# The Food Desert Effect: Nutrition Inequality and Metabolic Health

## PERT and Gantt Charts

**Course: DATA 245 Machine Learning**
**Group 3**

Savitha Vijayarangan

Rishi Visweswar Boppana

Kapil Reddy Sanikommu

Jane Heng

# Part 1: PERT Chart

## 1. Identified Tasks

For our Food Desert Effect project, we broke down the work into 11 main tasks. We tried to make them specific enough to track but not so detailed that we'd get lost in the weeds.

| Task ID | Task Name | What We Need to Do |
|---|---|---|
| **A** | Data Collection | Download datasets from USDA, CDC, and Census Bureau |
| **B** | Data Cleaning | Fix missing values, normalize the data, make sure formats are consistent |
| **C** | Data Integration | Merge everything using FIPS codes (this is trickier than it sounds because of different geographic levels) |
| **D** | Feature Engineering | Build composite indices like Food Access Barrier Index and Socioeconomic Vulnerability Index |
| **E** | Exploratory Data Analysis | Run correlations, make plots, figure out what patterns exist |
| **F** | Regression Modeling | Train OLS, Ridge, and Lasso models with cross validation to find which works best |
| **G** | K Means Clustering | Group counties into nutritional risk clusters |
| **H** | PCA Analysis | Reduce dimensions so we can visualize the data better |
| **I** | MVS Computation | Calculate the Metabolic Vulnerability Score using our regression results |
| **J** | Visualization Dashboard | Build a Streamlit app so people can explore the data |
| **K** | Final Report | Write up everything and prepare presentation |

These tasks cover the full pipeline from raw data to final deliverables.

## 2. Task Dependencies

Some tasks have to happen in a certain order, while others can run at the same time. Here's what depends on what:

### Tasks that must happen sequentially:

Task B depends on Task A finishing first. This makes sense because we can't clean data we haven't downloaded yet.
Task C depends on Task B. We learned from our intermediate report that trying to merge dirty data just creates more problems, so cleaning has to come first.
Task D depends on Task C. We need the merged dataset before we can engineer new features from multiple data sources.

Task E also depends on Task C. Can't do exploratory analysis until we have all the data integrated.

#### Tasks where we have some flexibility:

Task F depends on both D and E. We need the engineered features AND we need insights from our exploratory analysis to know which features to include in the regression.
Task G depends on Task E. Once we have clean integrated data and understand it, we can run clustering.
Task H also depends on Task E. Same reasoning; we need to understand the data structure before doing PCA.

#### The convergence point:

I need F, G, AND H to all be done. This is where we combine everything into the Metabolic Vulnerability Score. It's kind of the bottleneck of the project but necessary.
Task J depends on Task I. The dashboard needs the MVS scores to display.
Task K depends on Tasks I and J. We can't write the final report until all analysis is done.
One nice thing we noticed is that Tasks F, G, and H can all run in parallel once Task E is done. This saves us about 3 days compared to doing them one after another.

## 3. Task Duration Estimates

We estimated how long each task will take using the PERT method with optimistic, most likely, and pessimistic scenarios:

| Task | Optimistic (days) | Most Likely (days) | Pessimistic (days) | Expected (days) |
|---|---|---|---|---|
| **A: Data Collection** | 2 | 3 | 4 | 3 |
| **B: Data Cleaning** | 3 | 4 | 6 | 4 |
| **C: Data Integration** | 2 | 3 | 5 | 3 |
| **D: Feature Engineering** | 2 | 3 | 4 | 3 |
| **E: EDA** | 1 | 2 | 3 | 2 |
| **F: Regression Modeling** | 3 | 4 | 6 | 4 |
| **G: K-Means Clustering** | 2 | 3 | 5 | 3 |
| **H: PCA Analysis** | 2 | 3 | 4 | 3 |
| **I: MVS Computation** | 1 | 2 | 3 | 2 |

| | | | | |
|---|---|---|---|---|
| **J: Dashboard** | 2 | 3 | 5 | 3 |
| **K: Final Report** | 1 | 2 | 3 | 2 |

**Why these estimates:**

For optimistic times, we assumed everything goes smoothly with no major roadblocks. Most likely times are based on our experience from other projects in this class and DATA 236. For pessimistic, we thought about worst case scenarios like discovering the datasets don't match up well or models performing poorly.
Task B has a wide range because data cleaning is unpredictable. We know from the intermediate report that about 12% of counties have missing data, but we won't know the full extent until we dig in.
Task F also has a large range because hyperparameter tuning can take longer than expected if the models don't converge well.

