University of North Carolina at Charlotte
ITCS 6110 Big Data Analytics for Competitive Advantage

Employees and Community/Society KPI Analysis

By,
Nisarg Shah - 801060905
Freny Savalia - 801075313
Urvi Gada - 801029135
Pranay Jain - 801058234
Herambh Yadavalli - 801076518

➢ Introduction to KPI:

A Key Performance Indicator is a measurable value that demonstrates how effectively a company is achieving key business objectives. Organizations use KPIs at multiple levels to evaluate their success at reaching targets. High-level KPIs may focus on the overall performance of the business, while low-level KPIs may focus on processes in departments such as sales, marketing, HR, support and others.

So, what is the definition of KPI? What does KPI mean? What does KPI stand for? Here are a couple other definitions:

Oxford's Dictionary definition of KPI: A quantifiable measure used to evaluate the success of an organization, employee, etc. in meeting objectives for performance.

Investopedia's definition of KPI: A set of quantifiable measures that a company uses to gauge its performance over time.

Macmillan's Dictionary definition of KPI: A way of measuring the effectiveness of an organization and its progress towards achieving its goals.

Now that we know KPI stands for key performance indicator it is only as valuable as the action it inspires. Too often, organizations blindly adopt industry-recognized KPIs and then wonder why that KPI doesn't reflect their own business and fails to affect any positive change. One of the most important, but often overlooked, aspects of KPIs is that they are a form of communication. As such, they abide by the same rules and best-practices as any other form of communication. Succinct, clear and relevant information is much more likely to be absorbed and acted upon.

In terms of developing a strategy for formulating KPIs, your team should start with the basics and understand what your organizational objectives are, how you plan on achieving them, and who can act on this information. This should be an iterative process that involves feedback from analysts, department heads and managers. As this fact-finding mission unfolds, you will gain a better understanding of which business processes need to be measured with a KPI dashboard and with whom that information should be shared.

➢ How to define a KPI?

Defining key performance indicators can be tricky business. The operative word in KPI is “key” because every KPI should related to a specific business outcome with a performance measure. KPIs are often confused with business metrics. Although often used in the same spirit, KPIs need to be defined according to critical or core business objectives. Follow these steps when defining a KPI:

- What is your desired outcome?
- Why does this outcome matter?
- How are you going to measure progress?

- How can you influence the outcome?
- Who is responsible for the business outcome?
- How will you know you've achieved your outcome?
- How often will you review progress towards the outcome?

As an example, let's say your objective is to increase sales revenue this year. You're going to call this your Sales Growth KPI. Here's how you might define the KPI:

- To increase sales revenue by 20% this year
- Achieving this target will allow the business to become profitable
- Progress will be measured as an increase in revenue measured in dollars spent
- By hiring additional sales staff, by promoting existing customers to buy more product
- The Chief Sales Officer is responsible for this metric
- Revenue will have increased by 20% this year
- Will be reviewed on a monthly basis

➢ What is a SMART KPI?

One way to evaluate the relevance of a performance indicator is to use the SMART criteria. The letters are typically taken to stand for **Specific, Measurable, Attainable, Relevant, Time-bound**. In other words:

- Is your objective Specific?
- Can you Measure progress towards that goal?
- Is the goal realistically Attainable?
- How Relevant is the goal to your organization?
- What is the Time-frame for achieving this goal?

➢ Problem Definition:

- This project aims to detect Employees and Community/Society Key Performance Indicators (KPI).
- Mainly situated in "Methods" or "Empirical analysis"
- Training data has: KPI + Description.
- Check what is extracted as description from these methods and empirical analysis.
- Are these summaries or as it is sections? Accordingly train the model to extract description from these two parts.
- You can also investigate automatically generating summary of a group papers, e.g. about the same group of KPIs.

➢ Data:

- The data has been used from "abs1Text" which contains 163 text files.
- The data includes information on a huge set of varied topics.
- First all the data files to be used were read manually to extract unique or important words related to Employee and Community/Society KPIs.
- The data is then cleaned, and all the punctuations, special characters and digits are eliminated using various methods obtained from the nltk packages.
- Next, all the uppercase alphabets are converted to lowercase alphabets.
- All the extracted words are then tokenized using the nltk packages.

