Google Quest Q&A Classification

Author: Jennie Tram Le

Background

Machine Computable Dimensions:

- Number of Characters
- Punctuation Density
- Readability
- Entropy of POS Tags
- Question/Answer Overlap

Question:

Why is the sky blue?

Answer:

A clear cloudless day-time **sky** is **blue** because molecules in the air scatter **blue** light from the sun more than they scatter read light.

Human Computable Dimensions:

- Is the question's intent understood well?
- Is the question interesting?
- Is the question looking for factual information?
- Does the answer satisfy the question's intent?
- Is the answer helpful?
- Is the answer well-written?

- Computers are good at answering questions with single, verifiable answers
- Humans are better at subjective questions that require a multidimensional understanding of context

PROBLEM STATEMENT

What are we proposed to do ?

Build classification models for Google Q&A to:

1. Classify topic category for questions
2. Classify types for each question-answer pair

Compare different models to identify the most efficient subjective question-answering topics and types algorithms

Why is this important ?

Contribute to Q&A Systems research by making the system become more human-like.

GOOGLE QUEST Q&A LABELLING DATASET

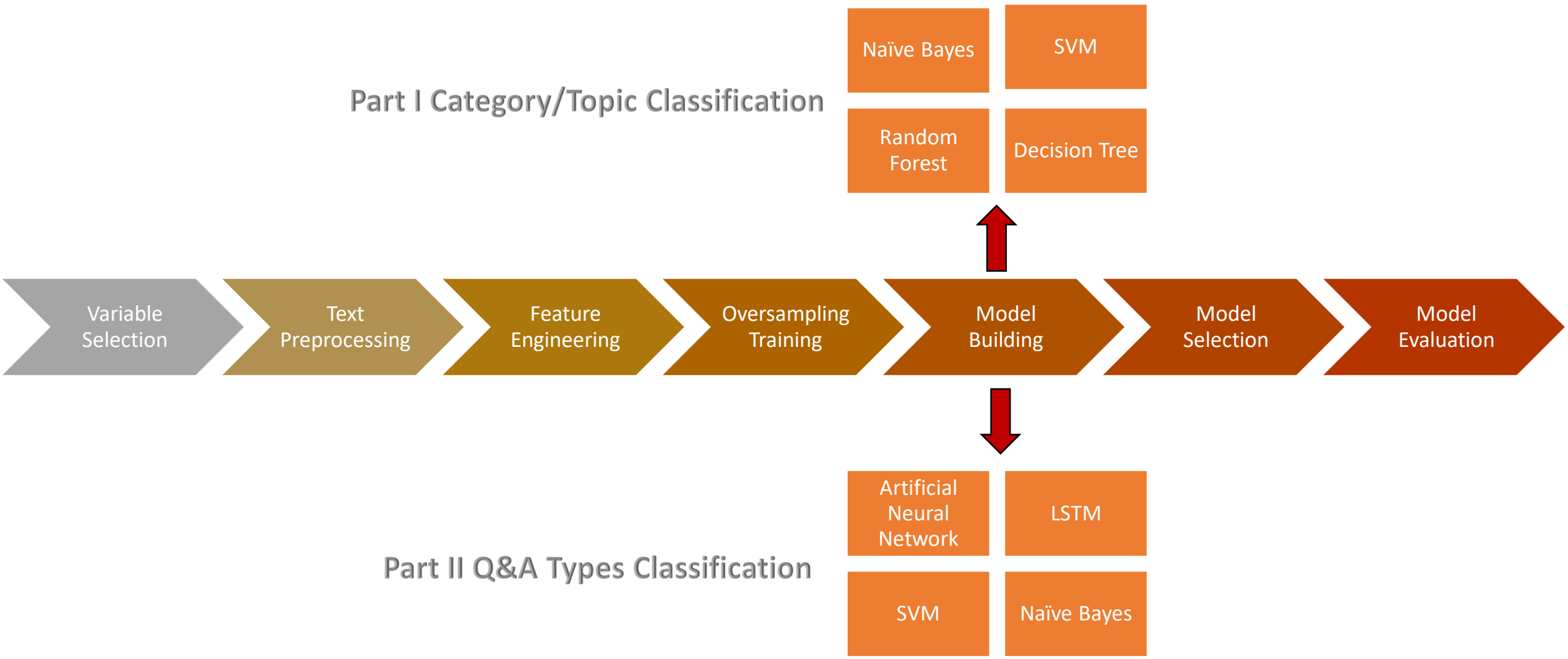
Data Collection

- Source: Kaggle Competition
- Collected by CrowdSource team at Google Research from various Stack Exchange properties
- The question-answer pairs were gathered from 70 websites

Data Description

- Format: One record contains one answer, one question, category, types, and other features (metadata) about the Q&A pair
- Train.csv: 6K pairs of question-answer (Answers with duplicate questions)
- Test.csv: 477 pairs of question-answer (No duplicated records)

METHODOLOGY DIAGRAM



VARIABLES SELECTION

- **Main inputs**
  - question_title
  - question_body
  - answer
- **Target**
  - category
  - question_type_etc
  - answer_type_etc

Q&A TYPE LABELING

- In the raw data, different values are assigned to each type for every question or answer. To make the data applicable to a classification model, we label the type based on the values.
- New target columns**: answer_target_mixed, question_target_mixed** – Target for Question & Type Classification

```
question_type_choice              0
question_type_compare             0
question_type_consequence         0
question_type_definition          0
question_type_entity              0
question_type_instructions        1
question_type_procedure           0
question_type_reason_explanation  0
question_type_spelling            0
```

"question_type_instructions"

```
question_type_choice              0.6666667
question_type_compare             0.6666667
question_type_consequence         0.0000000
question_type_definition          0.3333333
question_type_entity              0.0000000
question_type_instructions        0.0000000
question_type_procedure           0.0000000
question_type_reason_explanation  0.3333333
question_type_spelling            0.0000000
```

"question_type_mixed"

DATA PREPROCESSING

- Lowercase
- Stop-words removal
- Stemming
- Remove some tag, non-alphabet character

