HW4 Group GWAR Assignment: Create PERT and Gantt charts for Job Scam Email Detection course project

Group 3

Laxmi Thrishitha Kalvakota - 017605640
Janani Kripa Manoharan - 016721159
Shivani Atul Beri - 018205018
Sreenidhi Hayagreevan - 018195489

Masters in Data Analytics, San Jose State University
DATA245 - Machine Learning
Vishnu S. Pendyala
April 11, 2025

Part 1 - PERT Chart

1. Tasks:

1. Finalize problem statement
2. Identify and prepare data (Find data, synthesize data for additional fields)
3. EDA
4. Discard useless features and retain unnecessary fields
5. Model development - DecisionTree, RandomForest
6. Model development - SVM, kNN
7. Model development - Naive Baye's classification and Logistic regression
8. Model development - LSTM and XGBoost
9. Evaluate and finetune
10. Final Evaluation and Model comparison report

2. Dependencies explained:

Steps 1, 2, 3 and 4 are sequential. First, the problem statement is decided and then the data is identified / synthesized / prepared based on the problem statement. After the data is prepared, Exploratory Data Analysis (EDA) is done. Upon studying the distributions of the data, correlations, etc., we decide what features need to be retained and what features can be discarded. After all the processing, once we have the final dataset with which we can train the models, we should be able to train different models in parallel.

Since it's 4 of us in the team, 5, 6, 7 and 8 can be done in parallel. Once the models are trained, they will be compared with each other and finetuned further. After all the steps, the final model comparison report will be generated.

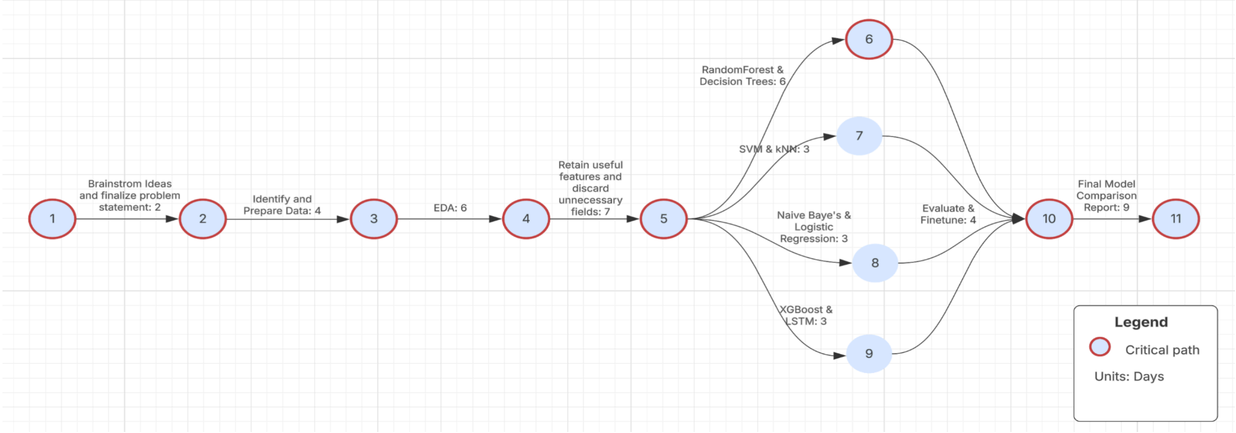
3. Tasks and their estimated durations (Color-coded to match the Gantt milestones & durations):

Task	Duration
1. Finalize problem statement	2 days
2. Identify and prepare data (Find data, synthesize data for additional fields)	4 days
3. EDA	6 days
4. Discard useless features and retain unnecessary fields (Feature Engg)	7 days
5. Model development - DecisionTree, RandomForest	6 days
6. Model development - SVM, kNN	3 days
7. Model development - Naive Baye's classification and Logistic regression	3 days
8. Model development - LSTM and XGBoost	3 days
9. Evaluate and finetune (Model Comparison)	4 days
10. Final evaluation and Model Comparison Report	9 days

4. PERT Chart:
Simple PERT Chart:

Fig1.

