B9AI109_2022_TMD3

CA-1 Part 1

Sunil Gauda

ID: 10595858

Exercise 2

Assessment

Context

Clinical Data comprises keywords that define the various aspects of the domain. A keyword can be the name of the disease, medication, treatment, symptom, and many other things. These keywords usually connect to define a diagnosis of a specific illness and treatment for it. Using these keywords, Natural Language Processing and programming techniques Author Veena G, Hemanth R, and Jatin Hareesh, developed a method of creating relationships between the keywords, which will help identify and simplify the various processes of the healthcare domain.

The paper is based on the previous work of Veena G et al., who have derived work from A. Makowiecka, M. Marciniak, and A. Kupsc in regards to the rule-based data retrieval system using clinical records and mammogram reports with an understanding of grammatical composition in medical documents. They improved upon it with the techniques of concept extraction by X. Fu and S. Ananiadou which uses machine learning for entity identification in the records, which was mainly discharged summaries and progressive notes of the patient in the hospital with pre/post-operating procedures, similar work was also done by W. T. Abdel-moneim, M. H. Abdel-Aziz, and M. M. Hassan on patient narratives using clinical texts. Further modifying the techniques of information extraction D. Demner-Fushman and J. Lin used to question and answer techniques to derive information for relationship management in the domain of proof-based medicine, The last work to which this paper refers is B. Rink, S. Harabagiu, and K. Roberts who have used a supervised machine learning system for similar relationship extraction task on medical records.

Gap

The key part of this research is data, which in other papers which was referred to were not easy to acquire, fetching data from hospitals is difficult unless one works in the hospital, In this paper, the author has acquired the data using python based web scraping on websites related to medicine, and Word-Net Lexical Database which provides the definition and meaning of the words extracted. Also reproducing the process is difficult due to a lack of Code and Mathematics.

Question

The Paper Aims to solve the issue of relationship understanding between clinical text, as doctors defer by experience and so the treatment or diagnosis of a particular disease can be differently applied to a particular patent, even diseases with different origin and symptoms can connect, this paper provides a simple pipeline to solve these problems.

How did they do the work?

Authors have used python and machine learning to derive meaning from clinical texts and set up a pipeline that uses various techniques to extract, process, Tag, Label, and derive mathematical similarities of data. The process uses various techniques at each stage to provide an appropriate process of relationship definition and extraction between nouns, which can be singular or plural in nature, as this data is easily available on the internet authors did not face much difficulty in applying the known techniques.

What did they find?

Authors find out that the different parts of the body and symptoms and treatments can be related to each other and could be traced back and forth to find out new ways of working on some problems of treatment, this further reduces the complexity of diagnostic methods and provide a wide overview of the field itself.

What is the answer?

The data suggest that the relationship exists between various nouns that are singular or plural used in the field of health science which can be outside the scope of a single method of diagnosis, in academics, it can help in research and learning of new and old healthcare challenges, and in practice, it can pace up the process of diagnostics.

Significance

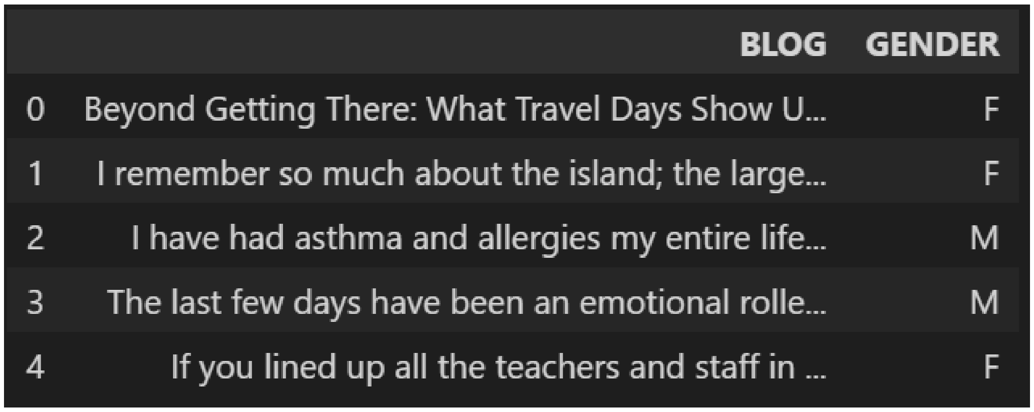
Author Veena G et al., have derived the relationships using clinical data scraped from websites, and have provided an example of working hierarchy and relationship identification of the data which can help identify and diagnose illness faster and help reduce health crises, but the definition of the infrastructure of the process is not completely added to the paper, and the paper lacks a mathematical explanation of POS and PS procedures used in the process.

