**\*\*\* This document to be submitted via email to** [**challenge2020@gsa.gov**](mailto:challenge2020@gsa.gov) **\*\*\***

# **Contact information for Official Representative:**

**Name:** Eugene Cho

**Email:** echo@guidehousefederal.com

**Team Name:** Guidehouse

# **Names of additional team members:**

**Name:** Laura Beaudry

**Name:** Chris Goodrich

**Name:** Devan Visvalingam

**Name:** Elena Ramlow

**Name:** Eshann Toteja

**Name:** Stephanie Ling

**Name:** Stephen Williams

# **Introduction to Team:**

Guidehouse offers skilled consulting teams with fresh perspectives to help our commercial and public sector clients achieve mission success in innovative yet practical ways. We present agile solutions with sustainable outcomes that meet mission requirements and drive the transformation required to compete, deter, and win in our evolving world.

Guidehouse has proven artificial intelligence and intelligent automation capabilities to assist organizations in assimilating available information to shape data-driven decision making and drive forward-looking insights and is the leading provider of AI/ML solutions in the public sector. We have successfully deployed AI solutions across 30+ government programs, including the development of AI-driven, real-time, open source Machine Learning solutions. Our team has extensive experience with enterprise commercial packages, as well as open source systems. Our understanding of these systems’ strengths and weaknesses positions us to provide sound implementation of AI and ML solutions. We have applied our AI/ML capabilities to a wide range of solutions enabling data-driven decision-making for Federal agencies, including novel approaches to predictive modeling and simulation. Guidehouse is continuously at the forefront of AI/ML solution delivery in the federal sector, being the first to deliver AI-based solutions at multiple agencies.

At GSA, Guidehouse (formerly PwC Public Sector) has been a trusted partner for 20 years in delivering data-driven solutions and capabilities across the agency. We have extensive experience with GSA business processes, information systems, and risk areas by virtue of supporting the organization. We currently provide strategic and operational services to the FAS, PBS, GSA IT, OGP, and OCFO communities. We have a dedicated GSA Resource Hub to execute on tasks for GSA with GSA badged and experienced resources and SMEs. At GSA, Guidehouse is known for not only being an advisor but also bringing highly skilled operational performers.

# **Executive Summary of Solution:**

Our custom-built Guidehouse solution uses an easily navigable user interface so that contracting officers can upload EULA documents in either PDF or word document format. Once uploaded, our solution parses and divides documents into their contained clauses and then employs deep learning algorithms to classify each clause as Acceptable or Unacceptable with an associated prediction probability from 0 to 1 so that the user knows how confident they can be in the tool’s assessment, and where they may need to take a closer look. Guidehouse trained our algorithm to identify clauses as Acceptable or Unacceptable based on pre-processing of the training data set provided by GSA as well as multiple feature generation techniques, which identify aspects of unacceptable clauses and their potential significance. Guidehouse developed an ensemble model that uses a weighted voting classifier to vote on the predictions of the individual models for each clause to obtain the most popular prediction.

# **Guidehouse EULA Evaluation Tool (GEET) Architecture:**

## **Technology Scope:**

*At a high level, what technologies does the solution use? (e.g. language, frameworks).*

* This solution uses the Streamlit open-source app framework as the user interface. PyPDF3 and python-docx were used to parse through the EULA documents. Python libraries such as NLTK, sklearn, and Pytorch are used to generate features and build the deep learning algorithms. During the development phase, processed data was stored in an AWS S3 bucket for data storage.

## **Functionality and User Interface:**

*What type of user interface does the solution provide (e.g. web interface, command line interface)? What input formats does the solution support? (e.g. PDF or MS Word)? How does the solution process batches of documents?*

* The GEET interface is a simple, user-friendly application based on the Streamlit open-source app framework for users to input end user license agreement (EULA) documents to receive predictions on individual clauses of text on whether they are compliant with EULA standards. The backend of this interface contains the trained models that read in EULA documents in either PDF or word document form.
* Inputted EULA documents are parsed and divided into their containing clauses. This derived data frame is fed into the trained machine learning model to classify each clause as either acceptable or unacceptable as well as output a prediction probability from 0 to 1, 0 being a 0% probability that the clause is unacceptable and 1 being a 100% chance probability that the clause is unacceptable.
* Users will see indicators that the model is running and when it is finished, and results are displayed directly in the application in a table that shows each clause in the document, its classification as 0 or 1 (acceptable or unacceptable), and the confidence score associated with the classification.
* Users can upload multiple documents in both .docx and .pdf format to the application and our trained models on the backend will parse them, analyze the clauses, and separate results, first by document name and then organized by clause.