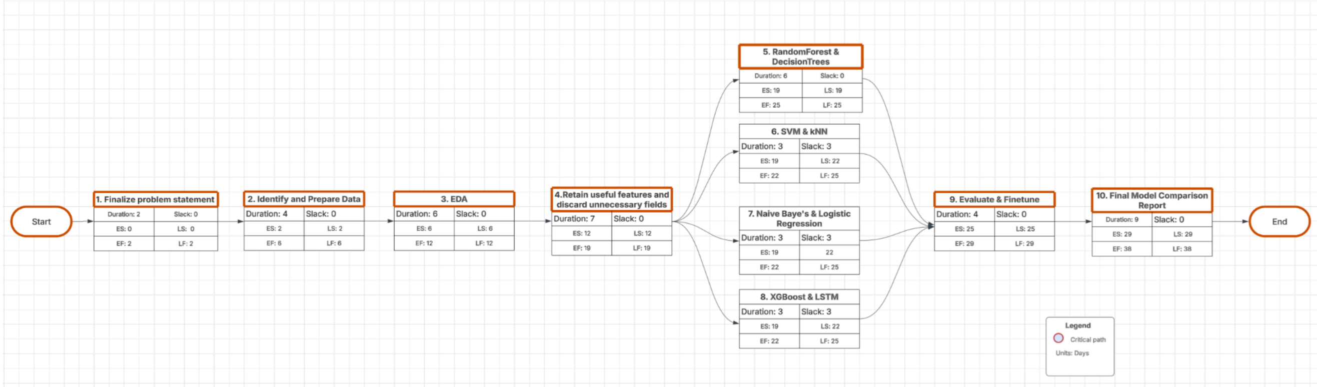
Simple PERT Chart depiction - Lucidchart document


6
RandomForest & Decision Trees: 6
7
SVM & kNN: 3
Brainstrom Ideas and finalize problem statement: 2
Identify and Prepare Data: 4
EDA: 6
Retain useful features and discard unnecessary fields: 7
Naive Baye's & Logistic Regression: 3
Evaluate & Finetune: 4
Final Model Comparison Report: 9
1
2
3
4
5
10
11
8
XGBoost & LSTM: 3
9
Legend
Critical path
Units: Days

Network PERT Chart:

Fig2.

Network PERT Chart depiction Lucidchart document



The critical path is highlighted in maroon that takes the longest time.

5. Calculation of Expected Duration:

Expected Duration (ED) is calculated using the following formula. Here, all the values are given in *days*.

Expected Duration = (Optimistic Time + 4 * Most Likely Time + Pessimistic Time) / 6

$ED = (O + 4m + P) / 6$

	Task	O	M	P	ExpectedDuration(days)
1	Finalize problem statement	1	2	3	(1 + 8 + 3) / 6 = 2.0
2	Identify and prepare data	3	4	6	(3 + 16 + 6) / 6 = 4.17
3	EDA	4	6	8	(4 + 24 + 8) / 6 = 6.0

4	Discard useless features, retain necessary fields	5	7	9	(5 + 28 + 9) / 6 = 7.0
5	Train - DecisionTree, RandomForest	4	6	8	(4 + 24 + 8) / 6 = 6.0
6	Train - SVM, kNN	2	3	5	(2 + 12 + 5) / 6 = 3.17
7	Train - Naive Baye's, Logistic Regression	2	3	5	(2 + 12 + 5) / 6 = 3.17
8	Train - LSTM, XGBoost	2	3	5	(2 + 12 + 5) / 6 = 3.17
9	Evaluate and fine-tune	3	4	6	(3 + 16 + 6) / 6 = 4.17
10	Final model comparison report	7	9	11	(7 + 36 + 11) / 6 = 9.0

Expected Duration of common parts of the path = 1, 2, 3, 4, 9, 10 = 2 + 4.17 + 6 + 7 + 4.17 + 9 = 32.34 days.

For paths with 5,6,7,8 the corresponding Expected Durations are 38.34, 35. 34, 35.34 and 35.34 days respectively.

6. Critical Path:

In PERT, the critical path is the longest sequence of dependent tasks that determines the minimum project completion time; any delay in these tasks will directly delay the project's overall completion. In other words, the critical path consists of activities where the difference between their earliest and latest start or finish times is zero.

