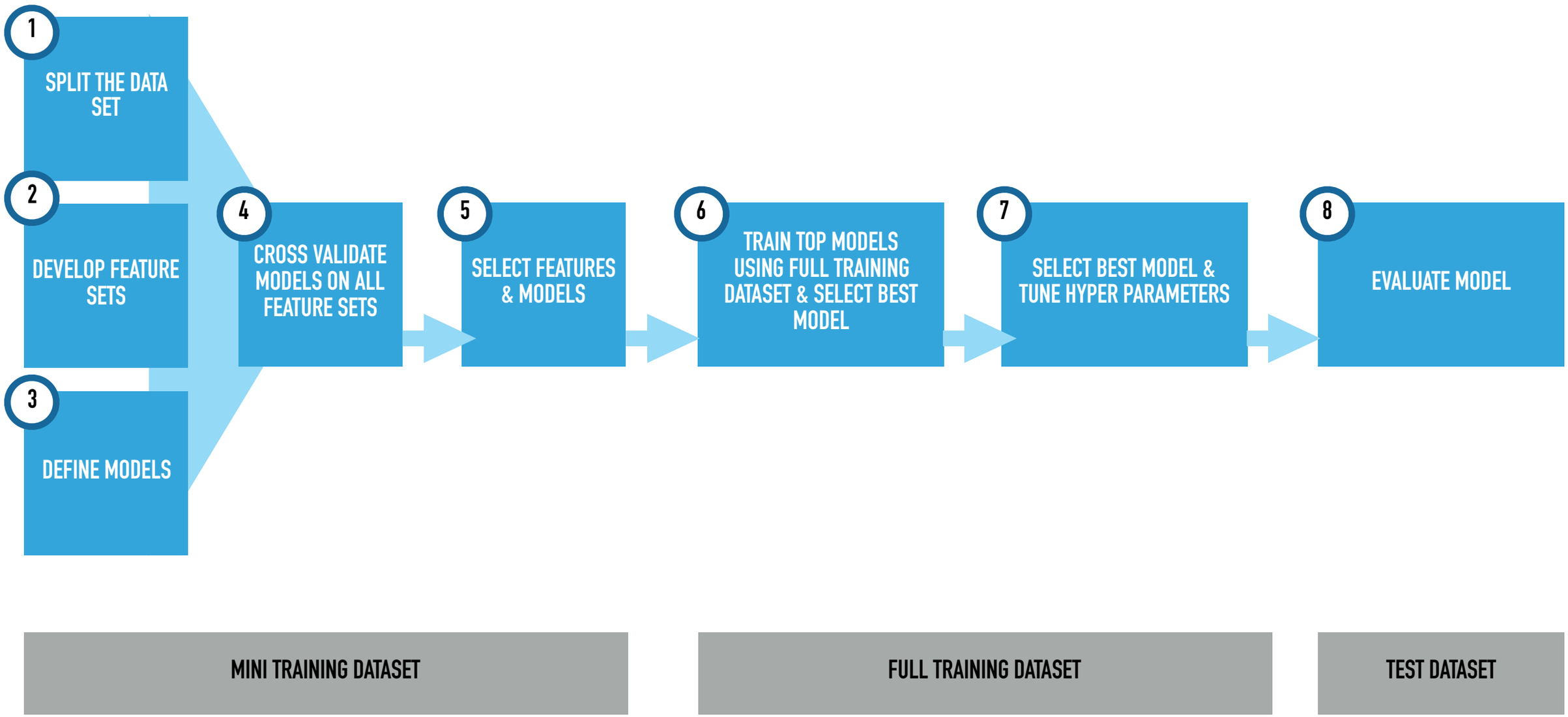
HELPING YELP

SPRING BOARD CAPSTONE PROJECT 1

FINAL REPORT

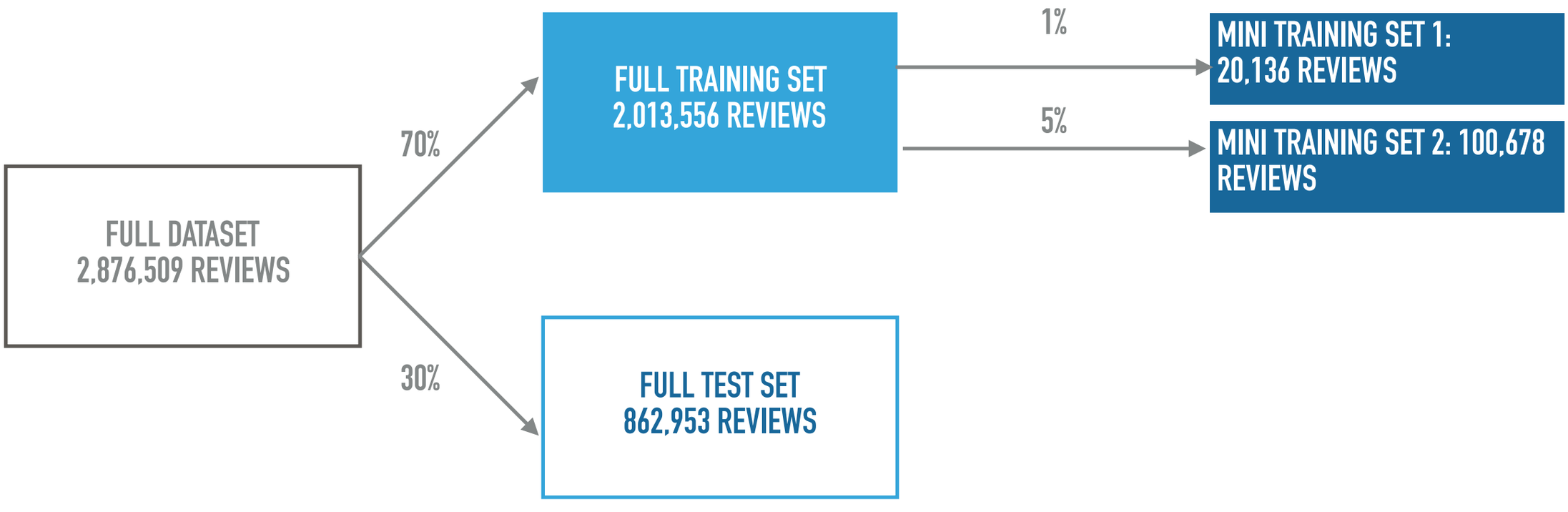
STEPS TAKEN


1
SPLIT THE DATA SET
2
DEVELOP FEATURE SETS
3
DEFINE MODELS
4
CROSS VALIDATE MODELS ON ALL FEATURE SETS
5
SELECT FEATURES & MODELS
6
TRAIN TOP MODELS USING FULL TRAINING DATASET & SELECT BEST MODEL
7
SELECT BEST MODEL & TUNE HYPER PARAMETERS
8
EVALUATE MODEL
MINI TRAINING DATASET
FULL TRAINING DATASET
TEST DATASET

NUMBER OF REVIEWS FOR MINI TRAINING SETS

A subset of the full training dataset was made for the purposes of selecting suitable feature sets and models in a manageable period of time. The size of the full dataset requires a substantial amount of time to train numerous models, cross-validate broad hyper parameters and test a variety of feature sets.

2 datasets of varying sizes were used so that the performance by different models and feature engineering techniques could be observed. This is to account for the model's varying tolerance to small datasets.



FEATURE SETS USED

BAG-OF-WORDS (BOW)

BASIC

A basic bag of words approach using scikit-learn's CountVectorizer was used, unigrams were the default setting.

CUSTOM STOP WORDS

The list of stop words used were updated based on the context of this project:

- neutral words related to restaurants to list of stop words (e.g. restaurant, place, bar, service, food, lunch, breakfast, dinner, price, order)
- stopwords that might reflect sentiment were removed (e.g. above, below, against).