Exercise 3

Business Understanding

Gender identification is majorly used to check the dynamics of the text which can be used to classify the text as per gender. The classification can form a basis for identifying needs regarding gender and can help us analyze the behavior of the user.

Data Understanding



	BLOG	GENDER
0	Beyond Getting There: What Travel Days Show U...	F
1	I remember so much about the island; the large...	F
2	I have had asthma and allergies my entire life...	M
3	The last few days have been an emotional rolle...	M
4	If you lined up all the teachers and staff in ...	F

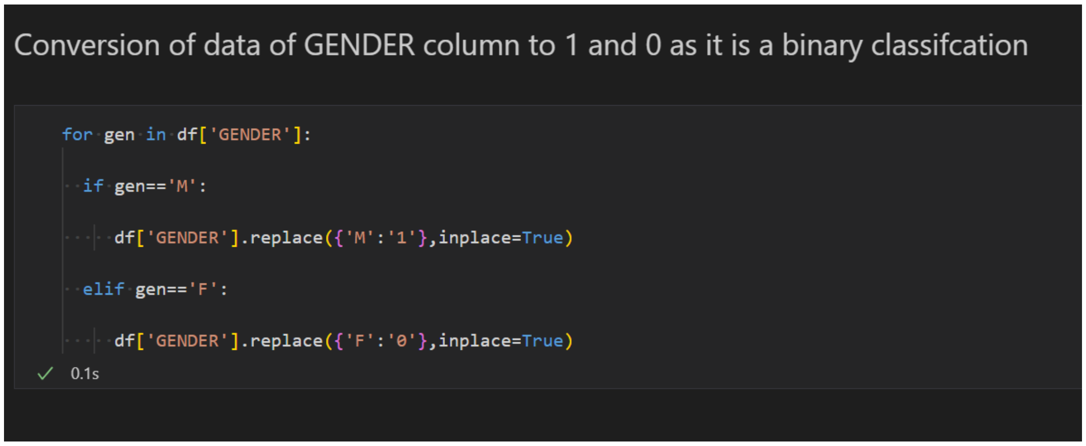
fig 1 Data

The Data above comprises Two Columns, BLOG, and GENDER, the data contains the necessary information for developing a classification system. By Processing the data using tokenization, lemmatization, and vectorization we can create a model to predict gender from the text.

Data Preparation

Numerical Output

As the classification is just binary, The column Gender can be converted to just 1, 0 for M and F representing Male and Female. This will help us computation time and prevent errors on model training as the output data is numerical.

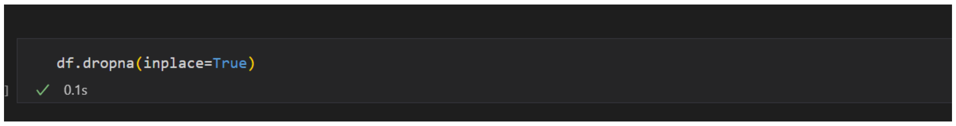


Conversion of data of GENDER column to 1 and 0 as it is a binary classifcation

```
for gen in df['GENDER']:
    if gen=='M':
        df['GENDER'].replace({'M':'1'},inplace=True)
    elif gen=='F':
        df['GENDER'].replace({'F':'0'},inplace=True)
✓ 0.1s
```

fig 2 Data Modification

Null Management



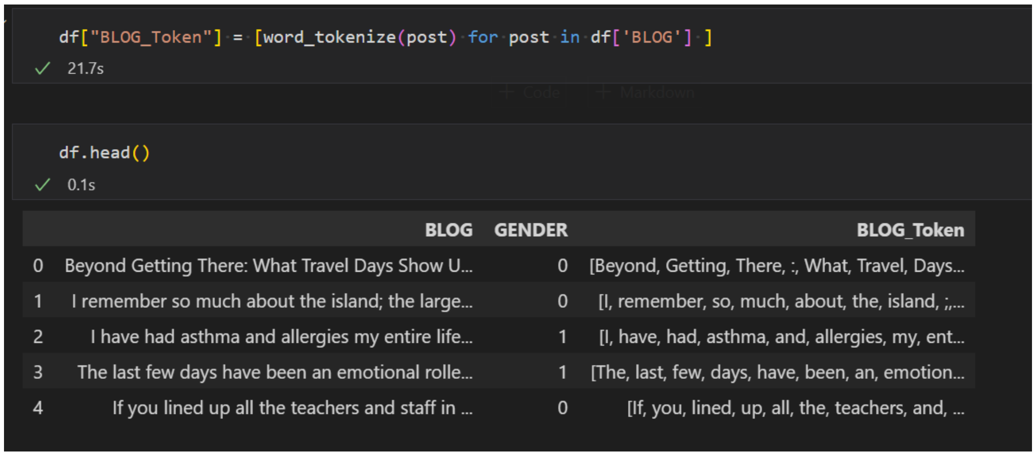
```
df.dropna(inplace=True)
✓ 0.1s
```

fig 3 Null Value Management

Dropping null columns to prevent errors due to invalid data, the data was already blanched and cleared, but as a caution, the null columns need to be dropped, as the data comprises text as input params we cannot generate or augment the data in any way, hence we need to drop the columns of the data.