Here, the critical path with the expected duration 38.34 days is given by:
1 → 2 → 3 → 4 → **5** → 9 → 10

The completion of (the longest sequence) tasks that involve Random Forest and Decision Trees is critical for the project to finish on schedule. Since most of the tasks here are sequential / dependent on the previous, there are hardly any slack days.

7. Guidelines:

- **Notation and Dependencies**

  Simple and network PERT charts have been created to represent the task flow of the project. Each task is represented as a circular node in the simple PERT chart and as a table with the durations in the Network Diagram and the dependencies are shown with directional arrows. Tasks 1 to 4 are sequential, and once the data preparation and EDA are over, model training (5 to 8) can be executed in parallel by different team members. This parallelism reduces project duration. Tasks 9 and 10 are again sequential and depend on outputs from all model training tasks.
- **The PERT Formula:** Each task's duration was estimated using the PERT formula and tabulated under 5: $ED = (O + 4m + P) / 6$
- **Paths and their durations:**

  Critical Path: 1 → 2 → 3 → 4 → 5 → 9 → 10

  Duration = 2 + 4.17 + 6 + 7 + 6 + 4.17 + 9 = *38.34 days*.

  Other paths (with parallel tasks like 6, 7, 8) yield durations of *35.34 days*.

  The critical path has the longest duration and no slack time. Delays in any of the tasks on this path would delay the entire project.

Part 2: Creating a Gantt Chart

1. Defining our Project Scope

Objectives and Aims:

- Develop a lean ML model to classify job scam emails (binary: scam/legitimate).

Deliverables:

1. Sourced, cleaned and normalized dataset (17,000 samples).
2. A report on Exploratory Data Analysis (EDA).
3. Feature-engineered dataset (metadata features).
4. ML model trained (e.g., Logistic Regression, SVM or Naive Bayes).
5. Testing , deployment and estimating performance metric (precision, recall, F1-score).

Key Milestones:

Phase	Deadline	Deliverable
Sourcing data	Week 1	Enron dataset Kaggle + Survey Data + Data Synthesis
Data Preprocessing		Cleaned dataset + EDA
EDA	Week 2	Pattern Finding + Visualization charts
Feature Engineering		Feature set + feature vectors
Model Development and Optimization	Week 3	Trained ML model (e.g., SVM)
Evaluation and Reporting	Week 4	Performance metrics + demo

2. Break Down Tasks (Work Breakdown Structure)

Hierarchy:

1. Data Preparation

1.1 Merge & Align Datasets(Enron email dataset + Kaggle + survey).

1.2 Text Cleaning & NLP Prep

1.3 Label data for modeling purposes.

2. Exploratory Data Analysis (EDA)

2.1 Scam email pattern analysis (common words, sender domain).

2.2 Visualization of key trends (word clouds, frequency distributions).

3. Feature Engineering

3.1 Extract relevant features from the dataset.

3.2 Reduce dimensionality to improve model performance.

4. Model Development

4.1 Train Random Forest and Decision Tree model on processed data.

4.2 SVM and KNN model for comparison.

4.3 Naive Bayes and Logistic Regression model for comparison.

4.4 XGBoost and LSTM model for comparison.

4.5 Model comparison report

5. Evaluation and Reporting

5.1 Final Model Evaluation.

5.2 Competitor Research

3. Estimate Task Breakdown with Estimated Durations

Phase 1: Data Preparation

Task	Primary Owner	Support Team	Justification	Resource Availability	Task complexity	Potential Challenges
1.1 Merge & Align Datasets	Sreenidhi	Shivani	Sreenidhi sourced data; Shivani handles technical integration.	**Datasets**: Enron dataset (emails), Kaggle (scam-related text), Survey data (user-reported scam markers). **Tools**: Python (Pandas, NumPy), **(Colab)** **People**: Sreenidhi (primary), Shivani (support).	High	Data schema mismatches, missing values
1.2 Text Cleaning & NLP Prep	Shivani	Janani	Shivani leads NLP (stopwords, lemmatization); Janani supports regex/EDA.	**Tools**: python notebook**(google colab)**	High	Handling slang/abbreviations in text
1.3 Label Data	Sreenidhi	All	Sreenidhi defines criteria, team cross-validates labels.	**Dataset**: EDA-processed dataset.	Medium	Subjective labeling criteria

