Group 7: Alan Lin , Trevor Petrin , Ganesh Kumar Ramar , Mingyu Chen

Repository: alanklin/AA-Capstone: Boston College Applied Analytics Capstone Project

Week 14 Assignment

This week the group shall review the project of the analytics project repository. Their team consists of Anita Gee, Sorachat Chavalvechaku, and Siwei Guo. Our team will provide individual feedback for each project to avoid potential overlap within a sectioned report.

Alan Lin

Just wanted to say first off, I was looking most forward to reading through your project at the end of the semester; it was one of the more interesting project proposals when we all met up at that office hours session.

In terms of the Github Repository structure, everything looks tidy. Notebooks and their outputs are saved together which isn't a big problem, but they could definitely be saved in a separate folder to avoid any confusion. In addition, the naming convention was slightly inconsistent, looking at Week 4 and 5 (seems to have forgotten the week_codeoutput scheme), but this is a really minor detail. Although the data folder is understandably in the .gitignore folder, the Combined Data.csv file doesn't reside in any of the folders within the data folder. It is pulled into your notebook by 'data/Combined Data.csv' which suggests that those folders aren't being used, so perhaps you can remove them to clean up the repository a bit more. Models are stored well in pickle format, making it easy to load in future runs. References folder appears to be empty, as well, so perhaps you can cull it from your repo.

For your notebook/code, feel free to look through our repo and see how you might be able to implement a _Setup.py file that can be used to install Python packages, import them, and create global variables that you can use in your notebooks. This should mitigate any issues that arise from sharing your notebook, as I recall your team had some issues that Elif mentioned. Moving onwards, EDA and Train-Test split are reasonably performed. Words clouds are always a nice visual but I would recommend providing more insights as to what they convey to you, especially in terms of the common terms that you mentioned. Your data preprocessing is thorough so I

don’t have much comment there; although I will mention that keeping stopwords is a good approach. Too often, stopword removal is applied inappropriately but I think you’re right in saying that you would lose a lot of contextual information key to this project. I also enjoyed reading your reasoning behind your choices, although this may be much more useful on a research style paper, rather than a Jupyter Notebook. To me, this notebook is a comprehensive callback to your semester’s work, so it does feel like it has some unnecessary code chunks in there that could be removed, such as evaluating multiple XGBoost models instead of just providing your final best performing model. It would be wise to mention this training and fine tuning step in the report only and detail how you came to your best model, but would strongly urge you to create an end-to-end pipeline that focuses only on the most important parts. This will make it look much more professional for job-seeking purposes, rather than looking like a school assignment. GPUs can be freely accessed with Google Colab so I highly recommend exploring that, as well since you mention the potential of this project with GPUs.
Your written report was also very thorough and covers every topic. Thinking ahead, your models work well with the current dataset but how would you make it into an actual product? For example, web scraping for more data and writing functions that parse and ensure it is in the correct format for your pipeline. Or even incorporating some sort of web-scraped test set that you can show off with predictions rather than the test split. I think you have a really promising project here that could be developed further to have real-time capabilities.

Again, great work and best of luck in your future projects/endeavors. I’ve learned a lot from your repository and report, and I hope you can say the same with ours!

Trevor Petrin

Hi Team!

First off very good documentation that you all have for the week 13 report - very professional format and good practice using the appendix. Very well done addressing the problem statement as to why your model adds value, very noble use case! The data sources you pulled from were reasonable as well, so no issue with that. No issue with anything with the train/validation/test split either. Good practice there.

As for the text preprocessing, one thing I noticed is that the approach of the TF-IDF was used, which I thought for the more mathematical models was a good approach for SVM and XGBoost. However, one thing I would recommend here would be taking the BERT approach for both the vector embeddings to get input and for output classification. It really is astounding how well it can perform with more naive training procedures to avoid some of the mundane data cleaning procedures such as removing stop words. Performance-wise as well, using the model in practice I was surprised how well it was able to perform when applied to sentiment problems in the past, it would be interesting to see how it would perform here!

As for the synthetic data and model sampling, I thought it was a solid approach to use SMOTE here with the TF-IDF approach there, thought that was a really creative way to go about that.

It would be interesting to see here the class imbalance (I know it's a small report so that may have been mentioned in an earlier week) to see which classes were underrepresented and the overall distribution of the classes. Good job touching more on that in the validation as well. But interesting note on the SMOTE, I thought it was great intuition on increasing the minority classes but it would have been good to understand the %change number in the report if there was a science to the way those numbers were chosen.