Tokenize The Data

The entire Sentence does not produce any viable meaning to a machine learning algorithm, so we need to split the data to each word and create smaller units that we can work with.



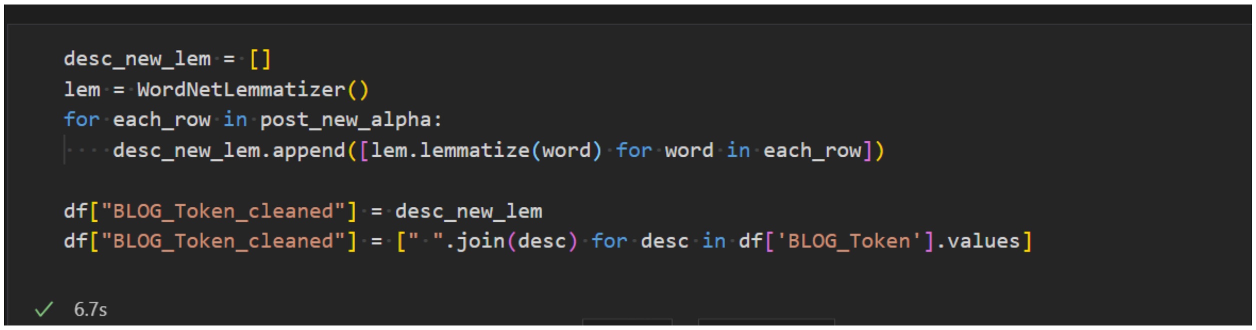
```
df["BLOG_Token"] = [word_tokenize(post) for post in df['BLOG'] ]
✓ 21.7s

df.head()
✓ 0.1s
```

	BLOG	GENDER	BLOG_Token
0	Beyond Getting There: What Travel Days Show U...	0	[Beyond, Getting, There, :, What, Travel, Days...
1	I remember so much about the island; the large...	0	[I, remember, so, much, about, the, island, ;,...
2	I have had asthma and allergies my entire life...	1	[I, have, had, asthma, and, allergies, my, ent...
3	The last few days have been an emotional rolle...	1	[The, last, few, days, have, been, an, emotion...
4	If you lined up all the teachers and staff in ...	0	[If, you, lined, up, all, the, teachers, and, ...

fig 4 Tokenizing

Lemmatization

The problem in text data is the repetition of the words, Which creates ambiguity in the meaning of the word, using lemmatization we can reduce the ambiguity by grouping similar words together, the challenge is to get reference data for the lemmatizer to work, WordNetLemmatizer is one such library that we can use to compare and group the words in a sentence.

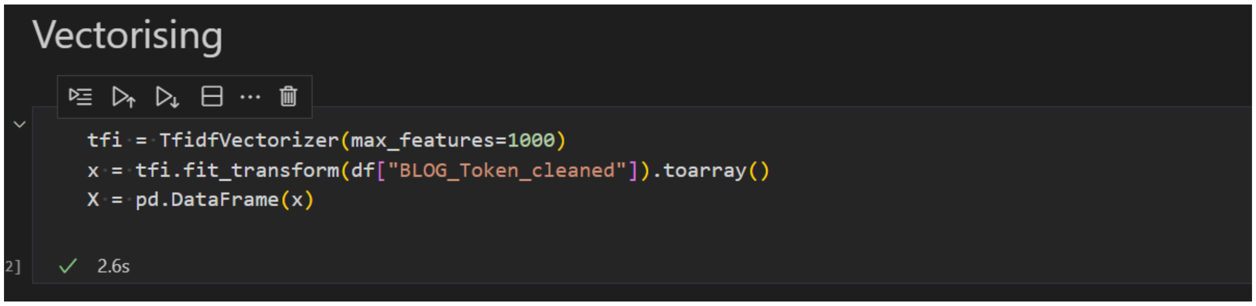


```
desc_new_lem = []
lem = WordNetLemmatizer()
for each_row in post_new_alpha:
    desc_new_lem.append([lem.lemmatize(word) for word in each_row])

df["BLOG_Token_cleaned"] = desc_new_lem
df["BLOG_Token_cleaned"] = [" ".join(desc) for desc in df['BLOG_Token'].values]

✓ 6.7s
```

fig 5 Lemmatization using WordNetLemmatizer

Vectorising

As a computer does not understand textual data, we use vectorization to convert textual processed data to numerical data, we have used TFIDVectoriser as we need to only focus on the frequency of the occurrence of the words in the textual data.