Phase 2: Exploratory Analysis

Task	Primary Owner	Support Team	Justification	Resource Availability	Task complexity	Potential Challenges
2.1 Scam Pattern Analysis (EDA)	Janani	Sreenidhi	Janani leads EDA; Sreenidhi provides scam markers from surveys.	**Dataset**: Cleaned dataset from Phase 1 (real+synthesized).	Medium	Identifying subtle scam patterns
2.2 Trend Visualization	Janani	Thrishitha	Janani creates HeatMaps/insights, Thrishitha formats for reports.	**Tools**: Google Colab, Python (Matplotlib, Seaborn, Plotly). **People**: Janani (primary), Sreenidhi (support for scam markers).	Low	Ensuring visual clarity for better understanding of underlying patterns

Phase 3: Feature Engineering

Task	Primary Owner	Support Team	Justification	Resource Availability	Task complexity	Potential Challenges
3.1 Feature Engineering (Extraction + PCA)	Sreenidhi	Janani	Merged tasks: Completed end-to-end feature optimization.	**Dataset**: EDA-processed dataset. **Tools**: Python (Scikit-learn, NLTK, PCA). **People**: Sreenidhi (primary), Janani (support).	High	Overfitting risk, PCA interpretation

Phase 4: Model Development and Optimization

Task	**Primary Owner**	**Support Team**	**Justification**	**Resource Availability**	**Task Complexity**	**Potential Challenges**
4.1 Train & Optimize SVM/KNN	Shivani	-	Shivani leads optimization. Optimization of SVM/KNN by various methods (eg. PCA)	**Dataset is common**: Feature-engineered dataset. **Tools are common** : Python (Scikit-learn, TensorFlow). **People: SVM/KNN:** Shivani.	High	Hyperparameter tuning complexity
4.2 Train Random Forest	Sreenidhi	-	High-complexity model requiring ML expertise.	**Random Forest:** Sreenidhi.	High	Computational resources for datasets
4.3 Train Naive Bayes	Janani	-	Lightweight model aligned with Janani's role.	**Naive Bayes:** Janani.	High	Oversimplification risk
4.4 Hyperparameter tuning	Shivani	-	Hyperparameter tuning, accuracy improvement, addition of features and its analysis.	Initial models from Phase 3 Tools: Python, TensorFlow Lite.	High	Long runtimes, overfitting, computation trade-offs.
4.5 Model Comparison Report	Janani	Sreenidhi, Shivani	Shivani compares SVM/KNN; others validate RF/NB metrics.	Coordinating inputs	Medium	Bias in model selection

Phase 5: Evaluation & Reporting

Task	Primary Owner	Support Team	Justification	Resource Availability	Task complexity	Potential Challenges
5.1 Final Model Evaluation	All	-	Team consensus for production model selection.	**Models**: Final optimized models. **Tools**: Python	High	Consensus delays
5.2 Competitor Research	Thrishitha	-	Non-technical benchmarking aligned with her role.	**Tools**: Google slides(reporting) **People**: All (consensus evaluation), Thrishitha	Medium	Limited public data on competitors

Final Gantt Chart Timeline

Phase	Start Day	End Day	Total Duration
1. Data Preparation	1	6	6 days
2. EDA	7	12	6 days
3. Feature Engineering	13	19	7 days
4. Model Development and Optimization	20	29	10 days
5. Evaluation and Reporting	30	38	9 days

