Model Selection

Predicting Bacterial Property

1. Random Forest Algorithm: A suitable choice for dealing with bacterial property datasets is the Random Forest algorithm, which is built upon the Decision Tree Classifier model [1]. Random Forest has been proven to provide high accuracy and is computationally efficient, even on less powerful computers [2].

2. Gradient Boosting Machine (GBM): GBM has the potential to provide better accuracy than Random Forest by reducing bias and improving generalization. It is an ensemble method that builds models sequentially, where each subsequent model corrects the errors of the previous models [3]. However, GBM tends to be computationally more expensive and slower to train, especially on larger datasets. Training GBM on a large dataset may become infeasible or impractical with limited computational resources.

3. Deep learning-based models: Deep learning models, such as neural networks, can be used for predicting bacterial properties if access to a GPU is available. However, for the given dataset size, the potential accuracy gains may not justify the additional complexity and training time compared to Random Forest or GBM.

Predicting Number of People on the Beach

1. Regression Model: A regression model would be a bad choice for predicting the number of people on the beach due to the non-linear relationship between weather features and crowd size, as well as the potential interactions between categorical features.

2. Random Forest or GBM: Random Forest or GBM would be suitable for this task as they can handle both numerical and categorical features, capture interactions between variables, and model non-linear relationships effectively. Random Forest can handle a large number of features without the need for explicit feature engineering or normalization. While deep

learning methods can also solve the GBM problem, the trade-off between high computation cost and minimal gain in accuracy may vary from problem to problem [2], [4].

Text-Image Search Engine

1. CNN + RNN: A combination of CNN architecture (such as ResNet or VGG) for image processing and an RNN model (such as LSTM) for generating text descriptions can be a good choice for a text-image search engine. The accuracy of these models depends on factors such as the size and quality of the training dataset, complexity of the relationships between text and images, and the chosen architecture and hyperparameters of the model.

2. Paper [5]: In this paper, the authors propose an image captioning model that generates textual descriptions of images. The model consists of an encoder-decoder framework, where a deep convolutional neural network (CNN) is used as the encoder to extract visual features from the input image, and a long short-term memory (LSTM) network is used as the decoder to generate the textual descriptions.

Matrix Multiplication

1. Traditional Approach: When optimizing for speed in matrix multiplication, especially with limited computational resources, a traditional approach using highly optimized linear algebra libraries like Intel MKL or OpenBLAS would be preferable. These libraries leverage efficient matrix multiplication algorithms and hardware optimizations.

2. Deep Learning Models: If accuracy is the primary concern and the dataset size is sufficient, deep learning models like neural networks can be employed for matrix multiplication tasks. Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) can be adapted for matrix multiplication. Utilizing libraries like TensorFlow or PyTorch, which offer GPU acceleration, can significantly speed up computation if a GPU is available. However, traditional linear algebra libraries are often faster than deep learning models due to their optimized implementations. Deep learning models are advantageous for tasks involving complex patterns or learning from the data, but they may not provide substantial improvement in accuracy compared to optimized traditional methods for matrix multiplication.

Bibliography

[1] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[2] R. Aljarbou and J. Li. "Classification of large datasets using random forests." (2013), [Online]. Available: `http://fac.ksu.edu.sa/sites/default/files/classification_of_large_datasets_using_random.pdf`.

[3] A. Kapoor. "Gradient boosting from scratch." (2017), [Online]. Available: `https://blog.mlreview.com/gradient-boosting-from-scratch-1e317ae4587d`.

[4] A. Kumar. "Random forest algorithm clearly explained!" (2016), [Online]. Available: `https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/`.

[5] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3156–3164, 2015.