CS777 – Term Project Proposal Submission Template

**Student: Srivatsav Shrikanth**

1. Data set description: Provide a detailed description of the public data set you have selected, including its source, format, and any relevant details about the data.

|  |
| --- |
| *I plan to use Yelp Open Dataset and restrict the scope to Canadian cities – primarily Toronto, Montreal, and Vancouver. The dataset includes businesses (metadata such as name, categories, price range, latitude/longitude), users, check-ins, and millions of reviews with free-text and star ratings. Files are provided as JSON and are well-suited to Spark ingestion from cloud storage (Google Cloud Storage). The dataset is large enough for distributed processing and can be scaled by adding additional geographies or newer dumps over time.* |

1. Research question: Clearly define your research question and explain why studying is important. What do you want to learn from the data?

|  |
| --- |
| *How can we recommend the right restaurant to a Canadian user in a specific city based not only on rating history but also on the specific aspects they care about (things like food quality, service speed, ambiance, price etc.)?*  *We aim to understand the latent dimensions of preference in Canadian cities (unsupervised topics/aspects) and combine them with user item interactions to produce personalized, aspect-aware recommendations.* |

1. Machine Learning model: Specify the type of machine learning model you plan to use, such as classification or clustering, and explain why you have chosen this model.

|  |
| --- |
| *I plan to adopt a hybrid approach –*  *\* Collaborative Filtering (ALS) : Spark ML ALS on user–restaurant interactions (ratings) to learn latent factors.*  *\* Unsupervised Topic Modeling : Spark ML LDA (or NMF) over review text to extract aspects (e.g., service, taste, ambiance, value etc.).*  *\* Semi-Supervised Sentiment: Starting with lexicon-based sentiment (weak supervision) to produce aspect level sentiment scores and improve using self training with a small labeled subset (Spark pipeline).*  *\* Clustering: K-Means (or mini-batch K-Means appropriately) to segment users by preference and restaurants by experience profiles - clusters become features for the hybrid recommender.*  *\* Hybrid Ranker: Merge ALS scores with content features (topics, aspect sentiment, price, neighborhood etc) for final ranking.* |

1. Expected outcomes: What do you expect to achieve after implementing your learning model? What do you hope to learn or discover from your data analysis?

|  |
| --- |
| *A ranked recommendations API (not necessarily deployed, but with a clear interface) for Toronto/Montreal/Vancouver.*  *Insight reports: top aspects driving satisfaction by city, neighborhood persona clusters, and user taste segments.*  *Demonstrated scalability via GCP Dataproc + GCS - PySpark code and pipelines.*  *A GitHub repo that looks like a real product : data pipelines, model training, evaluation notebooks, and an architecture diagram (if possible and I can achieve it within deadline)* |

1. Evaluation plan: Explain how you plan to evaluate your project and assess the correctness of your model. What metrics or methods will you use to evaluate the effectiveness of your learning model? How well do you expect the model to work, and how will you measure its performance?

|  |
| --- |
| *Recommender quality:*  *Precision, Recall, MAP, NDCG (holdout by time or random split).*  *Cold start slice (new users / businesses) to show hybrid advantage.*  *Topic/Sentiment quality:*  *Topic coherence (UMass/NP-PMI) and manual interpretability checks.*  *Aspect-sentiment sanity checks on a hand-labeled sample / report accuracy/F1.*  *Clustering quality:*  *Silhouette score and stability under resampling.*  *Business face validity (e.g. - cluster descriptors that make sense: “cheap-&-fast,” “date-night,” “brunch-hotspots”).*  *Data splits & rigor:*  *Train / validation / test via Spark & cross-validated hyper parameters for ALS (rank, regParam and alpha).*  *Ablations: ALS only vs Hybrid - no-sentiment vs with sentiment / no-topics vs with topics.*  *Expected performance:*  *Hybrid desired model should outperform ALS only on NDCG and MAP (especially on cold start and niche tastes).*  *Topic coherence in acceptable range = sentiment accuracy competitive for weak supervision*  *Environment:*  *\* PySpark on GCP Dataproc, data in Google Cloud Storage, will try to incorporate BigQuery for analytics tables if time permits*  *\* Reproducible runs via make/shell scripts and README.md with clear steps.* |