Vectorising

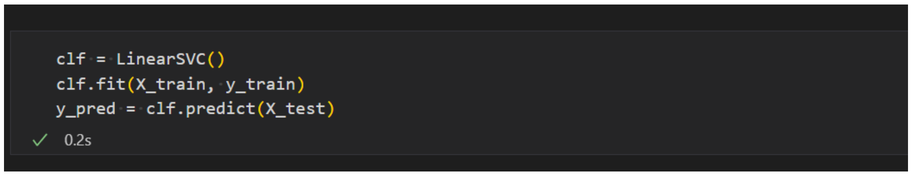
```
tfi = TfidfVectorizer(max_features=1000)
x = tfi.fit_transform(df["BLOG_Token_cleaned"]).toarray()
X = pd.DataFrame(x)
```

```
2]  ✓ 2.6s
```

fig 6 TFIDF Vectorising

Modeling

Modeling comprises the selection of appropriate machine learning algorithms to train on our data. LinearSVC Performs better on small datasets.



```
clf = LinearSVC()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

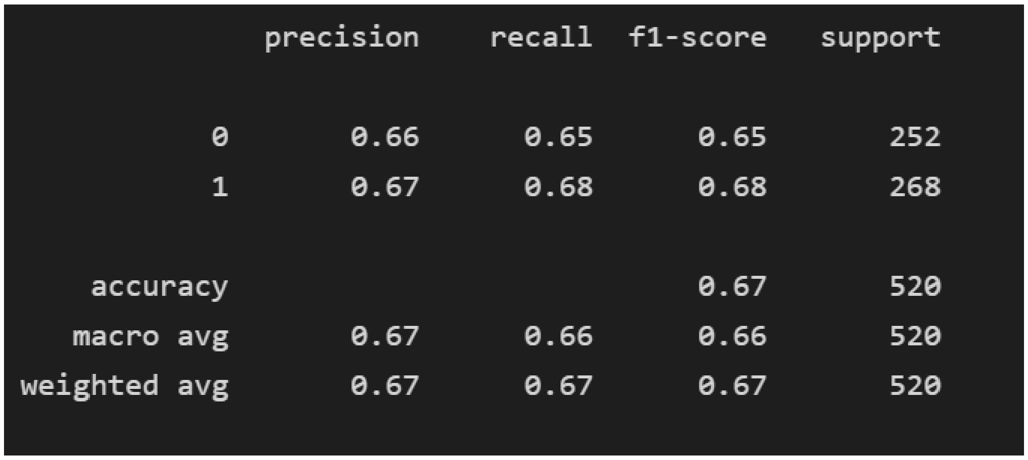
```
✓ 0.2s
```

fig 7 LinerSVC Modelling

Evaluation

We need to Evaluate the model to check its accuracy, in classification we have different means of validating the results of which, Classification Report and Confusion Matrix is used.

Classification Report



```
              precision    recall  f1-score   support

           0       0.66      0.65      0.65       252
           1       0.67      0.68      0.68       268