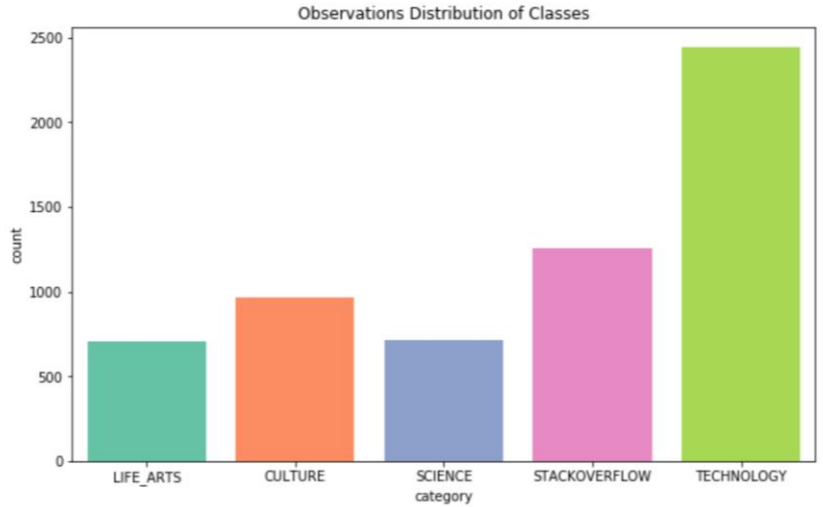
```python
from nltk.corpus import stopwords
# Define preprocessing function
def preprocessing_text(text):
    # Stopwords Removal
    my_stopwords = stopwords.words('english')
    # Stemming
    wnl = nltk.WordNetLemmatizer()
    tag_remove = re.compile(r'<[^>]+>')
    text = tag_remove.sub('',text)
    # tokenization
    tokens = nltk.word_tokenize(text)
    # alphabet / lowercase / stopwords removal
    words = [w.lower() for w in tokens if w.isalpha if w.lower() not in my_stopwords]
    stem = [wnl.lemmatize(w) for w in words]
    clean_text = ' '.join(w for w in stem)
    #clean_text = clean_text.replace("``","")
    return clean_text
```

```
"This is an issue that really limits my enjoyment of Linux. If the application isn't on a repository or if it doesn't have an
nstaller script, then I really struggle where and how to install an application from source.\n\nComparatively to Windows, it'
easy. You're (pretty much) required to use an installer application that does all of the work in a Wizard. With Linux... not
much.\n\nSo, do you have any tips or instructions on this or are there any websites that explicitly explain how, why and when
to install Linux programs from source? \n"
```

```
"issue really limit enjoyment linux . application n't repository n't installer script , really struggle install application sou
rce . comparatively window , 's easy . 're ( pretty much ) required use installer application work wizard . linux ... much . ,
tip instruction website explicitly explain , install linux program source ?"
```

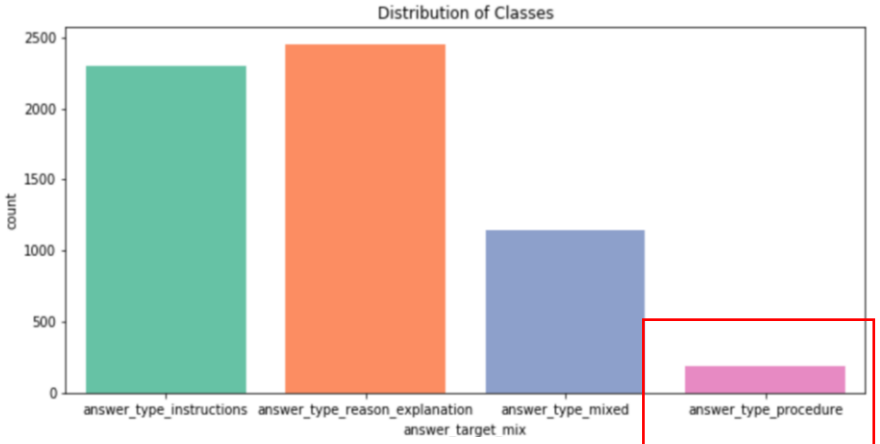
Q&A Category Classification

```
Observation Distribution in Training Data (%):
TECHNOLOGY       40.0
STACKOVERFLOW    21.0
CULTURE          16.0
SCIENCE          12.0
LIFE_ARTS        12.0
Name: category, dtype: float64

Text(0.5, 1.0, 'Observations Distribution of Classes')
```

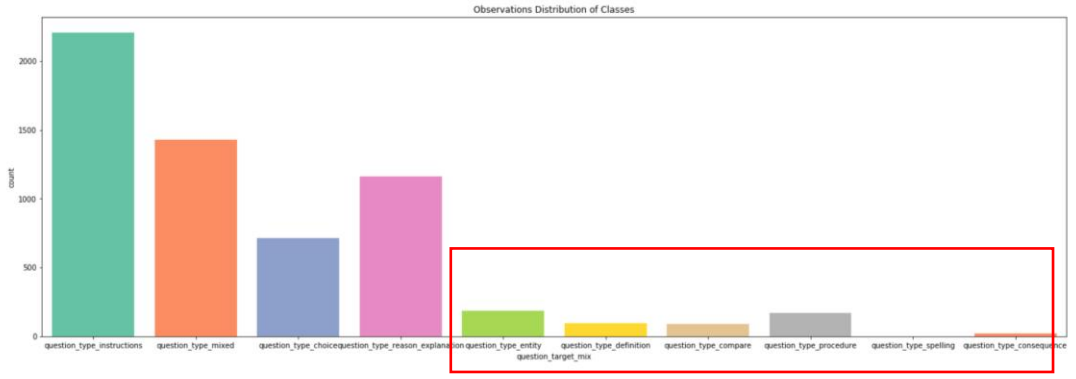


Q&A Type Classification

```
Observation Distribution of 4  Answer Types (%):
answer_type_reason_explanation    40.27
answer_type_instructions          37.87
answer_type_mixed                 18.77
answer_type_procedure              3.09
Name: answer_target_mix, dtype: float64

Text(0.5, 1.0, 'Distribution of Classes')
```




```
Observation Distribution of 10  Questions Types (%):
question_type_instructions          36.34
question_type_mixed                 23.49
question_type_reason_explanation    19.10
question_type_choice                11.71
question_type_entity                 3.08
question_type_procedure              2.83
question_type_definition             1.56
question_type_compare                1.53
question_type_consequence            0.33
question_type_spelling               0.03
Name: question_target_mix, dtype: float64

Text(0.5, 1.0, 'Observations Distribution of Classes')
```



Data Exploratory Analysis | Target Distribution

Problem: Unbalanced Classes

- ❖ Observations are not equally distributed among classes regarding Q&A Category and Q&A Type Classification
- ❖ Extremely affect the accuracy rate of the model – the model can be bias towards the data with more observation

Solution: Over-sampling

- ❖ Randomly oversampling all observations regarding each target to balance the distribution between classes

FEATURE ENGINEERING - TFIDF

1. Logarithmic Form of Frequency (Normalization)
2. Minimum numbers of document a word must be present to be kept is 1
3. Features Vector with Euclidean norm of 1
4. Unigram vs Bigram Combination

```
# TFIDF features
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer(sublinear_tf=True, # use logarithmic form for frequency
                                   min_df=1, # The minimum numbers of document a word must be present to be kept
                                   norm='l2', # Ensure all features vectors have a eulicidean norm of 1
                                   encoding='latin-1',
                                   smooth_idf=True,
                                   use_idf=True,
                                   ngram_range=(1,2)) #unigram and bigram
```

TOPIC: 'CULTURE':
. Most correlated unigrams:
. bike
. god
. english
. game
. word
. Most correlated bigrams:
. move something
. native speaker
. work ability
. present tense
. could say

TOPIC: 'LIFE_ARTS':
. Most correlated unigrams:
. phd
. tax
. camera
. student
. lens
. Most correlated bigrams:
. credit score
. white balance
. zoom lens
. star trek
. focal length