## 4. PERT Chart Network Diagram

The PERT chart below shows the network of tasks, their dependencies, and durations:

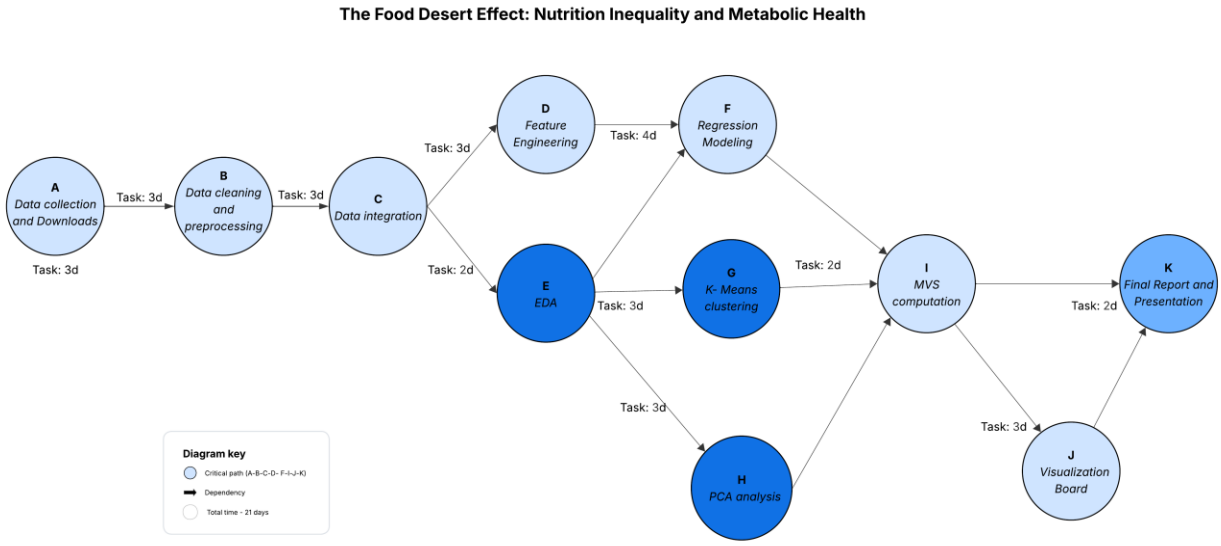


***Figure 1: PERT Chart showing task dependencies and durations***

The PERT chart should show all 11 tasks as nodes with arrows connecting them. Each node should have a task letter and duration (like "A: 3"). The critical path (A, B, C, D, F, I, K) should be highlighted in red or made bold, so it stands out.
We recommend using Lucidchart or Draw.io to make this. Both have templates that make it straightforward. The key is making sure all the dependency arrows are correct, and the critical path is obvious.

**Layout tips from our experience:**

- Put START on the left, END on the right
- Keep the critical path flowing through the middle
- Show F, G, and H branching off in parallel
- Make sure the arrows clearly show direction

## 5. Expected Duration Calculation

We used the PERT formula to calculate expected durations:
**Expected Duration = (Optimistic + 4 x Most Likely + Pessimistic) / 6**
The formula gives more weight to the most likely scenario, which makes sense.
**Example for Task B (Data Cleaning):**
Expected = (3 + 4×4 + 6) / 6 = (3 + 16 + 6) / 6 = 25 / 6 ≈ 4 days
**Total project timeline based on critical path:**
We added up all the tasks on the critical path:

| Critical Path Task | Duration |
|---|---|
| A: Data Collection | 3 days |
| B: Data Cleaning | 4 days |
| C: Data Integration | 3 days |
| D: Feature Engineering | 3 days |
| F: Regression Modeling | 4 days |
| I: MVS Computation | 2 days |
| K: Final Report | 2 days |
| **Total** | **21 days** |

So, we're looking at about 21 days minimum to finish the project if everything on the critical path goes according to plan.