➢ Algorithm Used:

- **Topic modeling** is a type of statistical modeling for discovering the abstract "topics" that occur in a collection of documents. **Latent Dirichlet Allocation**(LDA) is an example of topic model and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.
- **Random forests** or **random decision forests** are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.
- Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because it's simplicity and the fact that it can be used for both classification and regression tasks. In this post, you are going to learn, how the random forest algorithm works and several other important things about it.

➢ Process:

- We are using Topic Modelling for extracting words related to Employee and Community/Society KPI.
- We then use these words to manually segregate the documents based on whether they consist of Employee and Community/Society KPI.
- Below is the implementation of the topic modeling using Latent Dirichlet Allocation,

```
print("LDA Model:")

for idx in range(NUM_TOPICS):
    # Print the first 10 most representative topics
    print("Topic #%s:" % idx, lda_model.print_topic(idx, 10))
print("=" * 20)
```

```
LDA Model:
Topic #0: 0.013*"performance" + 0.011*"management" + 0.006*"practices" + 0.006*"journal" + 0.005*"research" + 0.005*"environmen
tal" + 0.005*"study" + 0.004*"social" + 0.004*"organizational" + 0.004*"also"
Topic #1: 0.014*"performance" + 0.010*"management" + 0.006*"journal" + 0.006*"human" + 0.005*"practices" + 0.005*"research" +
0.005*"study" + 0.004*"resource" + 0.004*"organizational" + 0.004*"social"
Topic #2: 0.011*"performance" + 0.009*"management" + 0.006*"environmental" + 0.005*"corporate" + 0.004*"human" + 0.004*"social"
+ 0.004*"study" + 0.004*"journal" + 0.004*"practices" + 0.004*"research"
Topic #3: 0.013*"performance" + 0.009*"management" + 0.006*"journal" + 0.005*"employees" + 0.005*"turnover" + 0.004*"may" + 0.0
04*"practices" + 0.004*"study" + 0.004*"business" + 0.004*"human"
Topic #4: 0.014*"performance" + 0.009*"management" + 0.006*"practices" + 0.006*"organizational" + 0.005*"journal" + 0.005*"envi
ronmental" + 0.004*"human" + 0.004*"firm" + 0.004*"work" + 0.004*"business"
Topic #5: 0.012*"performance" + 0.007*"management" + 0.006*"journal" + 0.005*"social" + 0.005*"corporate" + 0.005*"firms" + 0.0
05*"firm" + 0.004*"research" + 0.004*"may" + 0.004*"organizational"
Topic #6: 0.011*"performance" + 0.008*"management" + 0.005*"environmental" + 0.005*"firm" + 0.004*"research" + 0.004*"practice
s" + 0.004*"journal" + 0.004*"social" + 0.004*"study" + 0.004*"corporate"
Topic #7: 0.013*"performance" + 0.006*"management" + 0.005*"journal" + 0.004*"may" + 0.004*"social" + 0.004*"research" + 0.004
*"relationship" + 0.004*"business" + 0.003*"firm" + 0.003*"environmental"
Topic #8: 0.009*"performance" + 0.008*"management" + 0.006*"journal" + 0.005*"csr" + 0.005*"environmental" + 0.004*"social" +
0.004*"firm" + 0.004*"research" + 0.004*"corporate" + 0.004*"study"
Topic #9: 0.008*"performance" + 0.008*"environmental" + 0.008*"management" + 0.005*"social" + 0.005*"journal" + 0.005*"corporat
e" + 0.005*"firm" + 0.004*"may" + 0.004*"research" + 0.004*"csr"
```

- We are using Python libraries like sklearn to import the data and splitting the dataset into train and test.

```
from sklearn.datasets import load_files
from sklearn.model_selection import train_test_split

datafiles = load_files(r"C:\Users\Nisarg Shah\Desktop\abs1Text\abs1Text - Copy\Text")

X,y = datafiles.data, datafiles.target
```