TOPIC: 'SCIENCE':
. Most correlated unigrams:
. times
. electron
. energy
. mathbb
. frac
. Most correlated bigrams:
. black hole
. angular momentum
. potential energy
. in mathbb
. frac frac

TOPIC: 'STACKOVERFLOW':
. Most correlated unigrams:
. void
. string
. return
. gt
. lt
. Most correlated bigrams:
. public void
. public class
. div gt
. lt div
. gt lt

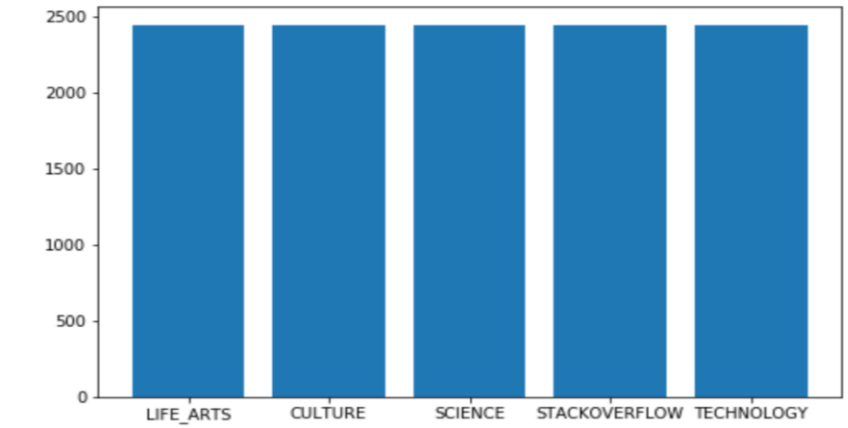
TOPIC: 'TECHNOLOGY':
. Most correlated unigrams:
. usepackage
. sudo
. install
. ubuntu
. server
. Most correlated bigrams:
. ip address
. sudo apt
. apt get
. end document
. begin document

Example
Top Words Most Correlated TFIDF Unigram/Bigram of Category Classification for Answer

OVERSAMPLING

```
# Setup of packages needed
# SMOTE
import imblearn
# print(imblearn.__version__)
from imblearn.over_sampling import SMOTE #use for oversampling
from collections import Counter
# X: input for oversampling
# y: target label for oversampling
oversample = SMOTE()
X_train_question, y_train_question = oversample.fit_resample(tfidf_train_question, train_labels)
X_train_answer, y_train_answer = oversample.fit_resample(tfidf_train_answer, train_labels)
```

```
: Counter({'LIFE_ARTS': 2441,
           'CULTURE': 2441,
           'SCIENCE': 2441,
           'STACKOVERFLOW': 2441,
           'TECHNOLOGY': 2441})
```



- The method of over sampling are used with **SMOTE** - This will **randomly oversample observations** for each category **until** they are **equally distributed** among the train data.

- Oversampling is only **implemented on training data**
- Example: TFIDF for question in part I after being over sampling

PART I
Q&A CATEGORY CLASSIFICATION

MODEL BUILDING

Models

- Bayes Net
- Support Vector Machine
- Random Forest
- Decision Tree

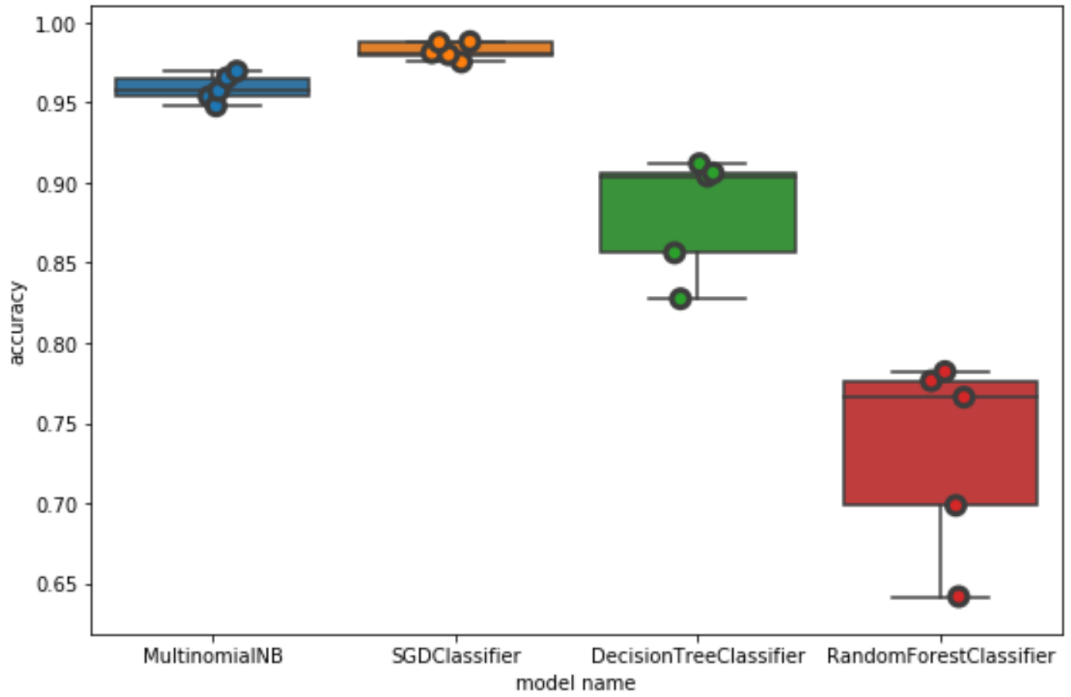
Cross-Validation

- 5-Fold
- Fit each model 5 times to evaluate the accuracy rate

```
# Model Building
from sklearn.naive_bayes import MultinomialNB # Naive Bayes
from sklearn.linear_model import SGDClassifier # Support Vector Machine (SDG)
from sklearn.tree import DecisionTreeClassifier # Decision Tree
from sklearn.ensemble import RandomForestClassifier #RandomForestClassifier

mnb = MultinomialNB() # Naive Bayes
svm = SGDClassifier(loss='hinge', max_iter=100) # SVM SDG
dct = DecisionTreeClassifier(criterion = "entropy", # Decision Tree
                             # set criterion as entropy for the information gain
                             #(measure the quality of the split)
                             random_state = 0)
rf = RandomForestClassifier(n_estimators=200, max_depth=3, random_state=0) # Random Forest
```

