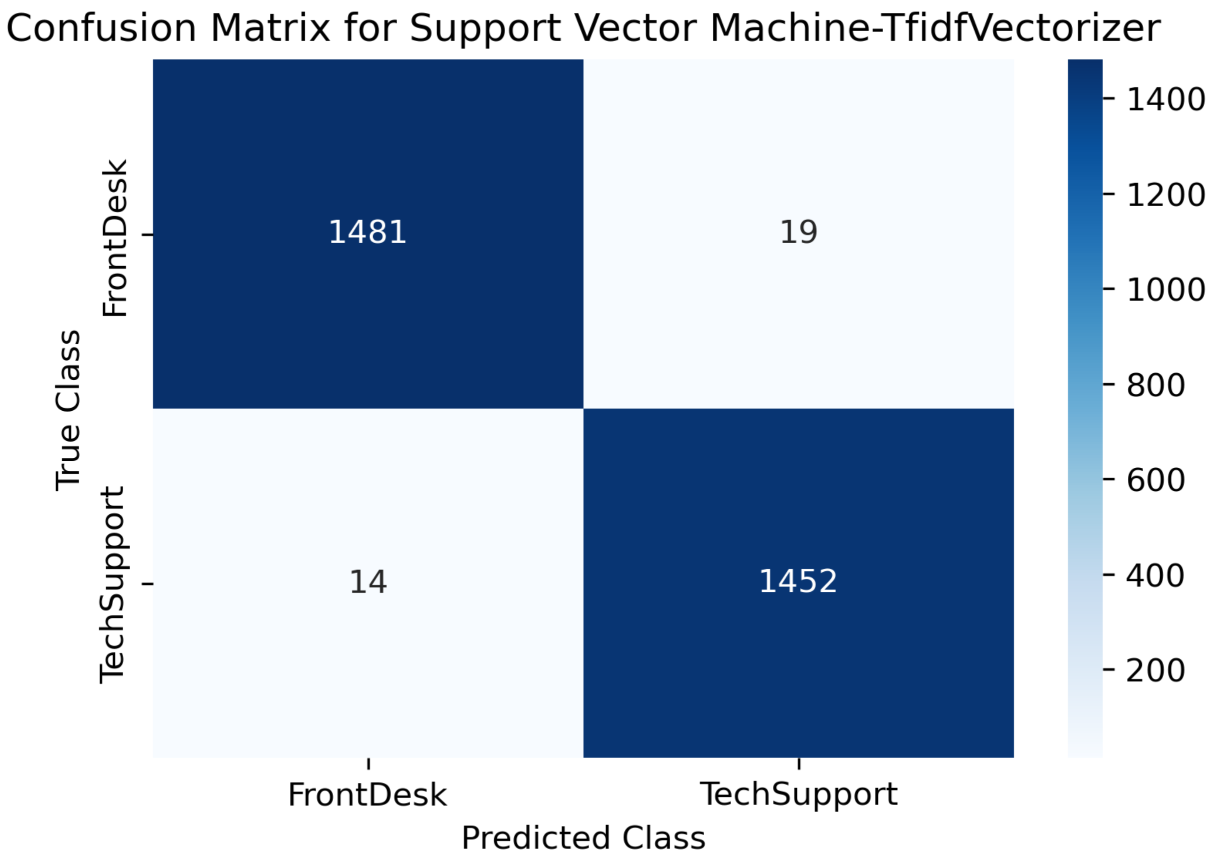
GridSearch parameters:

- 'tvec__max_features': [None, 2500, 3000, 3500],
- 'tvec__max_df': [.8, .7, 1.0],
- 'tvec__stop_words':['english', None],
- 'svm__kernel': ['poly', 'rbf', 'sigmoid’]

Best Parameters:

- svm__kernel': 'sigmoid',
- 'tvec__max_df': 0.7,
- 'tvec__max_features': None,
- 'tvec__stop_words': 'english’