## **Application of Artificial Intelligence/Machine Learning (AI/ML):**

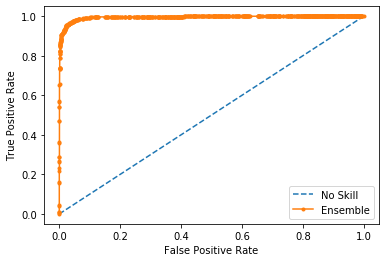
*Provide a description of the ways in which the technology leverages AI/ML. Please specify general approaches (supervised, unsupervised) and conceptual description of how these apply to the challenge.*

This solution provides clause acceptance predictions of a given EULA document in PDF or word document format, utilizing machine learning techniques at every stage, including data processing, feature generation, prediction, and interpretability analysis. Specific applications are discussed within each stage:

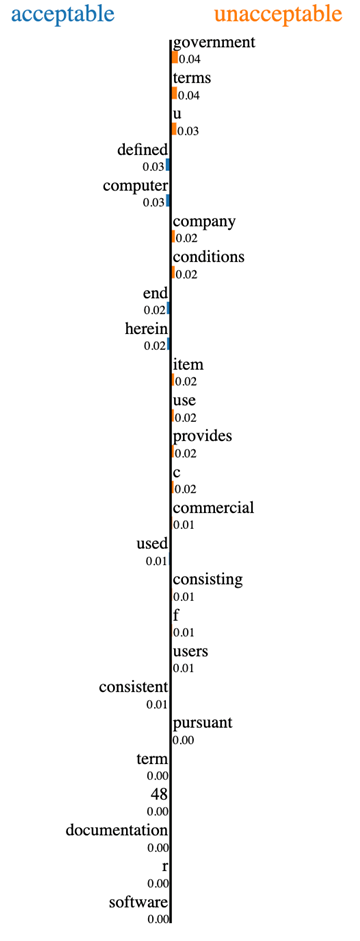
**Data Preprocessing:** The Guidehouse team preprocessed the AI\_ML\_Challenge\_Training\_Data\_Set\_1\_v1.csv using natural language processing (NLP) packages in Python before performing feature generation. We tested several techniques including special character removal, stop word removal, non-English word removal, stemming, lemmatization, and contraction expansion. We combined the preprocessing techniques with different feature generation techniques and tested the combinations on several machine learning classification models. Based on model performance, the optimal preprocessing techniques used included removing punctuation, special characters, section headings (e.g. IV. Or 3.), stop words, and non-English words using NLTK.

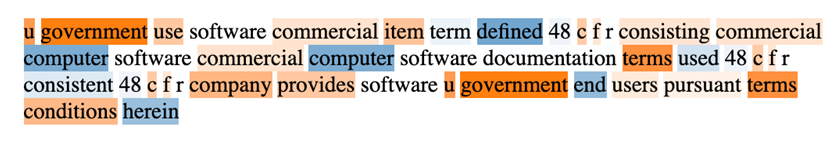
**Feature Generation:** The Guidehouse team tested multiple feature generation techniques including term frequency-inverse document frequency (TF-IDF), principal components analysis (PCA), Word2Vec, Universal Sentence Encoding (Cer et al., 2018), and bidirectional encoder representations from transformers (BERT). TF-IDF, PCA, and Word2Vec vectorization used only the limited training data provided, while the Universal Sentence Encoding and BERT methods employ transfer learning due to being pretrained on massive datasets. Given the small set of training data and variety in clause vocabulary, both transfer learning techniques performed better than the other methods. We selected a smaller version of BERT, DistilBERT (Sanh, Debut, & Chaumond, 2019) for efficiency and performance compared to Universal Sentence Encoding. We used the pretrained DistilBERT to extract document level numeric representation vectors for each clause in the training data. The team used these vectors as input into each of the classification models we tested.

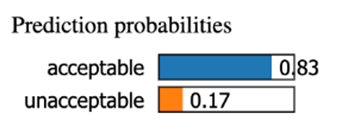
**Model Execution:** Multiple supervised classification methods were fit with the generated features. The following models were tested individually: logistic regression, K-nearest neighbors, random forest, and a support vector machine. The specific hyperparameters that distinguish models and help them train and perform more accurately were fine-tuned by testing a variety of combinations and assessed by evaluating F1 scores. After selecting the hyperparameters associated with the best F1 scores, we ensembled a naïve Bayes, along with each of the aforementioned models, to create a single supervised model. This ensemble model uses a weighted voting classifier to vote on the predictions of the individual models for each clause to obtain the most popular prediction. Each classifier captured different aspects of the underlying data generating process, so combining them allowed for better overall predictions relative to any one of them individually. We tested the final ensemble model on the given training data and obtained an F1 score of 0.88, a Brier Score of 0.04, and a Receiver Operating Characteristic (ROC) area under the curve (AUC) of 0.995. Because we were given a limited training set, we did not allocate any of the training data as a hold-out set, and all the training data was used to fit the model. Out of sample testing would be needed to test the model’s generalizability (the validation data labels were not provided and are assumed to be evaluated by GSA).