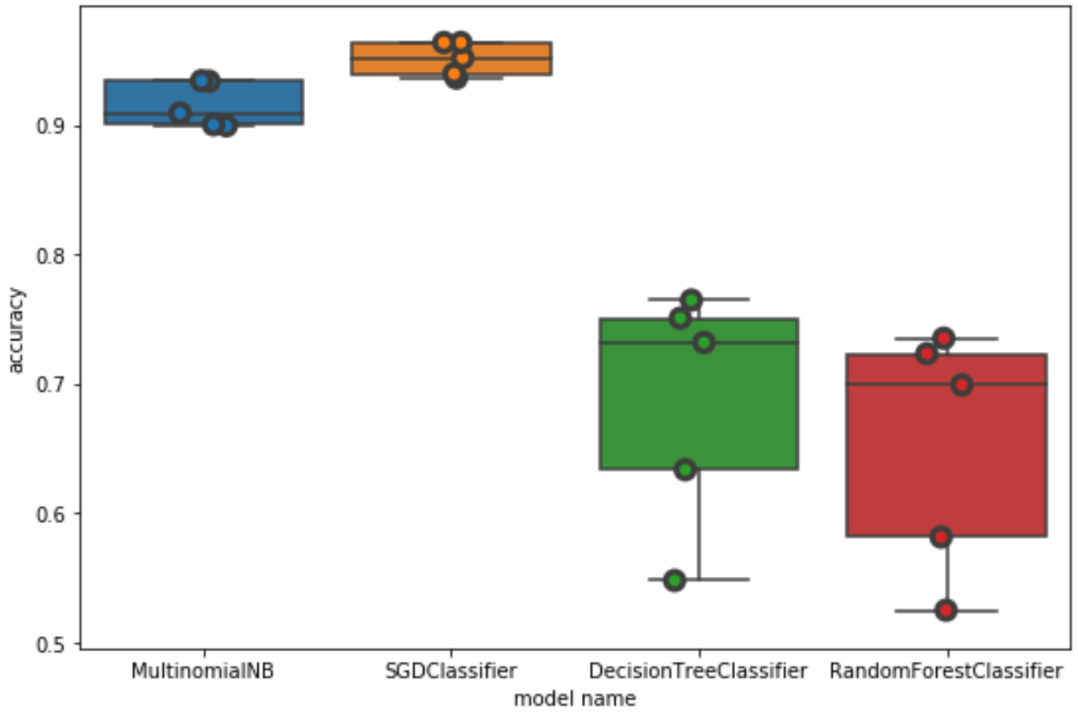
MODEL SELECTION | SUPPORT VECTOR MACHINE

Question | SVM – 98% Accuracy




```
model name
DecisionTreeClassifier    0.881196
MultinomialNB             0.958869
RandomForestClassifier    0.732896
SGDClassifier             0.982466
Name: accuracy, dtype: float64
```

Answer | SVM – 95% Accuracy




```
model name
DecisionTreeClassifier    0.685948
MultinomialNB             0.915608
RandomForestClassifier    0.653011
SGDClassifier             0.951249
Name: accuracy, dtype: float64
```

MODEL EVALUATION |PRECISION – RECALL – F1 SCORE

Question

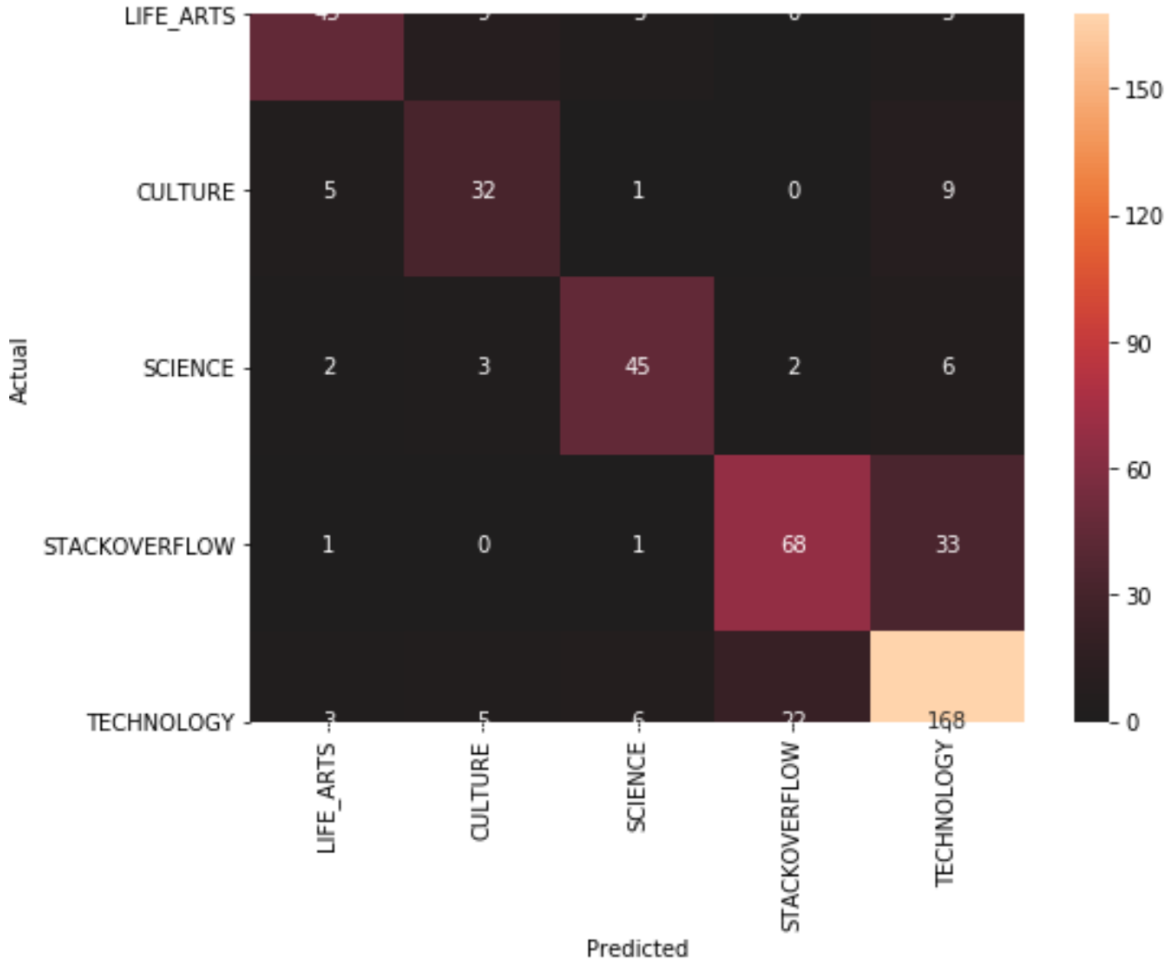
	precision	recall	f1-score	support
LIFE_ARTS	0.80	0.70	0.75	64
CULTURE	0.65	0.68	0.67	47
SCIENCE	0.78	0.78	0.78	58
STACKOVERFLOW	0.74	0.66	0.70	103
TECHNOLOGY	0.76	0.82	0.79	204
accuracy			0.75	476
macro avg	0.75	0.73	0.74	476
weighted avg	0.75	0.75	0.75	476

