# Final Project: Course 1: Introduction to Data Analytics Project Title: J P Morgan classification for legal documents

## Question:

Distinct the process based on all the steps in CRISP-DM.CRISP-DM Approach to Automate the Classification of Various Legal Documents

## My Approach:

### 1. Business Understanding

When I decided to automate the classification of legal documents—taking inspiration from JP Morgan's COIN software—I first tried to understand why this system was important. In my situation, reviewing complex legal documents was taking a lot of time and money. Lawyers and loan officers had to read through these contracts carefully, and it often took thousands of hours to spot any mistakes. This time-intensive process made me realize:

- Goal: I wanted a solution that could quickly review legal documents and classify them into different categories, or identify important clauses or "attributes."
- Benefits: Automating this process would save a huge amount of time (so lawyers could focus on more critical or strategic tasks) and reduce the chance of missing essential details due to human fatigue.

Ultimately, my main purpose was to cut down on the time and labor spent reading contracts while still improving accuracy.

### 2. Data Understanding

Once I was clear about my reasons, I moved on to understanding the data involved in this classification task. In JP Morgan's scenario, the data consists of various types of legal contracts, such as:

- Commercial loan agreements
- Credit-default swaps
- Custody agreements
- Other financial/legal documents

These documents might come in different formats (like PDFs, scanned images, or Word files), and each contract can be lengthy, containing multiple clauses. When I looked into these documents, I noticed:

- Content Variation: Sometimes, the same clause is worded differently across contracts.
- Structure Differences: Some agreements follow standard templates, while others are more unique.
- Text Quality: If documents are scanned, I might need Optical Character Recognition (OCR) to convert them into machine-readable text.
- Volume: There could be thousands or millions of pages in total, so manual review is very time-consuming.

**3. Data Preparation**

After I understood what data I was dealing with, I had to prepare it for the machine learning (or algorithmic) model. I approached data preparation in the following way:

1) Collect and Organize: I gathered all the legal documents in one system.
2) Data Cleaning: I removed any weird symbols, corrected formatting issues, and resolved errors from OCR (like misspelled words or strange characters).
3) Labeling: Whenever I knew a clause belonged to a specific category (e.g., “repayment clause,” “default clause,” or “interest rate clause”), I labeled it. This labeling process helped the model learn how to categorize new documents.
4) Splitting Data: I divided the labeled data into training, validation, and test sets. This step let me train my model on one set of data and then check how well it did on other, unseen data.

I found that good data labeling is absolutely essential because if the data is messy or mislabeled, the model might not learn correctly.

**4. Modeling**

With my data ready, I moved on to the modeling stage. I like to think of modeling as choosing the right technique or “algorithm” to detect patterns in the text. For legal document classification, I considered:

1) Text Classification Models: These rely on natural language processing (NLP). Models like Logistic Regression or Random Forest can work well, and sometimes advanced neural networks (like BERT) can offer better results.
2) Pattern Recognition: If there were contracts with repeated templates, I could build rule-based systems at first, then combine them with machine learning to handle more complex cases.
3) Clustering: If I wasn’t sure of all the possible clause types, I could use clustering (like k-means) to automatically group similar text segments.

During modeling, I experimented with different algorithms and hyperparameters (like learning rate or the number of layers in a neural network). I also tried different text features (like keywords, n-grams, or embeddings).

## 5. Evaluation

Once I had a model, I needed to check how accurately it classified the documents or clauses. Key metrics I focused on included:

- Accuracy: The overall percentage of clauses that got the correct label.
- Precision & Recall: These were critical because I didn't want to miss important clauses, and I also didn't want to create too many false alarms.
  * Precision: Out of all the clauses my model labeled as "default clause," how many were actually correct?
  * Recall: Out of all the actual "default clauses," how many did my model catch?
- F1-Score: A balance between Precision and Recall.
- Validation on Real Data: I also tested my system on new, real-world documents to ensure that the model performed well beyond just the training dataset.

If my model didn't perform well enough, I would loop back, either gather more data or tweak the model further.

## 6. Deployment

When I was finally satisfied with my model's performance, I considered how to deploy it:

- Integration: The model might need to connect with existing document management systems at a bank or another organization.
- Monitoring & Maintenance: I realized I had to keep an eye on how the model performed over time. If the types of documents changed or if new legal language appeared, I'd likely need to re-train or refine the model.
- User Feedback: If I gave access to lawyers or loan officers, I wanted them to be able to report incorrect classifications or highlight important findings, which would help me improve the system continuously.

## 7. Possible Next Steps

CRISP-DM is a continuous cycle, so even after deploying, I wasn't finished. I planned to:

- Collect More Data: The model can always benefit from more examples, especially if it struggles with certain clause types.
- Review Model Performance: I would keep measuring the model's accuracy and comparing it to previous benchmarks.
- Enhance the System: Eventually, I might add features like automatic summarization or highlighting particularly risky clauses.

## Conclusion

By breaking everything down using CRISP-DM, I felt like I had a clear roadmap: I understood the business need (saving time and reducing contract review errors), gathered and prepared the data, chose and trained a suitable model, evaluated its performance, and finally deployed it. This step-by-step framework helped me ensure that my final solution met my original objectives and continues to get better over time.

**Please find the google drive link below to access the document and video explanation:**

Google Drive