Executive Summary

- Data was slightly dirty
- Target was very imbalanced
- Value in both the sentences (words) & labels
- Found the most valuable was the word vectors of each sentence
- Built 2 models
  - Initial model: Predict the most common class
  - Final model: Predicts the final class
- GBM Model has a weighted average precision of **82%**

Test Data Accuracy

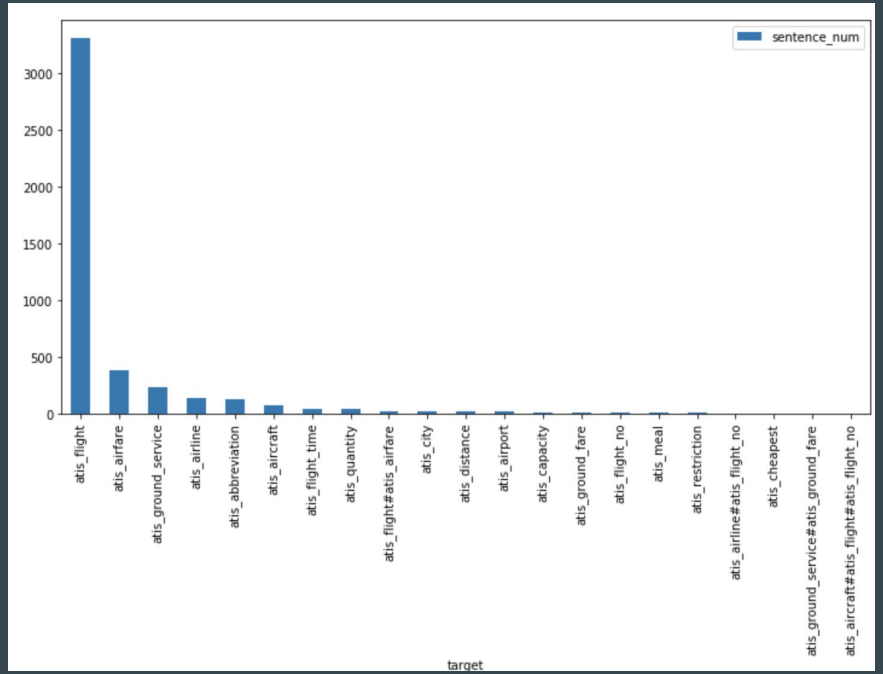
	precision	recall	f1-score	support
atis_abbreviation	0.88	0.88	0.88	17
atis_aircraft	0.50	0.36	0.42	11
atis_airfare	0.47	0.63	0.54	38
atis_airfare#atis_flight_time	0.00	0.00	0.00	1
atis_airline	0.65	0.61	0.63	18
atis_airport	1.00	0.33	0.50	3
atis_capacity	1.00	1.00	1.00	1
atis_city	0.00	0.00	0.00	1
atis_distance	0.00	0.00	0.00	3
atis_flight	0.90	0.95	0.92	357
atis_flight#atis_airfare	0.00	0.00	0.00	2
atis_flight_time	0.00	0.00	0.00	9
atis_ground_fare	0.00	0.00	0.00	3
atis_ground_service	0.95	0.72	0.82	25
atis_quantity	1.00	0.50	0.67	10
atis_restriction	0.00	0.00	0.00	1
micro avg	0.84	0.84	0.84	500
macro avg	0.46	0.37	0.40	500
weighted avg	0.82	0.84	0.82	500

Steps Taken

1. Explored the raw data
2. Preprocessed the data into an usable pandas dataframe
3. Built multiple types of features
4. Built initial model on all types of feature sets
5. Evaluated each initial model & selected best feature set
6. Built final initial and secondary models
7. Built classes and scoring scripts

Data Discovery

- 5,000 samples (90%/10% train/test)
- Unfamiliar IOB format
- Dirty Data
  - Sentences had “BOS” & “EOS”
  - Additional label in the label columns
  - New line character at the end of the line
- Classes
  - Imbalanced
  - Some classes are just multiple classes put together



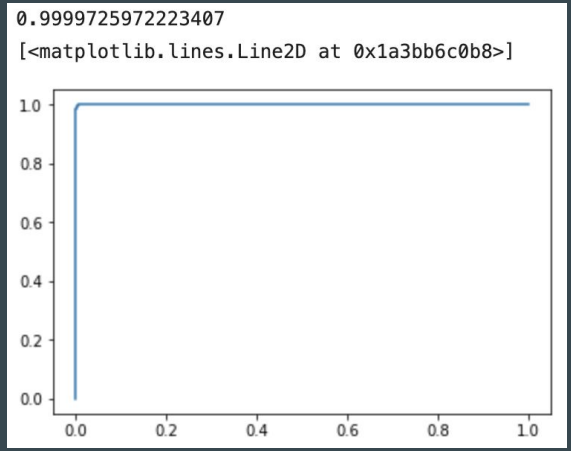
Feature Sets Built

- TF-IDF
  - Labels
  - Sentences
- Word Counter
  - Labels
  - Sentences
- SpaCy Word Vectors
  - Sentences
- Metadata
  - Number of words
- Additional potential targets

Modeling

- Built an initial model to predict the majority class
- Built a final model with input from the initial to predict the final class
- Selected Gradient Boosted Machine
- Performed GridSearch
- Used KFold
- Model is pretty blatantly overfit

Initial Model Training Performance



Final Model Training Performance

	precision	recall	f1-score	support
atis_abbreviation	1.00	1.00	1.00	130
atis_aircraft	1.00	1.00	1.00	70
atis_aircraft#atis_flight#atis_flight_no	1.00	1.00	1.00	1
atis_airfare	1.00	1.00	1.00	385
atis_airline	1.00	1.00	1.00	139
atis_airline#atis_flight_no	1.00	1.00	1.00	2
atis_airport	1.00	1.00	1.00	17
atis_capacity	1.00	1.00	1.00	15
atis_cheapest	1.00	1.00	1.00	1
atis_city	1.00	1.00	1.00	18
atis_distance	1.00	1.00	1.00	17
atis_flight	1.00	1.00	1.00	3309
atis_flight#atis_airfare	1.00	1.00	1.00	19
atis_flight_no	1.00	1.00	1.00	12
atis_flight_time	1.00	1.00	1.00	45
atis_ground_fare	1.00	1.00	1.00	15
atis_ground_service	1.00	1.00	1.00	230
atis_ground_service#atis_ground_fare	1.00	1.00	1.00	1
atis_meal	1.00	1.00	1.00	6
atis_quantity	1.00	1.00	1.00	41
atis_restriction	1.00	1.00	1.00	5
micro avg	1.00	1.00	1.00	4478
macro avg	1.00	1.00	1.00	4478
weighted avg	1.00	1.00	1.00	4478

With More Time

Feature Engineering

- Combined many features built
- With more data, built a custom language model
- Built Doc2Vec model for sentence vectors
- Conduct some more NLP specific extraction (standardization, trimming, etc.)
- Used more metadata features

Modeling

- OneVsAll model
- LSTM for label order prediction
- Classifier chains

Code & Execution

classes.py

- contains a class to process the data and a class to score the dataset

execute.py

- Loads and executes the classes above with input parameters
- Prints classification report to console

Sample Commands

- `python execute.py <input_data_path> <output_data_name>`
- `python execute.py test.iob processed_test_data.pkl`