

# PDF Classifier

By: Matthew Wong

# Motivation

- ❑ Businesses get hundreds of PDFs
- ❑ PDFs are sorted by employees
- ❑ Takes a lot of time and costs money
- ❑ Goal: Classify PDFs



# Data

- ❑ 14 imbalanced classes
  - ❑ Appraisal
  - ❑ Escrow
  - ❑ Insurance
- ❑ ~700 PDFs
- ❑ PDFs in each classification vary
  - ❑ Length
  - ❑ Type: Forms, Scans, Faxes

# Text Extraction

- ❑ PDFminer
  - ❑ Works well for PDFs with text layers
  - ❑ Much quicker than PyOCR
- ❑ PyOCR (python wrapper for Tesseract)
  - ❑ Works well for most PDFs
  - ❑ Takes a long time

# Different Models

## Naive Bayes

Simpler NLP model

Worked well for some categories and not so great with others

## TF-IDF

Average Cosine Similarity

Tried to capture signal from variation in documents of the same category

## TF-IDF

kNN Cosine Similarity

Only looked at the n most similar documents

# Cross Validated Accuracy

- ❑ Used Stratified Cross Validation due to imbalanced classes

	Naive Bayes	Avg Cos Sim	kNN Cos Sim
PDFminer:	~ 0.52	~ 0.65	~ 0.70
PDFminer & PyOCR:	~0.57	~0.87	~0.85(5) / ~0.91(7)
PyOCR:	~0.48	~0.60	~0.78

# kNN Cosine Similarity

- ❑ On Average
  - ❑ 7 Classes had  $> 90\%$  accuracy
  - ❑ 2 Classes had  $\sim 80\text{-}90\%$  accuracy
  - ❑ 3 Classes had  $\sim 50\text{-}80\%$  accuracy
  - ❑ 2 Classes almost always predicted wrong

# Future Work

- ❑ Renaming/Filing documents automatically
- ❑ Extracting specific text strings
  - ❑ Name
  - ❑ Address
  - ❑ \$\$\$ Values



# Thank You

# Questions?

Matthew Wong

[mjwong1991@gmail.com](mailto:mjwong1991@gmail.com)

<https://www.linkedin.com/in/matthew-j-wong/>

<https://github.com/Unrelenting>