NGRAMS

Unigrams, bigrams and trigrams were all included in the feature set. This almost triples the length of columns for the feature set.

TOPIC MODELLING

Latent Dirichlet Allocation (LDA) was used to model topics. 300 topics were selected.

TERM FREQUENCY–INVERSE DOCUMENT FREQUENCY (TFIDF)

BASIC

Scikit-learn's TfidfVectorizer helped weighs words by frequency and discounts words that are too frequent (adding to noise)

TOPIC MODELLING

Non-negative Matrix Factorization (NMF)¶ was used to model topics. 300 topics were selected.

WORD TO VECTOR

To account for the semantic differences between words, a word2vec model was developed based on the text.

100 FEATURES

A Word2Vec model with 100 features was created

200 FEATURES

A Word2Vec model with 200 features was created

1000 FEATURES

A Word2Vec model with 1000 features was created

GOOGLE NEWS

A Word2Vec model with trained on Google News created Google

MODELS USED FOR MULTI-CLASS CLASSIFICATION

NAIVE-BAYES

Scikit-learn's Naive-Bayes' model with alpha values of 0, 0.333, 0.666 and 0.999

LOGISTIC REGRESSION

Scikit-learn's logistic regression model with l1 and l2 type penalties and C values of 0.0001, 1, 100

SUPPORT VECTOR MACHINES (SVM)

Scikit-learn's SVM model with linear and rbf kernels

RANDOM FOREST

Scikit-learn's Random Forest model with n_estimators 150, 300, 500 and min_samples_leaf of 5, 10

STOCHASTIC GRADIENT DESCENT (SGD)

Scikit-learn's SGD model with penalty types 'l1', 'l2', 'elasticnet' and an l1_ratio 0.1, 0.3, 0.5

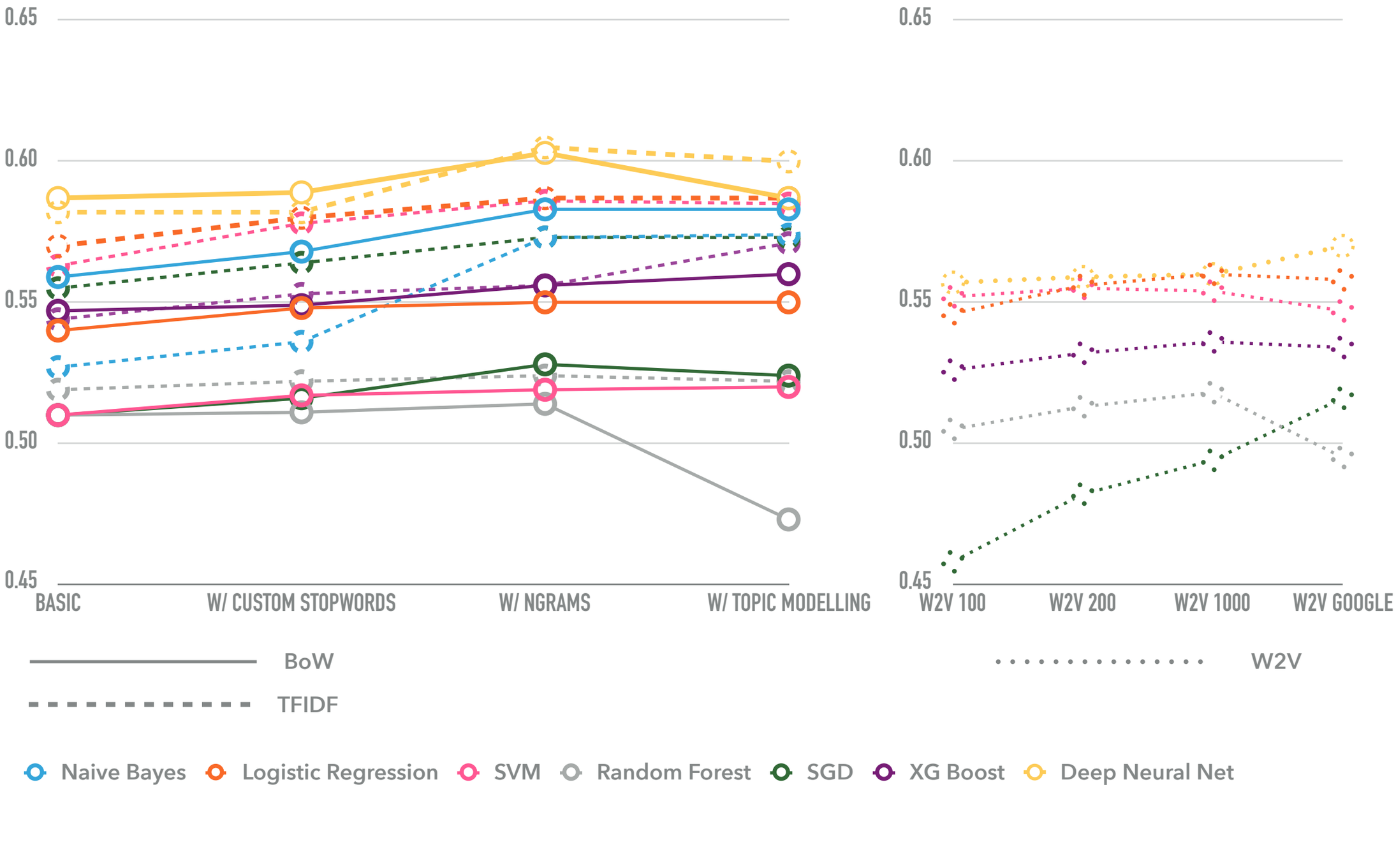
EXTREME GRADIENT BOOSTING (XGB)