    accuracy                           0.67       520
   macro avg       0.67      0.66      0.66       520
weighted avg       0.67      0.67      0.67       520
```

fig 8 Classification Report

The model gives an accuracy of 67%, and a precision of 66% with a recall of 65%.

Confusion Matrix

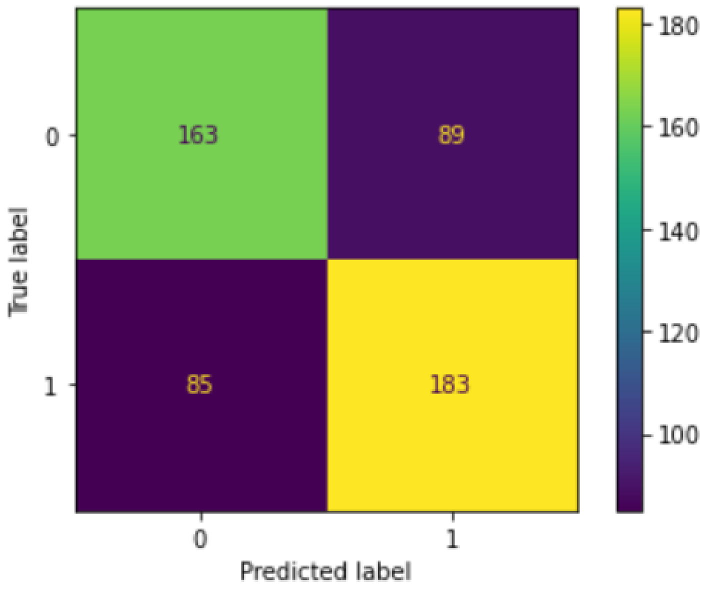
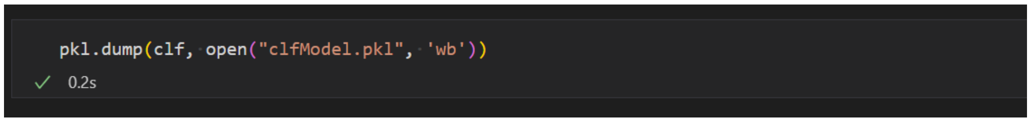


fig 9 Confusion Matrix

Deployment

Using Pickle

We can use a Built-In module called pickle from python to store and load the model to predict the outcome, but the data processing is needed before the input is provided to the pickled model.



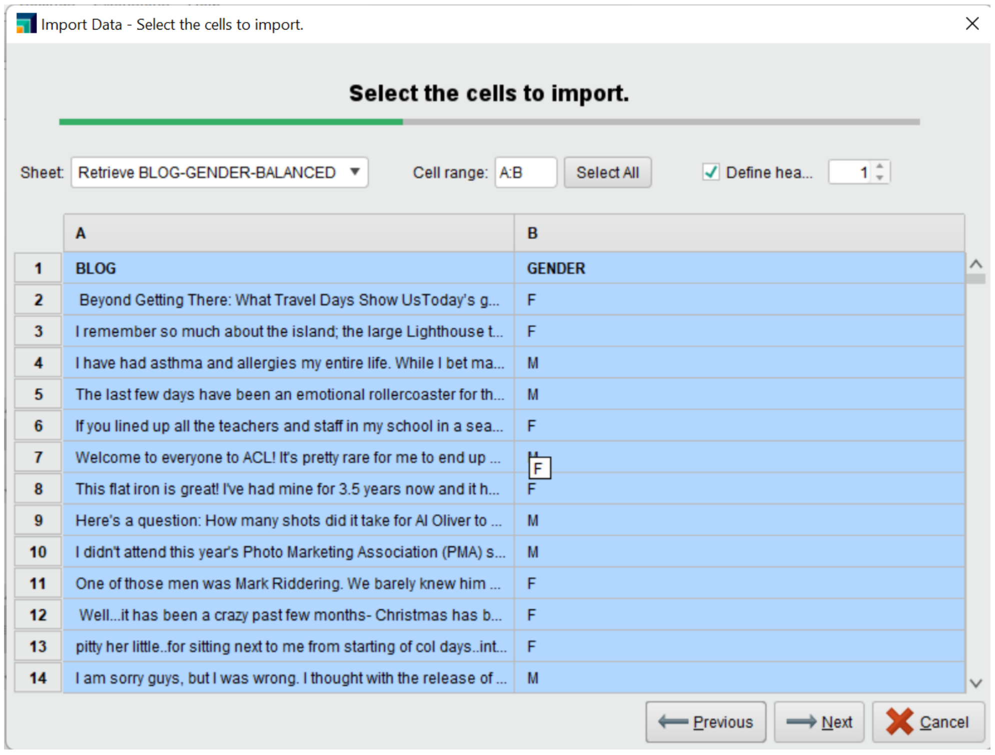
```
pkl.dump(clf, open("clfModel.pkl", 'wb'))
✓ 0.2s
```

fig 10 Saving the Model

RapidMiner Automodel Process

Rapid Miner is a platform that does all the processes above with ease and simplicity, it applies the models and provides us the results, and has an entire pipeline to do rapid machine learning development.

Uploading Data



Import Data - Select the cells to import.

Select the cells to import.

Sheet: Retrieve BLOG-GENDER-BALANCED | Cell range: A:B | Select All | ✓ Define hea... 1

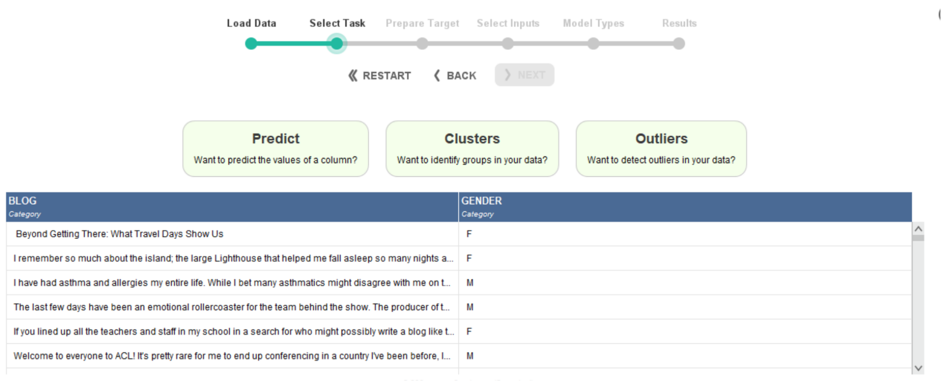
	A	B
1	BLOG	GENDER
2	Beyond Getting There: What Travel Days Show UsToday's g...	F
3	I remember so much about the island; the large Lighthouse t...	F
4	I have had asthma and allergies my entire life. While I bet ma...	M
5	The last few days have been an emotional rollercoaster for th...	M
6	If you lined up all the teachers and staff in my school in a sea...	F
7	Welcome to everyone to ACL! It's pretty rare for me to end up ...	M