Recall Score: 0.988

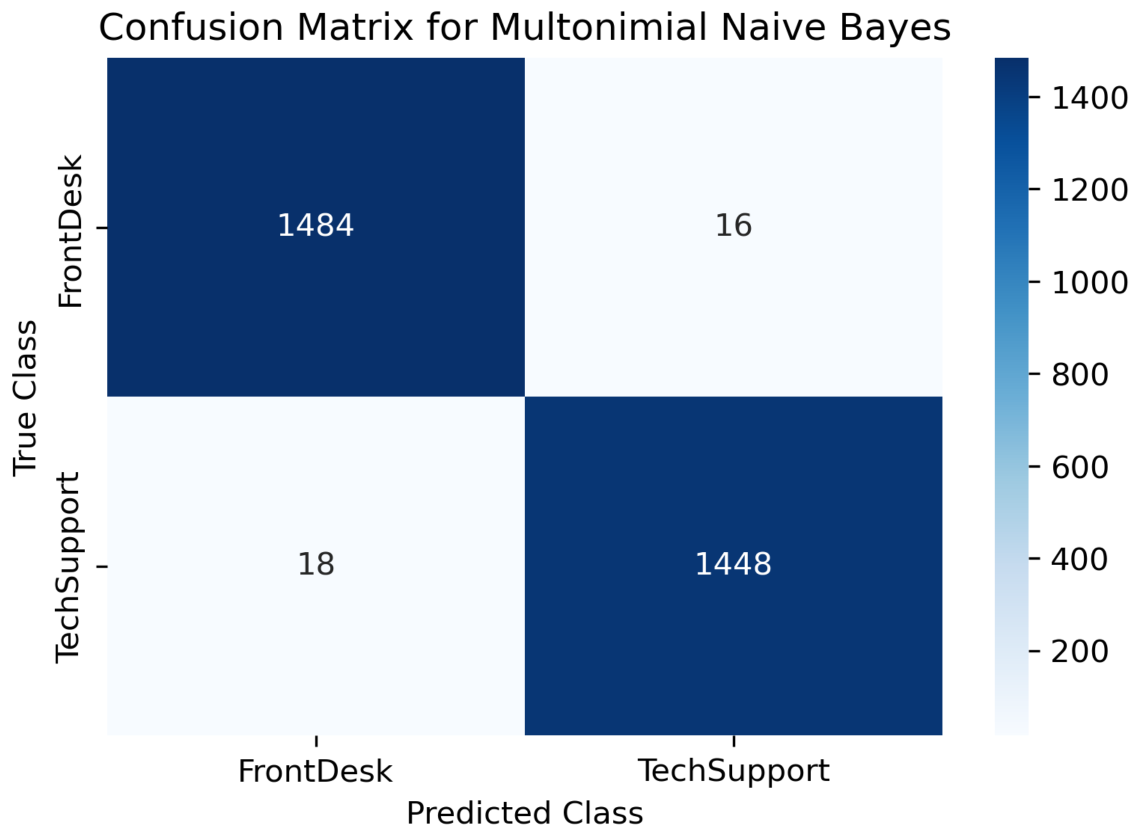


Model 3 – *Multinomial Naïve Bayes*

GridSearch parameters:
- 'cvec__max_features': [None, 2500, 3000, 3500],
- 'cvec__max_df': [.8, .7, 1.0],
- 'cvec__stop_words':['english', None],
- 'mnb__alpha': [0.001, 0.1, 1.0]

Best Parameters:
- 'cvec__max_df': 0.8,
- 'cvec__max_features': 3500,
- 'cvec__stop_words': 'english',
- 'mnb__alpha': 0.001