Answer

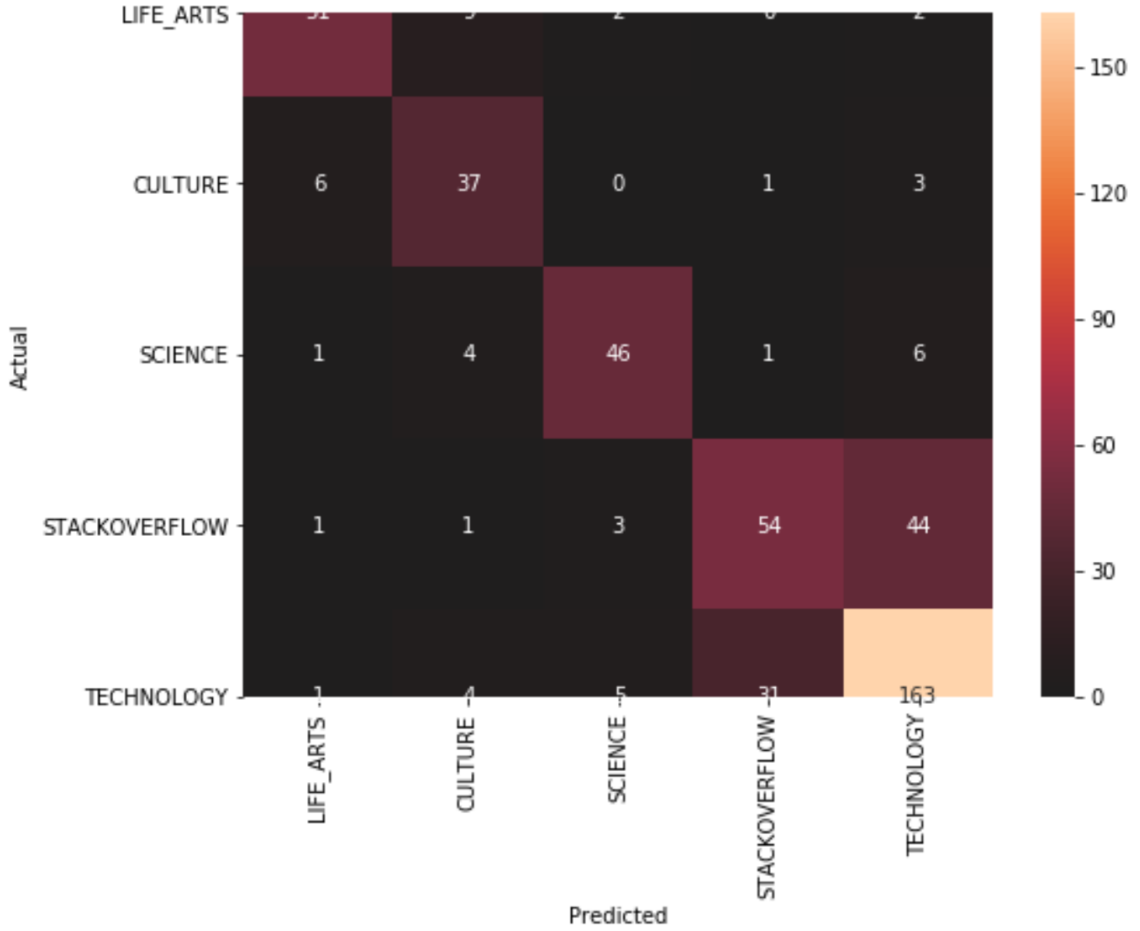
	precision	recall	f1-score	support
LIFE_ARTS	0.85	0.80	0.82	64
CULTURE	0.67	0.79	0.73	47
SCIENCE	0.82	0.79	0.81	58
STACKOVERFLOW	0.62	0.52	0.57	103
TECHNOLOGY	0.75	0.80	0.77	204
accuracy			0.74	476
macro avg	0.74	0.74	0.74	476
weighted avg	0.74	0.74	0.73	476

MODEL EVALUATION | CONFUSION MATRIX

Question



Answer



CATEGORY MISCLASSIFICATION EXAMPLE

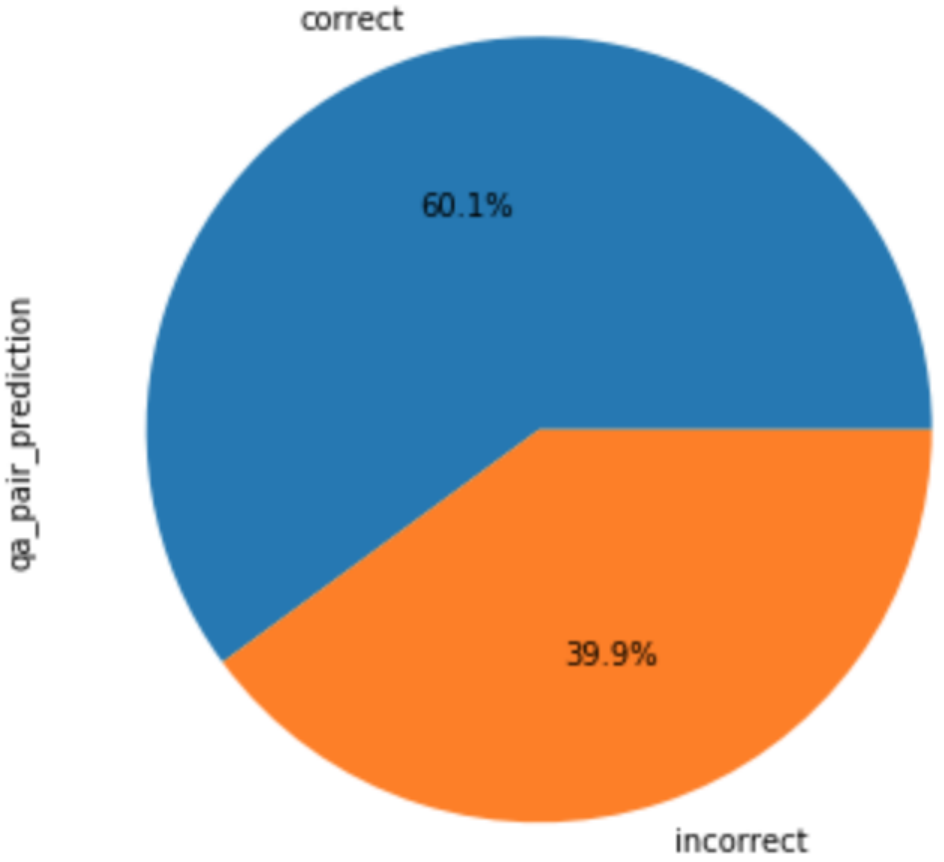
QUESTION

```
Actual Label: TECHNOLOGY
Predicted Label: STACKOVERFLOW
Document:-
what is the location of imei number for android 2.3.3 ?   my phone blacked out after I factory resetted my phone. And need t
o restore my imei back on my  htc droid incredible 2
```