8	This flat iron is great! I've had mine for 3.5 years now and it h...	F
9	Here's a question: How many shots did it take for Al Oliver to ...	M
10	I didn't attend this year's Photo Marketing Association (PMA) s...	M
11	One of those men was Mark Riddering. We barely knew him ...	F
12	Well...it has been a crazy past few months- Christmas has b...	F
13	pitty her little..for sitting next to me from starting of col days..int...	F
14	I am sorry guys, but I was wrong. I thought with the release of ...	M

fig 11 Importing Data

We have to select the File from our system to upload it to rapid-miner.

Selecting the Column and the Task we need to perform

We select, predict, and select the column GENDER as it is our target variable.


Load Data
Select Task
Prepare Target
Select Inputs
Model Types
Results
RESTART
BACK
NEXT
Predict
Want to predict the values of a column?
Clusters
Want to identify groups in your data?
Outliers
Want to detect outliers in your data?

BLOG *Category*	GENDER *Category*
Beyond Getting There: What Travel Days Show Us	F
I remember so much about the island; the large Lighthouse that helped me fall asleep so many nights a...	F
I have had asthma and allergies my entire life. While I bet many asthmatics might disagree with me on t...	M
The last few days have been an emotional rollercoaster for the team behind the show. The producer of t...	M
If you lined up all the teachers and staff in my school in a search for who might possibly write a blog like t...	F
Welcome to everyone to ACL! It's pretty rare for me to end up conferencing in a country I've been before, I...	M