Recall Score: 0.988



Model 3 – *Voting Classifier*

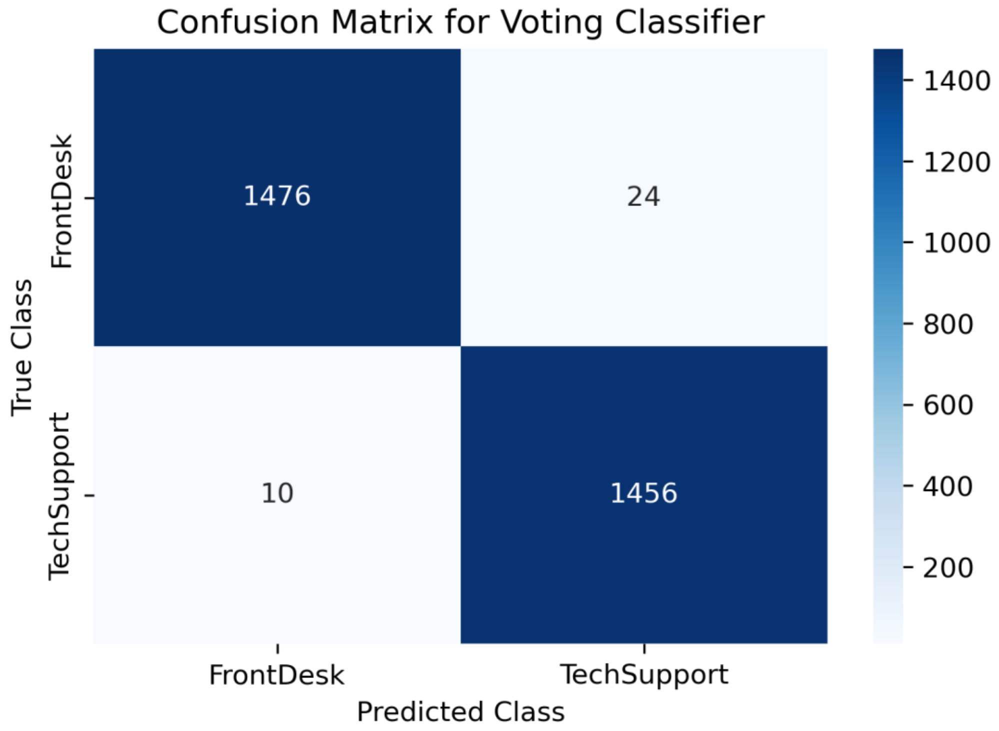
GridSearch parameters:

- 'cvec__max_features': [None, 2500, 3000],
- 'cvec__max_df': [.7, 1.0],
- 'cvec__stop_words': ['english', None],
- 'model__ada__base_estimator__max_depth': [3, 4,],
- 'model__ada__base_estimator__min_samples_split': [2, 5,],
- 'model__gbc__n_estimators':[100, 150]

Best Parameters:

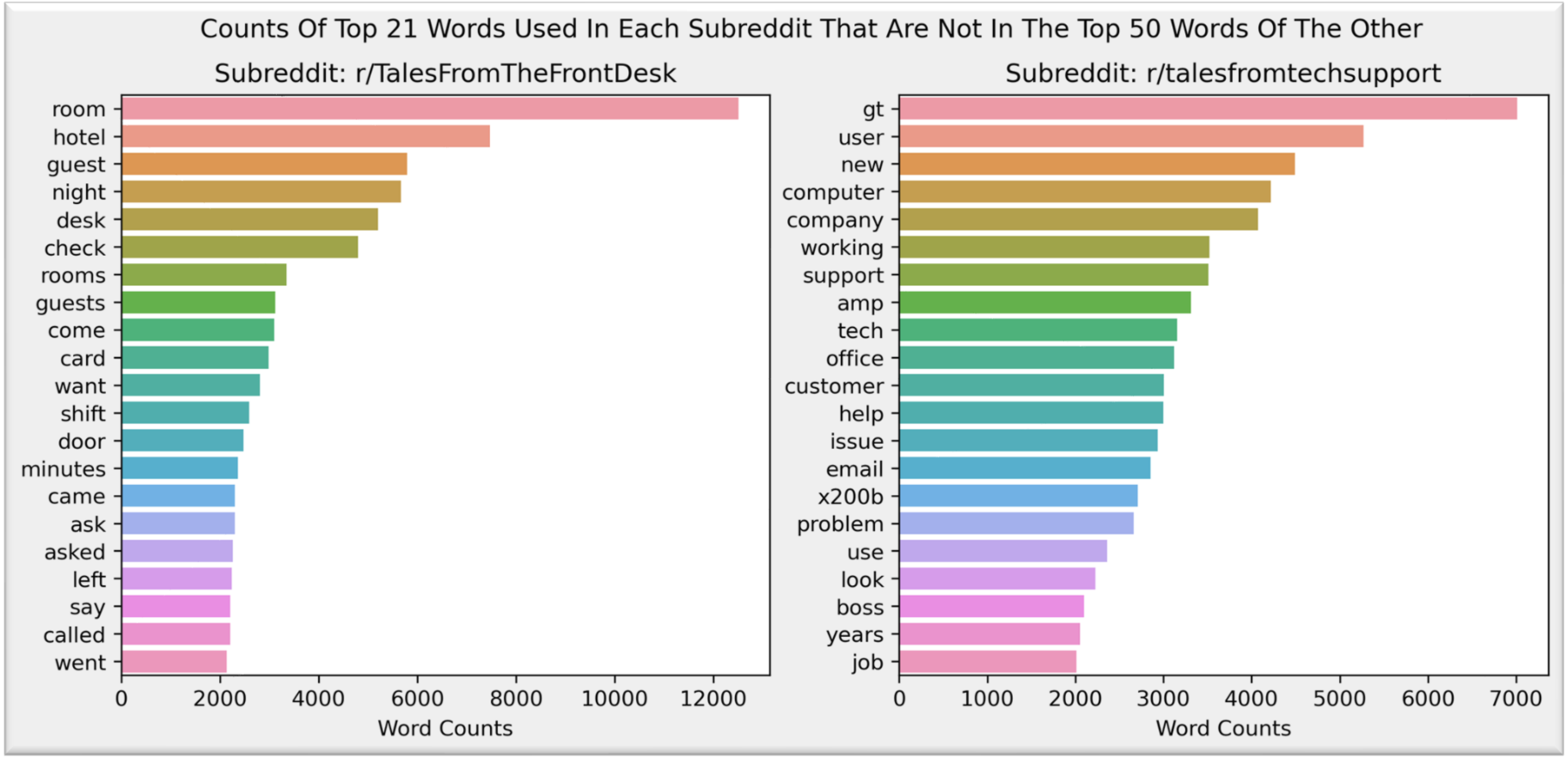
- 'cvec__max_df': 1.0,
- 'cvec__max_features': None,
- 'cvec__stop_words': None,
- 'model__ada__base_estimator__max_depth': 3,
- 'model__ada__base_estimator__min_samples_split': 5,
- 'model__gbc__n_estimators': 100