xgboost's XGBoost model with min_child_weight of 3 and max_depth of 4

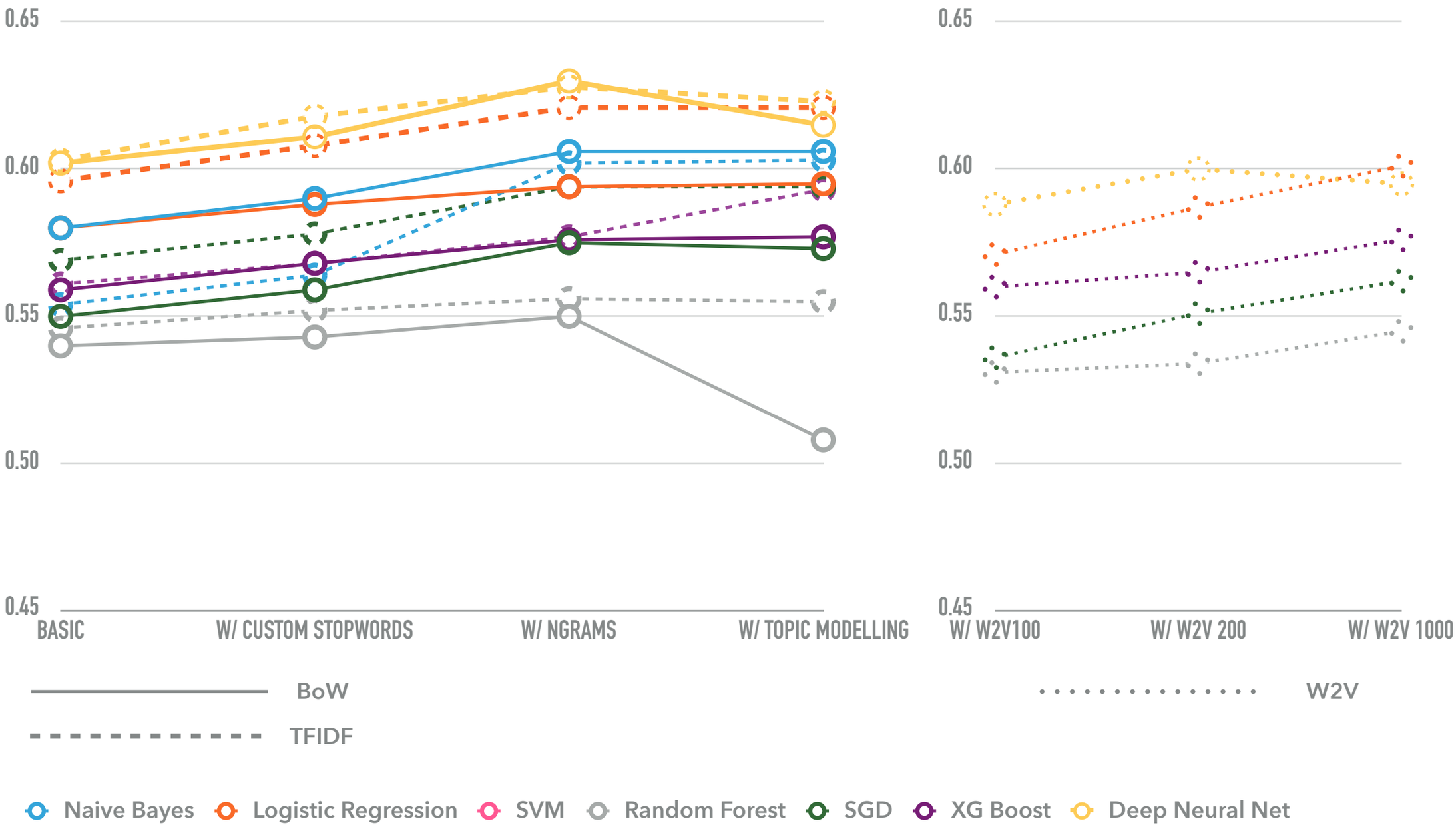
DEEP NEURAL NETS (DNN)