## 6. Critical Path Analysis

The critical path is **A, B, C, D, F, I, K** with a total duration of 21 days.
This is important because any delay in these tasks will push back our entire project deadline. The other tasks (E, G, H, J) have some slack time, meaning they can be delayed a bit without affecting the final deadline.

### Breaking down the critical path:

Starting from Day 1:

- After Task A (Day 3): We have data downloaded
- After Task B (Day 7): Data is clean
- After Task C (Day 10): Everything is integrated
- After Task D (Day 13): Features are engineered
- After Task F (Day 17): Regression models are trained
- After Task I (Day 19): MVS scores calculated
- After Task K (Day 21): Project complete

### What about the non-critical tasks?

Task E (EDA) has some slack. It needs to finish before Task F starts, but Task C finishes at Day 10 and Task F doesn't start until Day 13, so we have buffer time.
Tasks G and H have slack too because they just need to finish before Task I starts at Day 17.
Task J (Dashboard) has flexibility between when Task I finishes (Day 19) and Task K needs it.

#### Why this matters for our team:

We need to make sure Savitha stays on track with Tasks A, B, C since they're all on the critical path. Jane needs to prioritize Task F over E if there's any conflict. The good news is Rishi can work on G and H without much time pressure as long as Task E gets done.

## 7. PERT Chart Guidelines

#### Using clear notation:

We're using standard PERT notation with circles for tasks and arrows for dependencies. Each task node shows the ID and expected duration. We kept it consistent throughout, so anyone looking at our chart can understand it.
The START and END nodes help show the project flow clearly.

#### Accounting for uncertainty:

The three point estimation method (optimistic, most likely, pessimistic) helps us deal with the uncertainty in our estimates. We can't know exactly how long things will take, especially with data quality issues.
Based on our intermediate status report, we know about 12% of counties have suppressed data that will need imputation. This is why we made Task B's pessimistic estimate for 6 days instead of 5.
For Task F, we're uncertain about model performance. If the R-squared is low initially, we might need extra time for feature engineering iteration or trying different model specifications. That's reflected in the 6 days pessimistic estimate.
Task C has uncertainty because merging census tract level data with county level data requires population weighting, and we might run into edge cases.

#### Focusing on the critical path:

Since Tasks A, B, C, D, F, I, and K form the critical path, these need the most attention. Any delay here delays everything.
Our strategy is:
Assign our most experienced person (Savitha) to the data tasks
Make sure Task F gets priority when Jane has to choose between tasks
Start non-critical tasks early if resources are available
Have buffer time built into the schedule
The parallel execution of F, G, and H is key to keeping the project timeline reasonable. If we had to do them sequentially, the project would take about 25 days instead of 22.

# Part 2: Gantt Chart

## 1. Project Scope Definition

### What we're trying to accomplish:

Our main goal is to understand how food access disparities affect metabolic health at the community level. We want to build a predictive model that can identify which communities are at highest risk.
Specifically, we want to:
Quantify the relationship between food deserts and health outcomes like diabetes and obesity
Create a Metabolic Vulnerability Score that public health officials can actually use
Find patterns in how different communities experience these issues
Make all this accessible through an interactive tool

### Concrete objectives:

We need to integrate three major datasets (USDA, CDC, Census) using FIPS codes. This is challenging because they're at different geographic levels.
We'll train regression models (OLS, Ridge, and Lasso) to see which predicts metabolic health outcomes best. We're aiming for R-squared above 0.60.
We'll use K-means clustering to group counties into distinct types. We want at least a 0.45 silhouette score to show the clusters are meaningful.
PCA will help us visualize high-dimensional data and understand which factors matter most.
Finally, we'll build a Streamlit dashboard so people can explore the data themselves.

### What we're delivering:

A cleaned dataset with 3000+ counties and 50+ integrated variables
Regression models with coefficient analysis
Cluster assignments with visualizations
PCA biplots
Interactive dashboard
Technical report (20-25 pages)
Executive summary (2 pages)
Presentation materials

### Timeline milestones:

Week 1 (by Day 7): All data collected, cleaned, and integrated Week 2 (by Day 14): Regression analysis complete, best model selected Week 3 (by Day 21): Clustering and PCA done Week 4 (by Day 28): Dashboard working and report written.