- Using nltk and re library in python to clean the data by removing digits, special characters and stopwords.

```
documents = []

from nltk.stem import WordNetLemmatizer
import re

stemmer = WordNetLemmatizer()

for sen in range(0, len(X)):
    # Remove all the special characters
    document = re.sub(r'\W', ' ', str(X[sen]))

    # remove all single characters
    document = re.sub(r'\s+[a-zA-Z]\s+', ' ', document)

    # Remove single characters from the start
    document = re.sub(r'\^[a-zA-Z]\s+', ' ', document)
    # Substituting multiple spaces with single space
    document = re.sub(r'\s+', ' ', document, flags=re.I)
    # Removing prefixed 'b'
    document = re.sub(r'^b\s+', '', document)
    # Converting to Lowercase
    document = document.lower()
    # Lemmatization
    document = document.split()
    document = [stemmer.lemmatize(word) for word in document]
    document = ' '.join(document)
    documents.append(document)
```

- Then, by using sklearn.feature_extraction.text CountVectorizer package to vectorize that data in numerical form.

```
from sklearn.feature_extraction.text import CountVectorizer
from nltk.corpus import stopwords
vectorizer = CountVectorizer(max_features=1500, min_df=5, max_df=0.7, stop_words=stopwords.words('english'))
X = vectorizer.fit_transform(documents).toarray()
```

- After that by using the same sklearn.feature_extraction.text library's TfidfTransformer, we save the data into tfidf fromat.

```
from sklearn.feature_extraction.text import TfidfTransformer
tfidfconverter = TfidfTransformer()
X = tfidfconverter.fit_transform(X).toarray()
```

- Then by using the train_test_split package of sklearn library we split the formatted data into training and testing datasets.

```
1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

- We used RandomForestClassifier package from sklearn.ensemble's library to implement a classification model to find out which documents include an Employee or Society KPI.

```
1 from sklearn.ensemble import RandomForestClassifier
2 classifier = RandomForestClassifier(n_estimators=1000, random_state=0)
3 classifier.fit(X_train, y_train)
```

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=1000, n_jobs=1,
            oob_score=False, random_state=0, verbose=0, warm_start=False)
```

- Finally, by using classification_report, confusion_matrix, accuracy_score packages of sklearn.metrics library we print the results which includes a confusion matrix and values of precision, recall, f1-score, support and accuracy which in our case is **<u>72.72%</u>**

```
1 from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
2
3 print(confusion_matrix(y_test,y_pred))
4 print(classification_report(y_test,y_pred))
5 print(accuracy_score(y_test, y_pred))
```

```
[[16  5]
 [ 4  8]]
             precision    recall  f1-score   support

          0       0.80      0.76      0.78        21
          1       0.62      0.67      0.64        12

avg / total       0.73      0.73      0.73        33

0.7272727272727273
```

➢ Results:

- As demonstrated above, the implementation of our model is giving an accuracy of 72.72%
- The precision, recall, f1-score, support values for the implemented model on our data is mentioned as follow:
    - **Precision: - 73%**
    - **Recall: - 73%**
    - **F1-Score: - 73%**
    - **Support: - 33**
- The accuracy is manually verified by printing the names of the documents of the test data and then segregating the documents based whether the document includes the Employee or Society KPI and calculating the percentage of documents including KPI out of the whole total of test data.

➢ How we checked correctness of algorithm:

- We printed the names of the documents which were in the test split.
- We noted all the names and made a list of how many of them included the Employee or Society KPI and how many of them did not.
- Then, we calculated the percentage of the total test split data which included the KPI which was approximately equivalent to the results we are getting.
- By this analysis, we can say that our algorithm is working fine and is giving desired results.

➢ What we learnt:

- Topic modelling using LDA
- Use of tools like Mallet and Gensim
- Use of nltk packages
- Use of scikit-learn library
- Implementation of Random Forest classifier