4. Assign Responsibilities

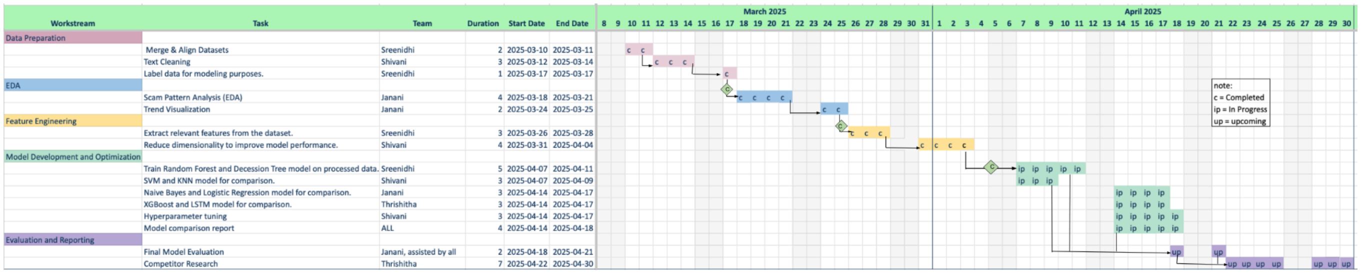
Task	Primary Owner	Support Team	Justification
1.1 Merge datasets	Sreenidhi	Shivani	Sreenidhi sourced all datasets; Shivani assists with technical integration handling data for accuracy, consistency, and completeness.
1.2 Text Cleaning	Shivani	Janani	Shivani leads NLP; Janani supports regex/EDA
1.3 Label data	Sreenidhi	All	Sreenidhi defines labeling criteria; team verifies.
2.1 Scam Pattern	Janani	Sreenidhi	Janani leads EDA; Sreenidhi advises on scam

Analysis			markers from survey data
2.2 Visualize trends	Janani	Thrishitha	Janani creates charts; Thrishitha formats visuals
3.1 Feature extraction	Sreenidhi	Janani	Sreenidhi owns ML pipeline; Janani assists with metadata features
3.2 PCA	Shivani	-	Shivani does PCA(critical for optimization)
4.1 Train RF	Sreenidhi	-	High-complexity task requiring ML expertise.
4.2 Train Bayes	Janani	-	Janani manages lightweight model training
4.3 Train SVM/KNN	Shivani	-	Shivani leads the training of SVM/KNN models, focusing on precision tuning and performance optimization for mid-to-high complexity classifiers
4.4 Train XGBoost and LSTM	Thrishitha	-	Thrishitha leads the training of XGBoost and LSTM models, emphasizing robust learning and sequential pattern recognition for advanced predictive tasks
4.5 Hyperparameter tuning	Shivani	Janani	Shivani leads model optimization; Janani tests configurations
4.6 Model Comparison Report	All	-	All members contribute to the Model Comparison Report, collaboratively evaluating model performance metrics to identify the most effective algorithms for deployment
5.1 Evaluate metrics	Janani	All	Janani owns the model metrics, and model selection is based on team consensus to ensure balanced and collaborative decision-making.
5.2 Competitor research	Thrishitha	-	Non-tech task aligned with Thrishitha's contributions

5. Create the Gantt Chart Gwar-_Gantt

Fig3.

Gantt Chart Depiction



Workstream	Task	Team	Duration	Start Date	End Date
Data Preparation					
	Merge & Align Datasets	Sreenidhi	2	2025-03-10	2025-03-11
	Text Cleaning	Shivani	3	2025-03-12	2025-03-14
	Label data for modeling purposes.	Sreenidhi	1	2025-03-17	2025-03-17
EDA					
	Scam Pattern Analysis (EDA)	Janani	4	2025-03-18	2025-03-21
	Trend Visualization	Janani	2	2025-03-24	2025-03-25
Feature Engineering					
	Extract relevant features from the dataset.	Sreenidhi	3	2025-03-26	2025-03-28
	Reduce dimensionality to improve model performance.	Shivani	4	2025-03-31	2025-04-04
Model Development and Optimization					
	Train Random Forest and Decession Tree model on processed data.	Sreenidhi	5	2025-04-07	2025-04-11
	SVM and KNN model for comparison.	Shivani	3	2025-04-07	2025-04-09
	Naive Bayes and Logistic Regression model for comparison.	Janani	3	2025-04-14	2025-04-17
	XGBoost and LSTM model for comparison.	Thrishitha	3	2025-04-14	2025-04-17
	Hyperparameter tuning	Shivani	3	2025-04-14	2025-04-17
	Model comparison report	ALL	4	2025-04-14	2025-04-18
Evaluation and Reporting					
	Final Model Evaluation	Janani, assisted by all	2	2025-04-18	2025-04-21
	Competitor Research	Thrishitha	7	2025-04-22	2025-04-30