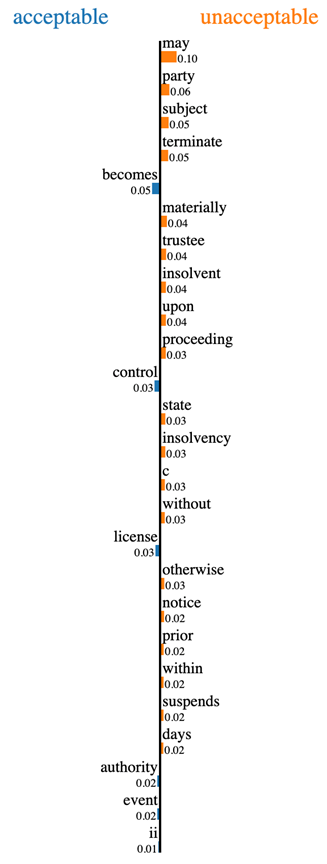


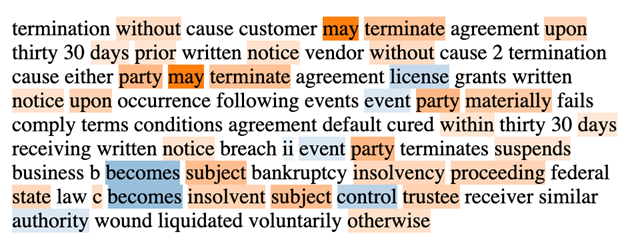
**Prediction Explanations:** We achieved transparency in model processes using Local Interpretable Model-agnostic Explanations (LIME). LIME trains an interpretable model by altering inputs, in this case words in a clause, and observing the change in the primary models' prediction (Ribeiro, Singh, & Guestrin, 2016). If an important word is altered, the LIME model would see a large change in the outcome of the classification model which signals its importance in the prediction of a given clause. This allows estimating the relative significance of input words to the final predicted status of a clause. We used a LIME model to output the relative importance of up to 25 words for a given preprocessed input clause in the training data. The pipeline creation and explainer algorithm training can be found in the project repository on Github, along with 100 training and 200 validation prediction explanations. Examples of both acceptable and unacceptable clause classifications on the validation data output by the explainer model can be found on the following pages.

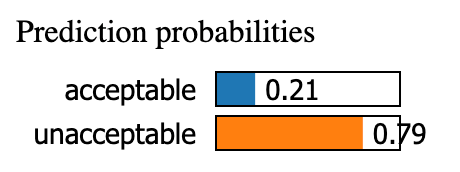
**Validation Data Explanation – Acceptable**

Clause 1302 from the validation data in its raw form is: “11.U.S. GOVERNMENT USE. The Software is a "commercial item" as that term is defined at 48 C.F.R. ¬ß 2.101, consisting of "commercial computer software" and "commercial computer software documentation" as such terms are used in 48 C.F.R. ¬ß 12.212. Consistent with 48 C.F.R. ¬ß 12.212, Company provides the Software to U.S. Government end users only pursuant to the terms and conditions herein.” Once the text is cleaned, using the methods described above, the clause looks as follows below. The highlighting of the words signifies their relative importance to the model’s prediction. The explainability algorithm is trained to predict both classes (with a sum of 1) versus the predicted probability 0-1 with 1 being unacceptable as in the actual model used in analysis. While the pipeline model and output probabilities are not identical to the actual models used, they are representative and can provide transparency into the complex featurization and classification of the clauses.



**Validation Data Explanation – Unacceptable**

Clause 7755 from the validation data in its raw form is: "Termination without Cause. Customer may terminate this Agreement upon thirty (30) days' prior written notice to Vendor, with or without cause.  
14.2.2. Termination with Cause. Either party may terminate this Agreement and its license grants by written notice upon the occurrence of any of the following events: (i) in the event the other party materially fails to comply with any of the terms and conditions of this Agreement and such default has not been cured within thirty (30) days after receiving written notice of the breach; or (ii) in the event the other party (A) terminates or suspends its business, (B) becomes subject to any bankruptcy or insolvency proceeding under Federal or state law, (C) becomes insolvent or subject to control by a trustee, receiver or similar authority, or (D) has wound up or liquidated, voluntarily or otherwise." As in the previous example, the explainer outputs the relative significance of the top 25 significant words within the cleaned clause, as well as predicted probabilities for each class.



**References**

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Cespedes, M, Yuan, S., Tar, C., Sung, Y., Strope, B., & Kurzweil, R. (2018). Universal sentence encoder. arXiv:1803.11175 [cs.CL].

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 [cs.CL].

Ribeiro, M.T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?” Explaining the predictions of any classifier. arXiv:1602.04938 [cs.LG].

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper, and lighter. arXiv:1910.01108 [cs.CL].