## 2. Task Breakdown (WBS)

We organized the work into four main phases:

### Phase 1: Data Acquisition

- Download USDA Food Access Research Atlas data

- Download CDC PLACES health outcome data
- Download Census Bureau socioeconomic indicators
- Check that all files downloaded correctly

#### Phase 2: Data Preparation

- Handle missing values (we'll use MICE imputation)
- Normalize numerical features using z-scores
- Merge datasets on FIPS codes
- Do population-weighted aggregation for census tracts
- Create composite features (Food Access Barrier Index, Socioeconomic Vulnerability Index)

#### Phase 3: Analysis

- Exploratory data analysis (correlations, distributions, VIF for multicollinearity)
- Regression modeling with OLS, Ridge, and Lasso
- K-means clustering (determine optimal k, validate clusters)
- PCA (reduce dimensions, create biplots)
- Calculate Metabolic Vulnerability Scores

#### Phase 4: Visualization and Reporting

- Build Streamlit dashboard with interactive maps
- Create geographic visualizations
- Write technical report sections
- Make presentation slides
- Final quality check and submission
  This structure keeps everything organized and makes it clear who's responsible for what.

### 3. Task Duration Estimates

#### Why tasks take as long as they do:

Task A (Data Collection) is straightforward but depends on download speeds and making sure we get all the right files. 3 days gives us buffer for any issues.
Task B (Data Cleaning) takes longer because we need to:

- Identify missing data patterns (we know about 12% is missing)
- Implement MICE imputation which is computationally intensive
- Handle outliers carefully (some counties have extreme values)
- Validate that our cleaning doesn't introduce bias
  Task C (Data Integration) is tricky. USDA data is at census tract level, CDC is at county level, and Census has both. We need to aggregate carefully using population weights. We've built extra time for debugging merge conflicts.
  Task D (Feature Engineering) requires domain knowledge. We need to research what makes a good food access barrier index. This involves literature review and testing different formulations.

Task E (EDA) should be relatively quick once we have clean data. It's mostly running analyses and making plots.
Task F (Regression) takes the longest in the analysis phase because:

- We're training three different models
- Each needs hyperparameter tuning
- We need to do k-fold cross-validation
- We have to check assumptions and do diagnostics

Tasks G and H are moderate length. They're well-defined ML techniques we've learned in class.
Task I (MVS Computation) is short once we have all the pieces.
Task J (Dashboard) time depends on Streamlit experience. Kapil has some but will need time for learning and debugging.
Task K (Report) is compressed because we'll be writing sections throughout.

#### Resource constraints:

We each have about 20 hours per week to dedicate to this. That's roughly 5 hours per day if we work 4 days a week. Some tasks can be split among us, others need to be done by one person to maintain consistency (especially the data pipeline).
**Project duration: approximately 4 weeks (22-23 days)**

## 4. Team Responsibilities

**Savitha Vijayarangan -** *Lead & Data Integration Specialist*

- Data collection and cleaning
- FIPS code merging and population-weighted aggregation
- GitHub repository management
- Documentation coordination

**Jane Heng -** *Statistical Analyst & Regression Lead*

- Exploratory Data Analysis
- Regression modeling (OLS, Ridge, Lasso)
- Coefficient analysis and interpretation
- Statistical validation

**Rishi Visweswar Boppana -** *Clustering & PCA Specialist*

- K-Means and hierarchical clustering
- Principal Component Analysis
- Cluster validation and interpretation
- Geographic heat map creation

**Kapil Reddy Sanikommu -** *Predictive Modeling & Visualization Lead*

- Metabolic Vulnerability Score computation
- Interactive Streamlit dashboard development
- Data visualization and infographics
- Presentation materials design

## 5. Gantt Chart Timeline

The Gantt chart below shows the timeline, task assignments, and current project status:

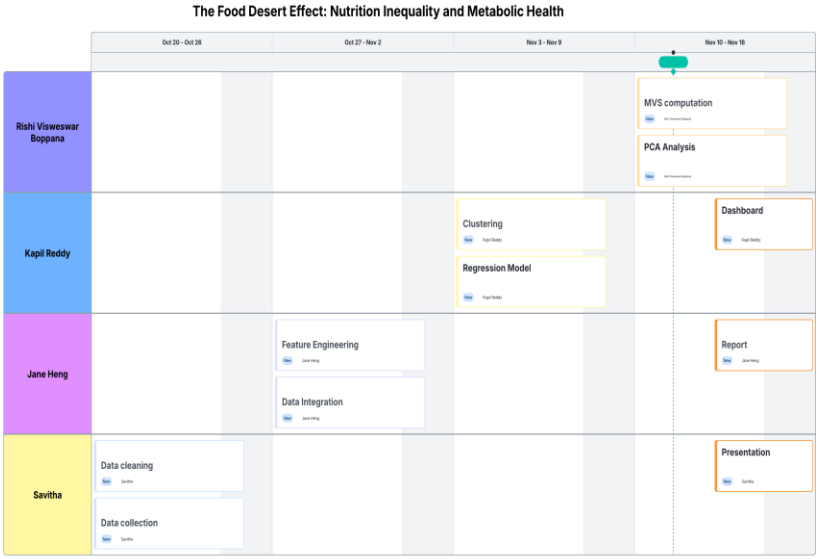


*Figure 2: Gantt Chart showing project timeline and task assignments*

### Current Status (Day 12):

According to our Intermediate Status Report:

- Data collection and cleaning: COMPLETED
- Data integration: COMPLETED
- Feature engineering: IN PROGRESS
- Exploratory analysis: IN PROGRESS
- Remaining tasks on schedule

## 6. Gantt Chart Guidelines

### Keeping the format clear:

Our Gantt chart uses a simple timeline with days numbered 1-28. Each task is shown as a horizontal bar where the length matches the duration. We're using three colors (green, yellow, gray) to show status at a glance.
Team member names are shown on or next to each bar so it's clear who's doing what. Week boundaries are marked so we can see our weekly milestones.

### Showing dependencies:

The chart layout shows dependencies through positioning. You can see that Task C starts right after Task B ends, Task D starts after Task C ends, etc.
Tasks F, G, and H all start around Day 13 and overlap, showing they can run in parallel. This is one of the key scheduling decisions we made.

### Managing resources:

We balanced the workload so everyone has roughly equal amounts of work:

- Savitha: 10 days of data work upfront
- Jane: 6 days of analysis, plus shared work
- Rishi: 6 days of ML work, plus shared work
- Kapil: 7 days of visualization, plus shared work
  Nobody is overloaded, and we built in some slack for the non-critical path tasks. The parallel work during Week 2-3 is crucial. While Jane does regression, Rishi does clustering and PCA. This saves us about 3 days compared to doing everything sequentially.

## Project Management Insights

### How PERT and Gantt work together:

We used PERT first to plan everything out. It helped us identify the critical path and understand which tasks have no flexibility (A, B, C, D, F, I, K). We also used it to account for uncertainty with the three-point estimates.
Now we're using the Gantt chart for day-to-day tracking. It's easier to see at a glance where we are and what's coming next. We update it weekly in our team meetings.
The PERT chart answers, "what's the minimum time to finish?" The Gantt chart answers, "Who's doing what when?"

### Risks we're watching:

Data quality is the biggest risk. We know about 12% of counties have missing data but won't know the full picture until we're deeper into cleaning. Our mitigation is starting Task B early and having Savitha (most experienced) handling it.
Multicollinearity might be an issue with socioeconomic predictors (income, education, employment are probably correlated). That's why we planned to use Ridge regression from the start.
Time pressure is real with a 4-week deadline. But we built some buffer by starting early and having parallel work streams.

#### Success criteria:

We'll consider this successful if:

- Everything finishes within 28 days
- Regression R-squared is above 0.60
- Clustering silhouette score is above 0.45
- Dashboard works without crashing
- Report is clear and well-written
  So far, we're on track (Day 12 of 28, 45% complete).

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

Yes. We used ChatGPT (Generative AI) to refine the grammar and improve the clarity of our report. It also assisted in brainstorming ideas for structuring the PERT

and Gantt sections. All project analysis, charts, and content were created independently by our team.