A 2 layer deep neural network with 512 nodes each on a relu activation. The model uses the adam optimiser and the categorical_crossentropy loss function

CROSS VALIDATION ACCURACY OF MODELS AND FEATURES (20K DATASET)



CROSS VALIDATION ACCURACY OF MODELS AND FEATURES (100K DATASET)



Note: SVM model was not trained for this data set as each cross validation took over 24 hours
GoogleNews word2vec not as 20k dataset did not suggest performance improvement to the w2v models trained on the Yelp dataset

FEATURE AND MODEL SELECTION

FEATURES

- Custom stop words showed a small improvement
- Unigram, bigram and trigram feature sets provided better performance for all models
- Topic modelling provided little to no improvement for all models (except XG boost's TFIDF)
- Word 2 Vector feature sets showed poorer performance vs ngrams feature sets for all models (except 100k logistic regression)

MODELS

- For 20k dataset, the top performing models in order, were: DNN (TFIDF ngrams), DNN (Count ngrams), logistic regression (TFIDF ngrams), SVM (TFIDF ngrams), Naive-Bayes (Count ngrams)
- For 100k dataset: DNN (Count ngrams), DNN (TFIDF ngrams), logistic regression (TFIDF ngrams), Naive-Bayes (Count ngrams), Naive-Bayes (TFIDF ngrams)