fig 12 Selecting Task and Output Column

Overview

Following is the overview of the data for the output variable

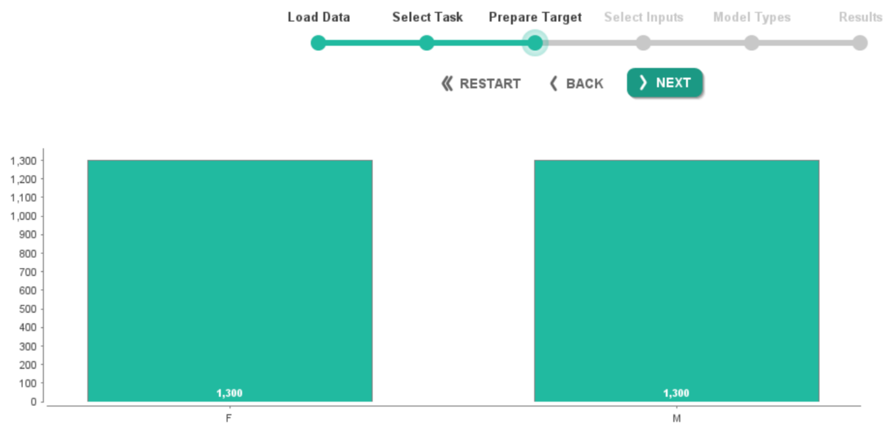
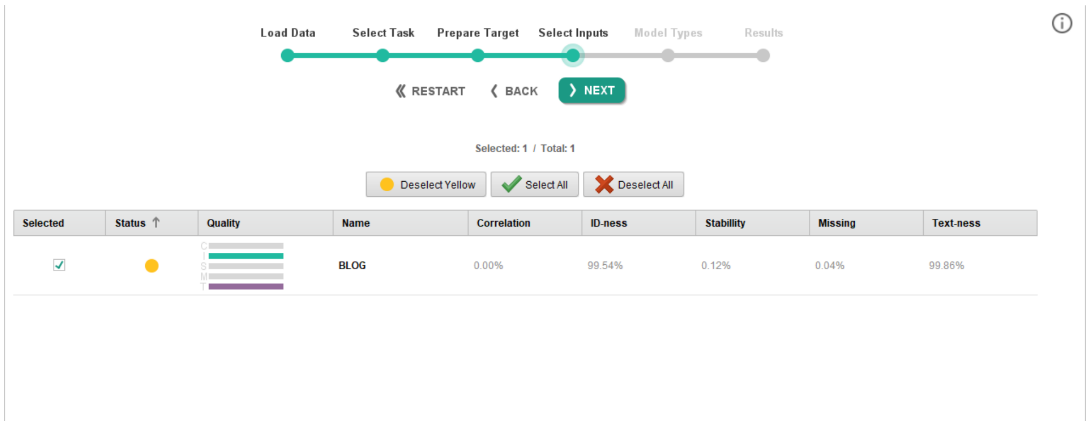


fig 13 Overview

Input Selection

We will select the only input displayed as its a binary classification



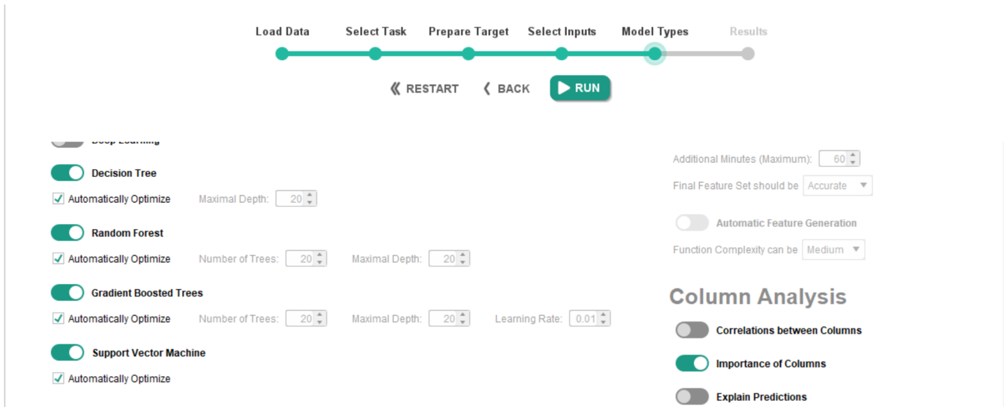
Selected: 1 / Total: 1

Selected	Status	Quality	Name	Correlation	ID-ness	Stability	Missing	Text-ness
✓	●		BLOG	0.00%	99.54%	0.12%	0.04%	99.86%

fig 14 Input Selection

Model Selection

Select what machine learning models we need to apply to our data, the models are implemented by a rapid miner, and parameters are adjusted by a rapid miner.



Decision Tree
✓ Automatically Optimize Maximal Depth: 20

Random Forest
✓ Automatically Optimize Number of Trees: 20 Maximal Depth: 20

Gradient Boosted Trees
✓ Automatically Optimize Number of Trees: 20 Maximal Depth: 20 Learning Rate: 0.01

Support Vector Machine
✓ Automatically Optimize

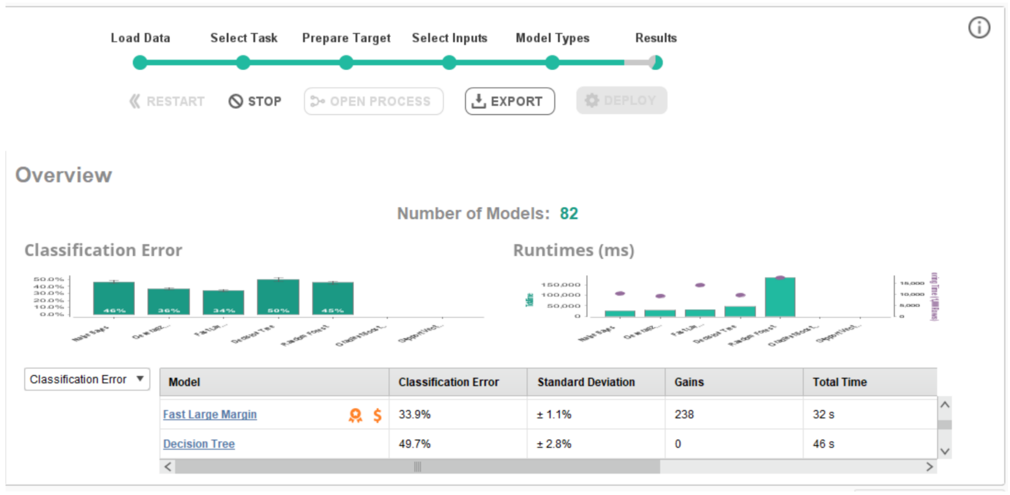
Additional Minutes (Maximum): 60
Final Feature Set should be Accurate
Automatic Feature Generation
Function Complexity can be Medium

Column Analysis
Correlations between Columns
Importance of Columns
Explain Predictions

fig 15 Model Selection

Results

As we can see the results of each model trained and evaluated by Rapid-Miner, we can further export and deploy the models using rapid miner tools.

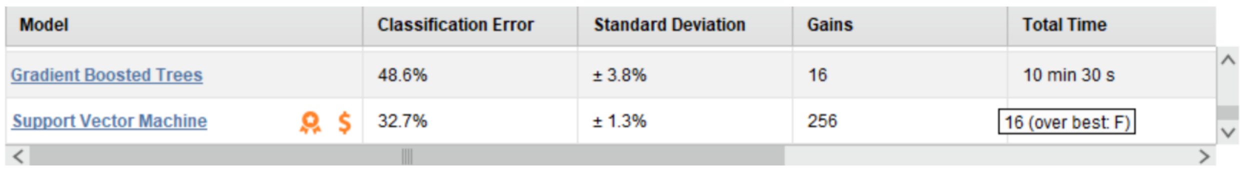


Number of Models: 82

Model	Classification Error	Standard Deviation	Gains	Total Time
Fast Large Margin	33.9%	± 1.1%	238	32 s