ANSWER

```
Actual Label: TECHNOLOGY
Predicted Label: STACKOVERFLOW
Document:-
You can use RewriteCond %{REMOTE_HOST} or RewriteCond %{REMOTE_ADDR} to set such redirects.  Try something like:   #A.B.C.D
is the referrer IP@ RewriteCond %{REMOTE_HOST} ^A\.B\.C\.D RewriteRule ^(.*)$ http://www.yoursite.com/landingpage.html [NC,
L,U,QSA]
Actual Label: STACKOVERFLOW
Predicted Label: TECHNOLOGY
Document:-
Use if all tables have identity field.  exec sp_MSforeachtable @command1 = 'DBCC CHECKIDENT(''?'', RESEED, 1)'   MSforeachta
ble is an undocumented, but extremely handy stored proc which executes a given command against all tables in your database.
To reseed ONLY tables with an identity column you can use the next script. It also makes use of sp_MSforeachtable but taking
into account the correct tables.  EXEC sp_MSforeachtable ' IF (SELECT COUNT(1)      FROM INFORMATION_SCHEMA.TABLES      WHER
E TABLE_TYPE = ''BASE TABLE''      AND ''['']+ TABLE_SCHEMA + ''].['' + TABLE_NAME + '']'' = ''?''      AND OBJECTPROPERTY(OB
JECT_ID(TABLE_NAME), ''TableHasIdentity'') = 1) &gt; 0  BEGIN     DBCC CHECKIDENT (''?'', RESEED, 1) END'   Use Following qu
ery to get Last Identity value inserted in tables in a database.     SELECT TableName = OBJECT_NAME(OBJECT_ID) ,        Colu
mnName = name ,        OriginalSeed = seed_value ,        Step = increment_value ,        LastValue = last_value ,        Is
NotForReplication = is_not_for_replication FROM sys.identity_columns   Refer this for more details
```

MODEL EVALUATION | Q&A Pair Prediction



60.1% ACCURACY

PART 2
Q&A TYPES CLASSIFICATION

Model Building

1. Partition **80-20** for training and testing
2. Models:

Support Vector Machine (TFIDF)

Naïve Bayes (TFIDF)

Long Short-Term Memory (TFIDF)

- Features Engineering: one hot encoding target
- **TFIDF** vectorizer as features
- Set **weight** to the model (imbalanced classes)
- Train for **2 epochs**
- **Validation dataset** is **20%** of the data

LSTM Deep Learning Demo
Question Type Classification Part

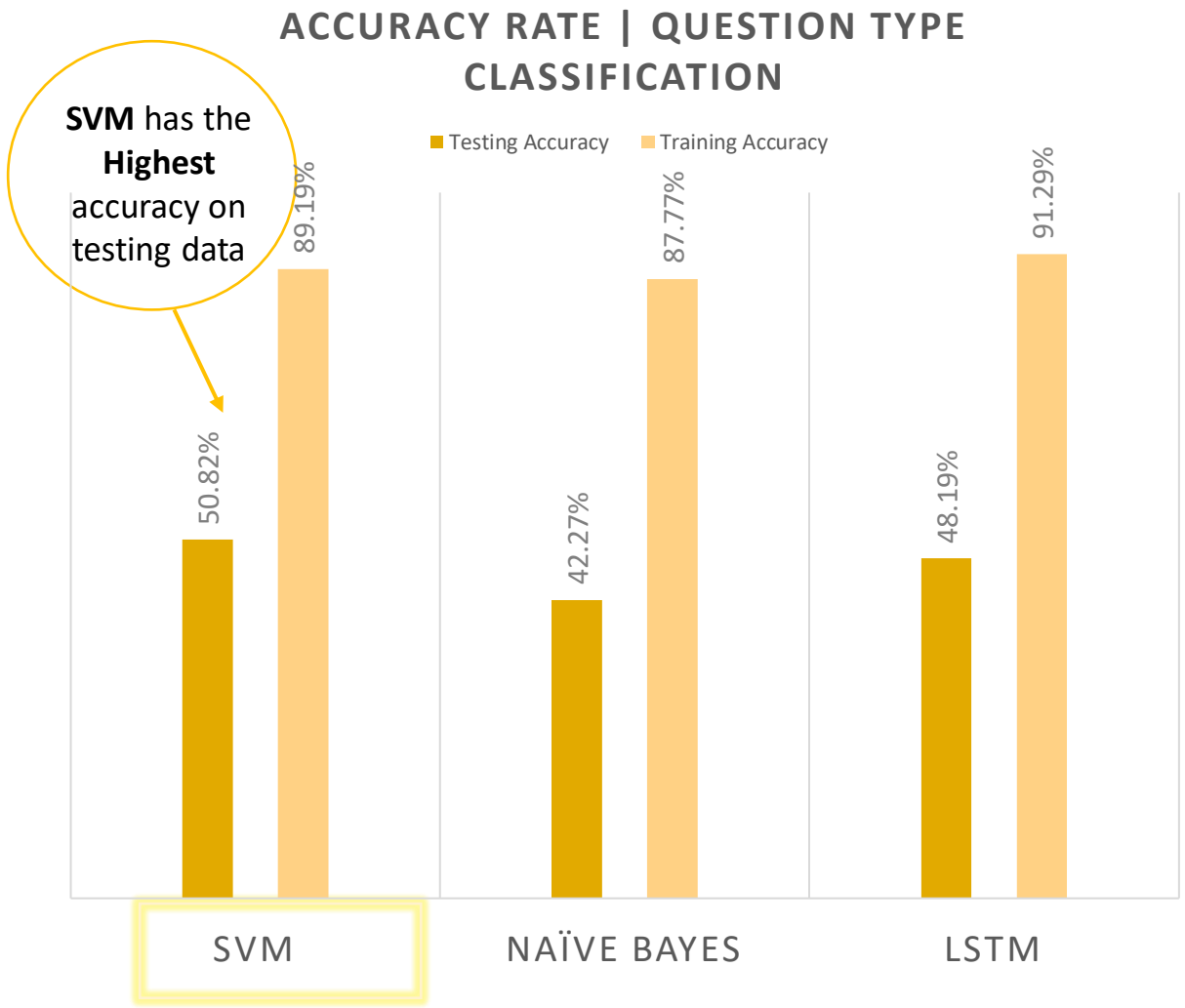
```
# Prepare feature for answer
encoder.fit(train_answer_id)
a_train = encoder.transform(resampled_train_answer_labels)
a_test = encoder.transform(test_answer_id)
a_train = encoder.transform(resampled_train_answer_labels)
a_test = encoder.transform(test_answer_id)

a_classes = np.max(a_train)+1
a_train = utils.to_categorical(a_train, a_classes)
a_test =utils.to_categorical(a_test, a_classes)

z_train = resampled_train_answer
z_test = tfidf_test_answer
```