Recall Score: 0.988



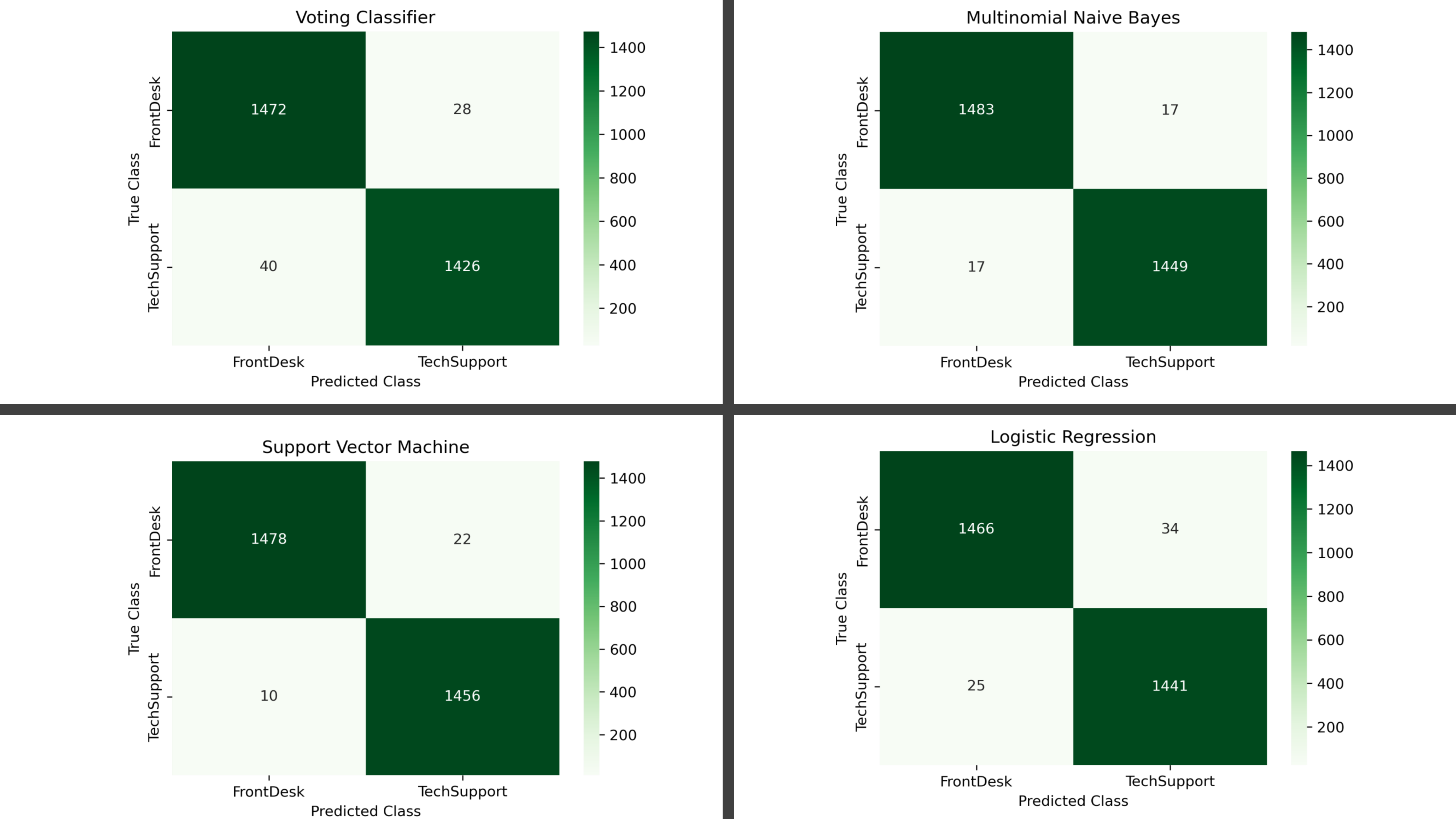
	Accuracy	Recall	Precision
SVM-TVec	0.988874	0.990450	0.987084
MNBayes	0.988537	0.987722	0.989071
VotingC	0.988537	0.993179	0.983784
LogReg-TVec	0.988200	0.991132	0.985085
LogReg-CVec	0.986514	0.986357	0.986357
SVM-CVec	0.982131	0.984993	0.978983

Results - Metrics



Final Models

Trained best models on Modified Data with unique key words removed



Voting Classifier
True Class
FrontDesk
TechSupport
1472
28
40
1426
Predicted Class
FrontDesk
TechSupport
200
400
600
800
1000
1200
1400
Multinomial Naive Bayes
True Class
FrontDesk
TechSupport
1483
17
17
1449
Predicted Class
FrontDesk
TechSupport
200
400
600
800
1000
1200
1400
Support Vector Machine
True Class
FrontDesk
TechSupport
1478
22
10
1456
Predicted Class
FrontDesk
TechSupport
200
400
600
800
1000
1200
1400
Logistic Regression
True Class
FrontDesk
TechSupport
1466
34
25
1441
Predicted Class
FrontDesk
TechSupport
200
400
600
800
1000
1200
1400

Results - Metrics

	Accuracy	Recall	Precision
SVM	0.989211	0.993179	0.985115
MNBayes	0.988537	0.988404	0.988404
LogReg	0.980108	0.982947	0.976949
VotingC	0.977073	0.972715	0.980743

Example of text all models got wrong

"Has anyone been on a LOTdodomu flight and could tell me how they handle luggage weight? You can add up to 5x23kg for free but I don't have any need for that many suitcases. I'm wondering if in this situation would they look past a 2-3kg overweight luggage?“

<u>A : Comes from TechSupport</u>

Conclusion

All four models did really well classifying posts into the appropriate subreddit

Support Vector Machine used with a TF-IDF vectorizer transformer has the best accuracy

Future Directions

- Remove more unique key words
- Train the best models on data from two other subreddits and see if they do just as well
- Create class imbalance and see how that affects performance

Thank you!

Questions?