Nice improvements via the data-centric model development! That was a reasonable sacrifice to boost the classes. Well done acknowledging the bias within the model as well, the class imbalance was certainly the biggest issue here.

Well done with the deployment, good practice using the .pkl file for the lightweight storage, makes it easy for distribution and usage. I agree with the intuition behind the immediate feedback, batch feedback would take too long to get an answer for these patients, especially those who are immediately at risk.

Good way to bring up the KPI’s as well, that would be a great benchmark for the model to be continued to be measured against for continuous improvement and to signal when to retrain. Great place to have the model, overall well done!

Great job with the markdown-generated report as well. We probably should have taken a page out of your playbook for that one, great job!

Overall, great job Team! Very well done, it is clear that you are passionate about this subject and did a great job producing a presentable deliverable. Good luck in the professional world, you will all do great!

**Saw the distribution of the classes was included in the final Markdown file

Ganesh Kumar Ramar

I had a chance to go through your GitHub repository and I just wanted to say great job on the overall structure. You've clearly put thought into organizing the project in a way that's both standardized and easy to follow. I really appreciated how you used a common directory layout with clear folders for raw, interim, processed, and external data, as well as separating out your scripts, notebooks, and reports. That kind of structure makes collaboration and reproducibility much smoother. The inclusion of Makefile, setup.py, and requirements.txt was great to see, it shows you're thinking about environment setup and execution in a very professional way. I also noticed you included a tox.ini file, which is awesome, it’s not something everyone includes, but it’s a strong indicator that you're considering testing and code quality. Overall, your repo gives a very clean and mature impression, and it sets a strong foundation for scaling or collaborating on the project.

Your project on Sentiment Analysis on Mental Health Data is both impactful and relevant, with a clear goal of classifying statements into "Supportive," "Unsupportive," and "Neutral." The exploratory data analysis (EDA) provides useful insights into missing values, class distribution, and word counts, and the addition of the statement_len feature is a thoughtful step that could aid

future modeling. However, the dataset appears imbalanced, which should be addressed through methods like class weights, resampling, or augmentation to improve model training and avoid bias. Including representative text samples for each sentiment category would also help with interpretability and context.

The data cleaning process is solid, with attention given to removing duplicates and handling outliers, especially considering that extreme sentiment can be valuable in the mental health domain. The metadata features, like sentence and character lengths, provide useful structural patterns that could further enhance model performance. Your text normalization process is thorough, and the choice of tokenization and stemming methods should improve data quality and generalization in the model.

I think XGBoost is a great choice for sentiment analysis, especially considering its ability to handle imbalanced data and noisy text. However, the model shows signs of overfitting, with training accuracy much higher than validation accuracy. You've already taken steps to address this by incorporating SMOTE, adding the statement length as a feature, and removing outliers, these are all good strategies that could help generalize the model further. The model's performance on the test dataset is decent (0.79 accuracy), but there is a noticeable gap between training and test accuracy, which suggests that further tuning or regularization may be needed to improve generalization.

The evaluation metrics you used, like the accuracy, confusion matrix, and classification report, provide a comprehensive understanding of model performance. The confusion matrix heatmap is a useful visualization, but the low recall for certain classes (e.g., "Personality disorder") indicates that the model may struggle with rare categories. You might want to explore oversampling, undersampling, or adjusting class weights to address this.

Your approach to feature importance extraction using XGBoost’s built-in tools is excellent. Visualizing the top features, including mental health-related terms like "bipolar" and "avpd," adds valuable context to the analysis. These insights could be used to refine the model and explore which features are driving the predictions.

Overall, I think this project is off to a great start. You've clearly put a lot of thought into the structure of the repository, ensuring that it is well-organized and easy to navigate. The preprocessing pipeline is well-constructed, and I appreciate the attention to detail in cleaning the dataset and adding features that could potentially enhance model performance. I also really liked your choice of XGBoost for sentiment analysis, and your reasoning behind it is sound, especially with regard to handling imbalanced data. The evaluation metrics you used provide a solid understanding of the model's strengths and areas for improvement, although there's still room for improvement in terms of generalization. The inclusion of ethical considerations around mental health data is crucial, and it would be great to see more focus on that as the project evolves. Overall, you've done a solid job, and with some further refinements, I believe this project has a lot of potential. Keep up the great work!

Mingyu Chen