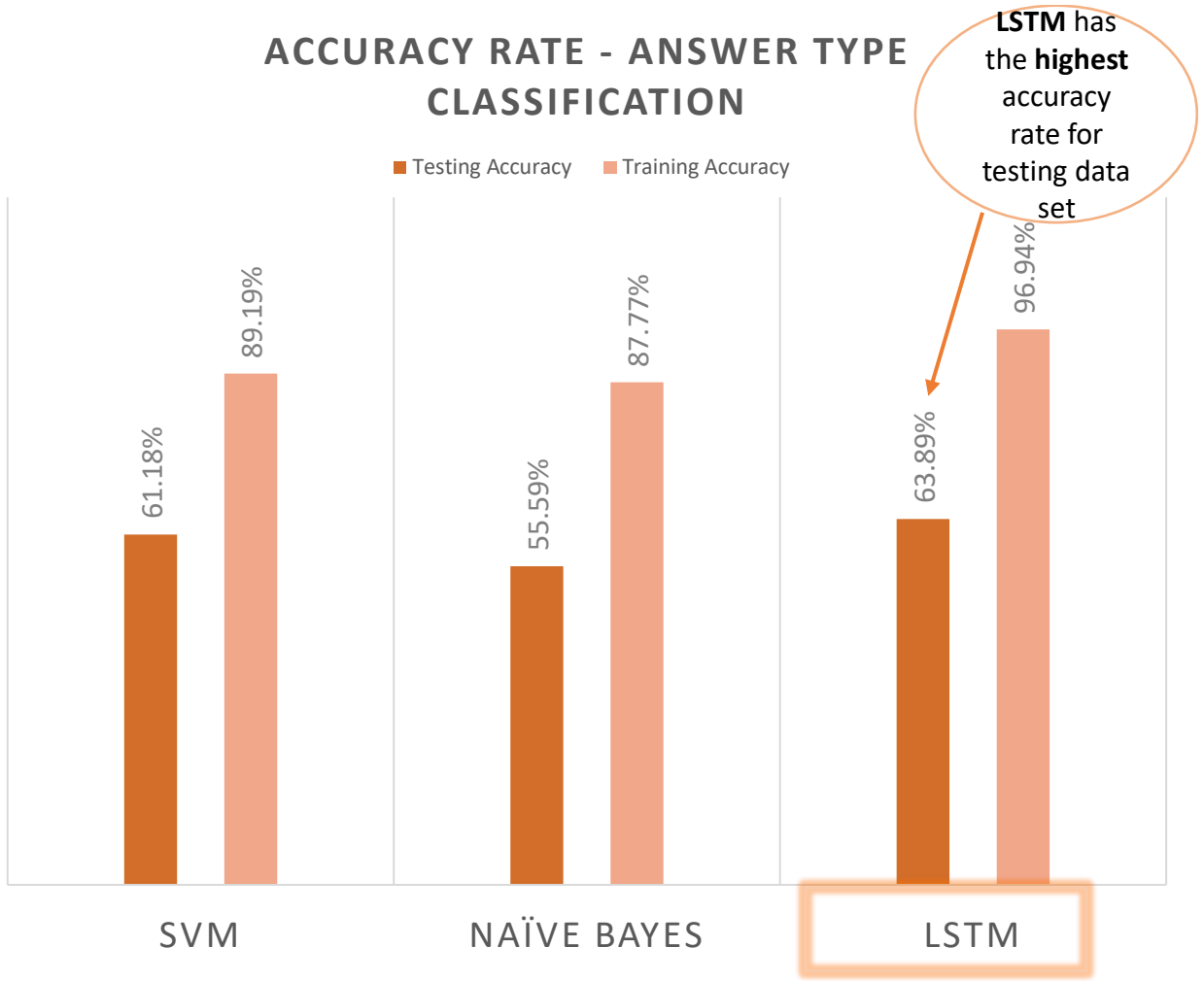
One hot encoding target – turning those target into dummy variables

```
# This model trains very quickly and 2 epochs are already more than enough - however we can still customize
# Training for more epochs will likely to lead to overfitting on the dataset
from sklearn.utils import class_weight
batch_size = 128
epochs = 2
# Adding class weight since our data are imbalance
class_weight = class_weight.compute_class_weight('balanced', np.unique(train_answer_labels), train_answer_labels)

# Build the model
model_a = Sequential()
model_a.add(Dense(512, input_shape=(z_test.shape[1],)))
model_a.add(Activation('relu'))
model_a.add(Dropout(0.5))
model_a.add(Dense(a_classes))
model_a.add(Activation('softmax'))

model_a.compile(loss='categorical_crossentropy',
                optimizer='adam',
                metrics=['accuracy'])
```

Setting weights to the model regarding the target distribution. This can help to deal with unbalanced class problem

MODEL SELECTION





MODEL EVALUATION | PRECISION – RECALL – F1 SCORE

Question Type | SVM

	precision	recall	f1-score
question_type_instructions	0.62	0.78	0.69
question_type_mixed	0.31	0.22	0.26
question_type_choice	0.41	0.41	0.41
question_type_reason_explanation	0.58	0.51	0.54
question_type_entity	0.35	0.27	0.31
question_type_definition	0.44	0.50	0.47
question_type_compare	0.24	0.29	0.26
question_type_procedure	0.13	0.14	0.14
question_type_spelling	0.00	0.00	0.00
question_type_consequence	0.33	0.50	0.40
accuracy			0.51
macro avg	0.34	0.36	0.35
weighted avg	0.49	0.51	0.49

- **Precision Score 49%, recall score 51%, F1 score 49%** - Fair Performance with **10 targets.**
- **Baseline Accuracy** = 1/10 = **10%** - Randomly choose one observation, we have roughly 10% of chance that the guess is right (Probability). Therefore, **the recall score** and **precision score** of SVM are **not so bad** after all.