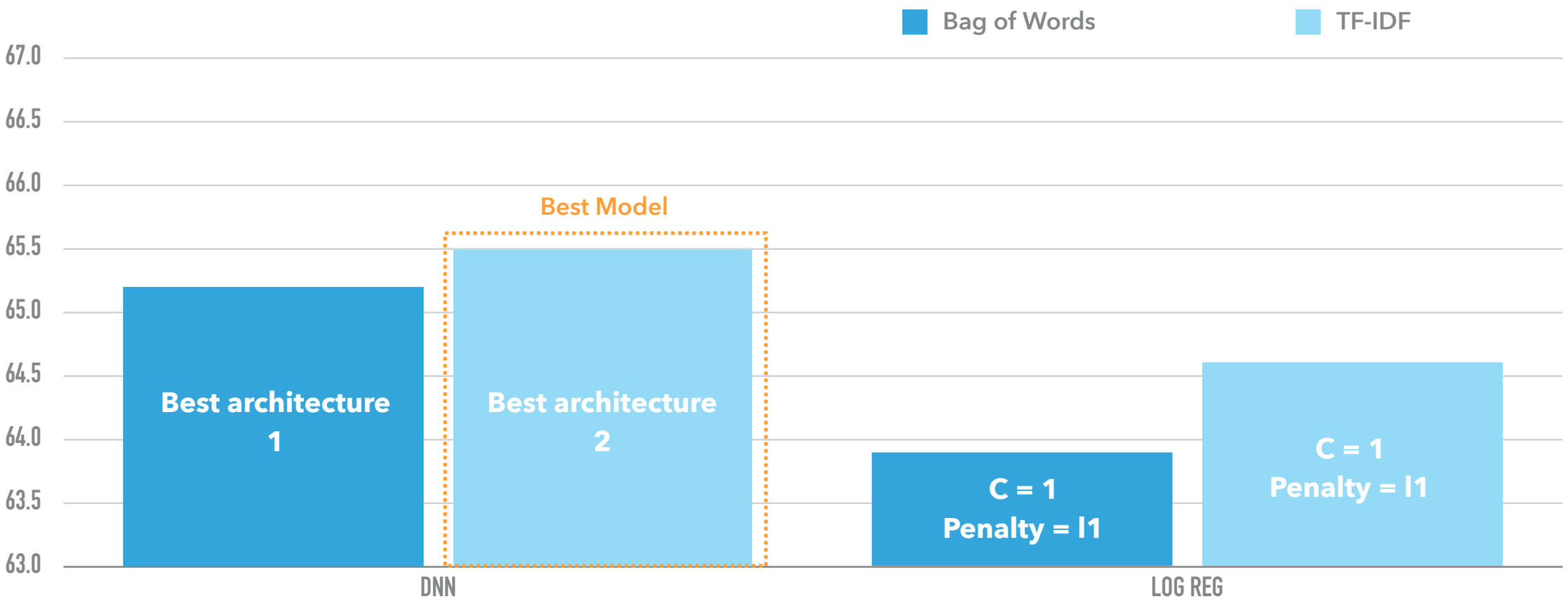
DATASET

- Without exception, all models trained on the 100k dataset showed noticeable improvement.
- The performance of the different models trained on the different feature sets did not show significant variance relative to each other when comparing the 20k and 100k datasets.
- The SVM model trained exceptionally slowly on the 100k dataset and had to be abandoned due to its inability to scale.

