The performance of the Recurrent Neural Network model that we chose with an LSTM for the deep learning portion certainty did not yield the most accurate results in terms of prediction accuracy. At the end of training the validation accuracy was < 25% after training for 5 epochs. The reason why an LSTM might not be the best way to categorize this type of data is because recurrent neural networks try to look for dependencies between points in data like in a text subsequence. With an LSTM there is a forget gate that controls how much information from the previous cell states is passed on and propagated through the network. If there is too much information from previous cell states there will be an exploding gradient problem. So this helps keep a balance of new and old information.

With this type of neural network architecture in mind, the struggles from the model to categorize this type of data based on categories rather than sequences of data points is evident. This type of categorical data might benefit more from a different type of model that may not necessarily be for deep learning. Even with the vast array of rows in the dataset available, the LSTM was not able to gain any sort of valuable insight into the patterns of the data, even when extracting the most important features. LSTMs can struggle when dealing with data that has many dimensions like these types of categories, and especially data without any sort of sequence. Even with many epochs, the training loss did not decrease and a low training/validation accuracy remained as well. Perhaps another model like a random forest classifier or XGBoost would be better suited for this sort of task. In addition to this, things like decision trees, logistic regression, and other types of simple architectures could work. In addition, these types of models are simpler and more cost effective to train, in addition to yielding higher performance on categorical data types. Other future considerations could be training a larger model, perhaps with more computational power for inference from an external GPU.

Feature importance helps identify the most influential features in a model's predictions, improving optimization and interpretability. Normalised importance, derived from the absolute weights of model features, provides a straightforward approach to rank feature contributions but is model-dependent and doesn’t capture the true impact on performance. In contrast, techniques like permutation importance or SHAP values directly measure the effect of each feature on the model's accuracy by analysing performance drops when feature values are shuffled. For example, our analysis highlighted that features like 'Location Description: Street' and 'Apartment' had the highest impact, while environmental factors such as 'Snow' and 'Precipitation' had negligible importance. However, these methods are computationally intensive and sensitive to dataset distribution, which is a consideration for large datasets.

Our analysis revealed limitations in our dataset, including redundant features, significant noise unrelated to crime, and over 50% of features with SHAP values of zero. Neither LSTM nor Logistic Regression achieved the desired accuracy, indicating the need for more sophisticated models like XGBoost or Random Forest Classifiers to handle non-linear relationships. Additionally, crime is a complex phenomenon influenced by socioeconomic and political factors. Incorporating data on poverty levels, hardship indices, and political indicators could provide valuable context and improve analysis, paving the way for more accurate and insightful predictions.