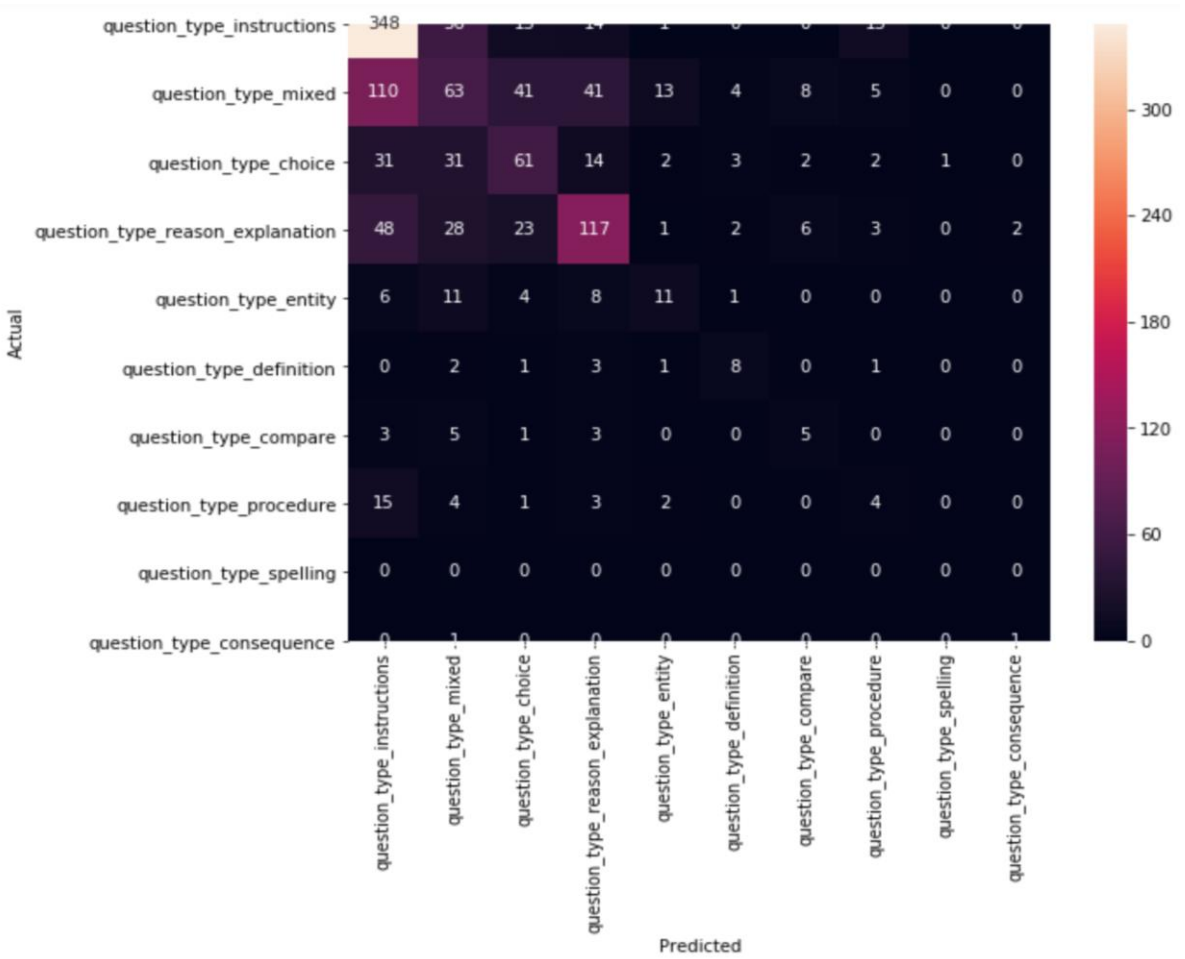
Answer Type | LSTM

	precision	recall	f1-score
answer_type_instructions	0.66	0.75	0.70
answer_type_reason_explanation	0.62	0.80	0.70
answer_type_mixed	0.23	0.07	0.10
answer_type_procedure	0.00	0.00	0.00
accuracy			0.61
macro avg	0.38	0.41	0.38
weighted avg	0.53	0.61	0.56

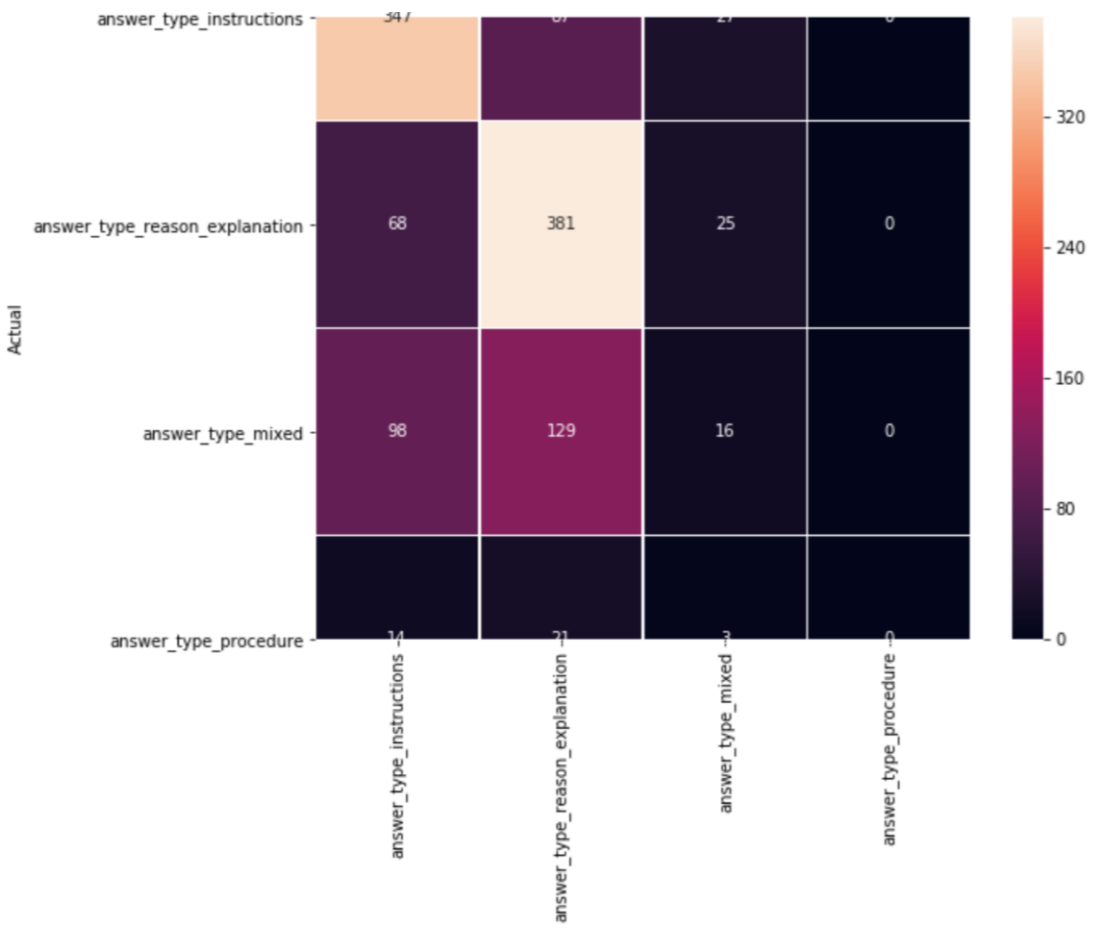
- **Precision Score 53%, recall score 61%, F1 score 56%** - Fair Performance with **4 targets**
- **Baseline Accuracy** = 1/4 = **25%** - Randomly choose one observation, we have roughly 25% of chance that the guess is right (Probability). Therefore, **the recall score** and **precision score** are **not so bad** after all. Furthermore, the **accuracy** is **63%**, roughly 3 times of baseline accuracy.

MODEL EVALUATION | CONFUSION MATRIX

Question Type | SVM



Answer Type |LSTM



Conclusion

1

Q&A Topic/Category Classification

- **Model:** SVM
- **Feature:** TFIDF + Unigram & Bigram
- **Question**: Accuracy of **98%** on **Training** & Precision of **75%** on **Testing**
- **Answer**: Accuracy of **95%** on **Training** & Precision of **74%** on **Testing**

2

Q&A Type Classification

- **Model:** SVM for Question Type & LSTM Neural Network for Answer Type
- **Feature:** TFIDF + Unigram & Bigram
- **Question Type:** SVM – Accuracy **of 89.1%** on **Training** & **50.82%** on **Testing**
- **Answer Type**: LSTM – Accuracy of **96.94%** on **Training** & **63.89%** on **Testing**

Limitation & Recommendation

LIMITATION

- Part II – **Extremely unbalanced classes** between observations. For instance, **question_type_spelling** only has **2 observations**.
- The testing data was split from the train data from Kaggle. Therefore, **the testing** data might **not include all types**. For instance, **question_type_spelling is missing in the testing** data for this project. This could affect the accuracy rate on testing set.
- For Question & Answer Type target**, multiple classes are mixed** due to the maximum probability score for each type is the same for multiple observations, which can **distract the learning ability of the model.**

RECOMMENDATION

- **Collect more data** on each types and **retrain the model.**
- **New feature selection**, such **as POS – part of speech** – This feature might consider the structure of sentence, which can relate to the type of question.


Any Question ?