SELECTION

DNN & LOGISTIC REGRESSION TRAINED ON BAG OF WORDS AND TF-IDF WITH UNIGRAMS, BIGRAMS AND TRIGRAMS

TOP PERFORMING DNN AND LOGISTIC RECESSION RESULTS



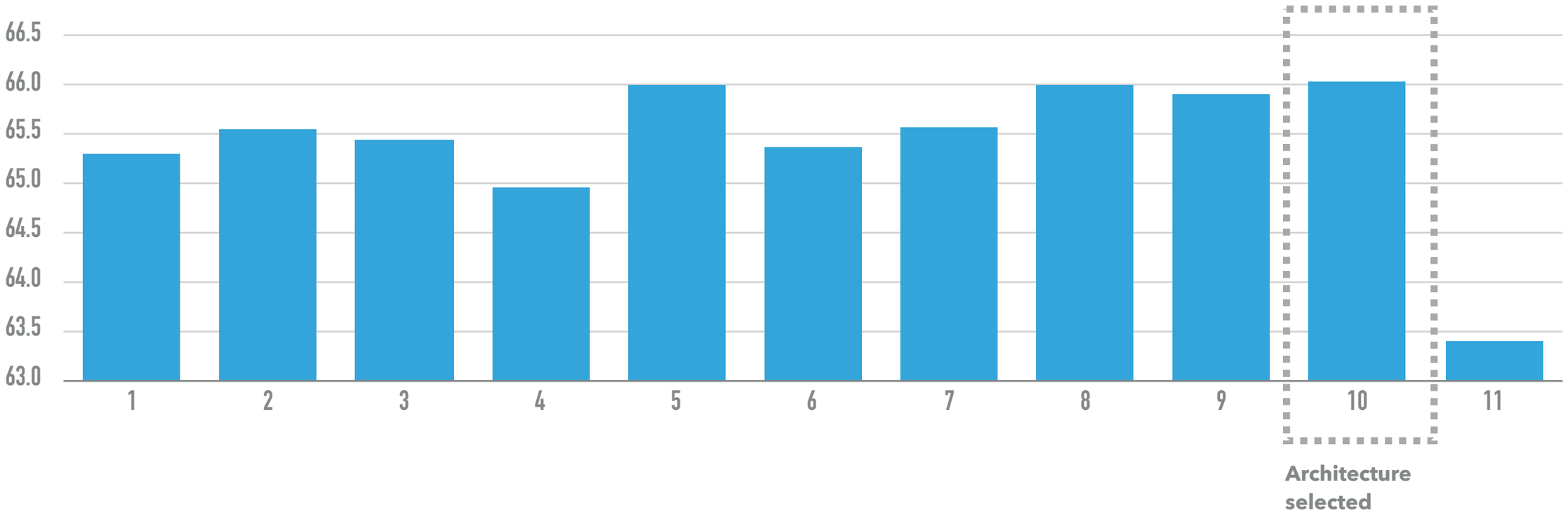
ARCHITECTURES

	Layers	Nodes per layer	Dropout	Activation
1	2	512	0.3	relu
2	2	256	0.3	relu
3	3	512	0.3	relu

Hyper-parameters used for training

- C: 0.0001, 0.01, 1, 100, 10000
- Penalty: l1, l2

TUNING OF HYPER PARAMETERS FOR DNN ON TF-IDF



Architectures used for training

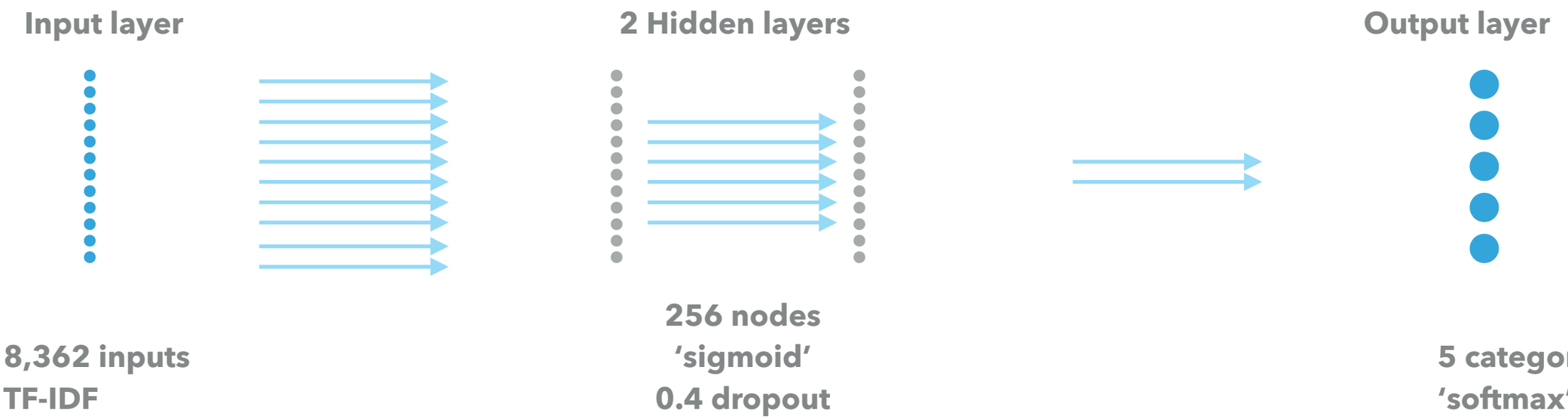
	Layers	Nodes per layer	Dropout	Activation	Optimizer
1	2	512	0.3	relu	adam
2	2	256	0.3	relu	adam
3	3	512	0.3	relu	adam
4	2	256	0.3	tanh	adam
5	2	256	0.3	sigmoid	adam
6	2	256	0.2	relu	adam