Decision Tree	49.7%	± 2.8%	0	46 s

fig 16 Result Window

SVM gave the best results, as we can see below.



Model	Classification Error	Standard Deviation	Gains	Total Time
Gradient Boosted Trees	48.6%	± 3.8%	16	10 min 30 s
Support Vector Machine	32.7%	± 1.3%	256	16 (over best: F)

fig 17 SVM Performance

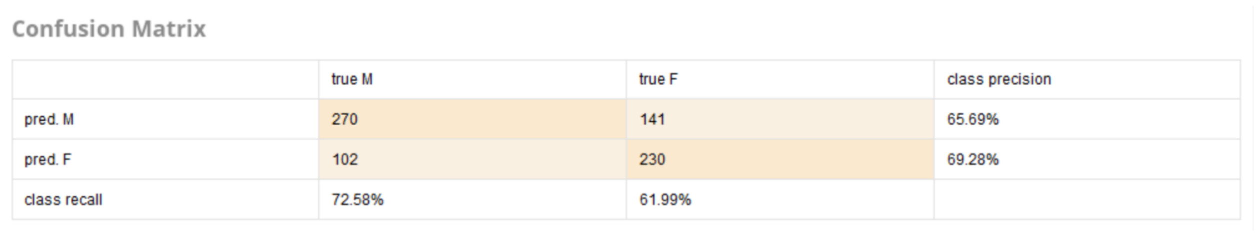
SVM Classification Report Specifies what is the performance of the model in rapid-miner

	NB		GLM		FLM		DT		RF		GBT		SVM	
Criterion	Val	SD	Val	SD	Val	SD	Val	SD	Val	SD	Val	SD	Val	SD
Accuracy	54.0%	±1.8%	63.8%	±1.4	66.1%	±1.1%	50.3%	±2.8%	54.5%	±2.0%	51.4%	±3.8%	67.3%	±1.3%
Classification_error	46.0%	±1.8%	36.2%	±1.4	33.9%	±1.1%	49.7%	±2.8%	45.5%	±2.0%	48.6%	±3.8%	32.7%	±1.3%

AUC	61.7%	±3.6%	70.9%	±1.9	72.9%	±1.8%	54.6%	±3.0%	66.7%	±4.8%	69.1%	±3.0%	73.3%	±2.0%
Precision	52.3%	±2.1%	60.2%	±3.9	66.6%	±5.2%	50.3%	±2.8%	52.6%	±2.4%	50.9%	±3.3%	69.4%	±4.6%
Recall	96.3%	±1.9%	82.1%	±3.3	65.1%	±3.8%	100%	±0.0%	97.4%	±2.2%	99.7%	±0.6%	62.2%	±3.9%
f_measure	67.8%	±1.8%	69.3%	±1.7	65.6%	±1.7%	66.9%	±2.5%	68.3%	±2.0%	67.4%	±3.0%	65.4%	±1.7%
Sensitivity	96.3%	±1.9%	82.15	±3.3	65.1%	±3.8%	100%	±0.0%	97.4%	±2.2%	99.7%	±0.6%	62.2%	±3.9%
Specificity	11.0%	±4.3%	45.8%	±3.6	67.4%	±4.7%	0.0%	±0.0%	11.1%	±2.7%	2.5%	±2.1%	72.8%	±3.4%

fig 18 Classification Report SVM

The Confusion Matrix of the model shows a similar result



Confusion Matrix

	true M	true F	class precision
pred. M	270	141	65.69%
pred. F	102	230	69.28%
class recall	72.58%	61.99%	

fig 19 Confusion matrix Rapid-Miner

Note

Please find the artifacts of the programms in the zip folder submitted.