note:
c = Completed
ip = In Progress
up = upcoming

6. Guidelines:

Guideline	**Implementation in Gantt Chart**
Clear & Consistent Format	● Color-coded phases: ○ Pink (Phase 1. Data Preparation), ○ Blue (Phase 2 . EDA), ○ Yellow (Phase 3. Feature Engineering), ○ Green (Phase 4. Model Dev), ○ Purple (Phase 5. Deployment) ● Uniform date format (YYYY-MM-DD)
Dependencies	● Arrows connect sequential tasks (Data Preparation → EDA→Feature Engineering → Model Development →Evaluation and Reporting) ● Green diamond (C) C marks phase completion milestones
Resource Allocation	**Owners & Resources Allocated**: ● **Sreenidhi**: ○ *Tasks*: Data Prep (1.1, 1.3), Feature Eng (3.1), Random Forest (4.2) ○ *Tools*: Python (Pandas, NumPy), Google Colab, Scikit-learn ● **Shivani**: ○ *Tasks*: Data Prep (1.2), SVM/KNN (4.1), Hyperparameter Tuning (4.4) ○ *Tools*: Python (NLTK, Optuna), TensorFlow Lite ● **Janani**: ○ *Tasks*: EDA (2.1, 2.2), Naive Bayes (4.3), Model Comparison (4.5) ○ *Tools*: Matplotlib, Seaborn, Plotly, Scikit-learn ● **Thrishitha**: ○ *Tasks*: Competitor Research (5.2) ○ *Tools*: Google Slides, Market Research APIs ● **All**: ○ *Tasks*: Final Model Evaluation (5.1) ○ *Tools*: Google slides(Presentation), Google Docs/Overleaf (Report Creation)
Progress Tracking	● Status markers: ○ c: Completed (Data Prep., EDA, Feature Eng.) ○ ip: In Progress (Model Dev) ○ up: Upcoming (Evaluation and Reporting)

Key Compliance with Requirements

1. **Format Clarity:** Each workstream has distinct colors and clear date ranges
2. **Dependency Mapping:** Data Prep → EDA→Feature Engineering → Model Development and optimization →Evaluation and Reporting.
3. **Update Readiness:** Status markers allow easy weekly progress updates.

Mandatory Question Response

Did you find or come across solutions to similar problems by using Generative AI or other sources?

Yes , we used Generative AI (specifically ChatGPT - GPT-4 OpenAI) to assist in designing and preparing Gantt and PERT charts. These are the following prompts that we used:

Prompt 1: “How do I make a PERT chart for my email classification and scam detection project?”
Help Provided: ChatGPT helped us to come up with our main phases in the workflow of our project: Data Gathering, Preprocessing (text cleaning, tokenization), Feature Engineering (TF-IDF, Embeddings), Model Training (logistic regression, XGboost), Evaluation and Deployment phases. ChatGPT aided us with laying dependencies and estimating possible durations using the PERT formulas..

Prompt 2: “Would you help me to define a job scam email detection project to sort out the tasks for my Gantt chart?”
Help Provided: It helped us work out a project schedule for the project in task level breakups and duration and deadlines. We scheduled the tasks by week, and then allocated people for responsibility, and mapped each task in a graphic display to see overlaying and sequential activities for time management.

Prompt 3: “What dependencies and time estimates should I consider when building a machine learning pipeline for detecting scam emails?”
Help Provided: ChatGPT helped us reason out our activities, e.g., showing model training depends on preprocessing and evaluation depends on model selection, and estimating each.

References:

Sample Gantt Chart Project. (2024, September 25). TeamGantt.

https://www.teamgantt.com/what-is-a-gantt-chart

Asana, T. (2025, March 6). What Is a PERT Chart? Create One Now (with Examples) [2025] Asana. *Asana*. https://asana.com/resources/pert-chart

AbhishekGupta. (2021, May 17). *Seven rules for delivering machine learning projects on time*. Data Science Central. https://www.datasciencecentral.com/seven-rules-for-delivering-machine-learning-projects-on-time/