	Layers	Nodes per layer	Dropout	Activation	Optimizer
7	2	128	0.3	relu	adam
8	2	128	0.3	sigmoid	adam
9	2	128	0.4	sigmoid	adam
10	2	256	0.4	sigmoid	adam
11	2	256	0.4	sigmoid	sgd

Note: Final layer has an activation of 'softmax' with a loss function of 'categorical cross entropy'

MODEL PERFORMANCE USING TEST SET

FINAL DNN MODEL



CONFUSION MATRIX

	1	2	3	4	5
1	71, 488	18, 530	2, 942	1, 178	1, 382
2	17, 547	41, 503	19, 881	3, 957	1, 401
3	3, 783	16, 802	58, 523	37, 859	5, 749
4	1, 130	1, 994	19, 139	135, 826	82, 350
5	969	581	2, 129	52, 600	263, 710

AVERAGE WEIGHTED

ACCURACY: 66.2%
PRECISION: 65.5%
RECALL: 66.2%

APPENDIX

CROSS VALIDATION ACCURACY OF FEATURE SETS

			Basic	w/ custom stopwords	w/ ngrams	w/ topic modelling
20K DATASET	BAG OF WORDS	Naive Bayes	0.559	0.568	0.583	0.583
		Logistic Regression	0.540	0.548	0.550	0.550
		SVM	0.510	0.517	0.519	0.520
		Random Forest	0.510	0.511	0.514	0.473
		SGD	0.510	0.516	0.528	0.524
		XG Boost	0.547	0.549	0.556	0.560
		Deep Neural Net	0.587	0.589	0.603	0.587
	TFDIF	Naive Bayes	0.527	0.536	0.573	0.574
		Logistic Regression	0.570	0.580	0.587	0.587
		SVM	0.563	0.578	0.586	0.585
		Random Forest	0.519	0.522	0.524	0.522
		SGD	0.555	0.564	0.573	0.573
		XG Boost	0.544	0.553	0.556	0.571
		Deep Neural Net	0.582	0.582	0.605	0.600
100K DATASET	BAG OF WORDS	Naive Bayes	0.580	0.590	0.606	0.606
		Logistic Regression	0.580	0.588	0.594	0.595
		SVM				
		Random Forest	0.540	0.543	0.550	0.508
		SGD	0.550	0.559	0.575	0.573
		XG Boost	0.559	0.568	0.576	0.577
		Deep Neural Net	0.602	0.611	0.630	0.615
	TFDIF	Naive Bayes	0.554	0.564	0.602	0.603
		Logistic Regression	0.596	0.608	0.621	0.621
		SVM				
		Random Forest	0.546	0.552	0.556	0.555
		SGD	0.569	0.578	0.594	0.594
		XG Boost	0.561	0.568	0.577	0.593
		Deep Neural Net	0.603	0.618	0.628	0.623

CROSS VALIDATION ACCURACY OF WORD 2 VECTOR FEATURES

20K DATASET

	100 features	200 features	1000 features	GoogleNews
Logistic Regression	0.546	0.556	0.560	0.558
SVM	0.552	0.555	0.554	0.547
Random Forest	0.505	0.513	0.518	0.495
SGD	0.458	0.482	0.494	0.516
XG Boost	0.526	0.532	0.536	0.534
Deep Neural Net	0.557	0.559	0.560	0.570

100K DATASET

Logistic Regression	0.571	0.587	0.601	
SVM				
Random Forest	0.531	0.534	0.545	
SGD	0.536	0.551	0.562	
XG Boost	0.560	0.565	0.576	
Deep Neural Net	0.588	0.600	0.595	