Comprehensive Project Plan

1. Data Preprocessing

- **Dataset**: Fashion MNIST (60k train, 10k test, 28x28 grayscale, 10 classes).
- **Steps**:
  - Normalize pixel values to `[0, 1]`.
  - Split data into **70% train**, **15% validation**, **15% test**.
  - Flatten images to 1D vectors (784 features) for non-NN models.

2. Dimensionality Reduction with PCA

- Apply PCA to training data.
- Choose components explaining **≥95% variance** (visualize cumulative variance with elbow plot).
- Transform validation/test data using PCA.
- **PCA** (95% and 98% variance), **t-SNE** (visualization), **UMAP** (50 components).

3. Model Selection & Regularization Techniques

Include **regularization methods** for applicable models:

Model Type	Regularization Techniques Tested	Notes
Baseline NN	L2 Regularization, Dropout (20%, 50%)	Add `kernel_regularizer` and `Dropout` layers.
PCA-NN	L2 Regularization, Early Stopping	Monitor validation loss with patience=5.
Logistic Regression	L1 (Lasso), L2 (Ridge), ElasticNet (α=0.1)	Use `penalty` and `solver` parameters.
SVM (RBF)	Adjust `C` (0.1, 1, 10)	Smaller `C` = stronger regularization.
Random Forest	`max_depth`, `min_samples_split`	Prune trees to reduce overfitting.
Gradient Boosting	`learning_rate`, `max_depth`	Lower learning rate + more trees for regularization.

4. Hyperparameter Optimization

- Tune regularization strengths (e.g., L2 λ values: 0.001, 0.01, 0.1).
- Compare dropout rates (20% vs 50%).

5. Training & Evaluation

- **Metrics**:

- Test accuracy, training time, validation loss curves.
- Overfitting analysis (train vs validation accuracy gap).

Sample Comparison Table with Regularization

Model	Regularization Method	Test Accuracy	Overfitting Gap (Train-Val)	Training Time (s)
Baseline NN	None	89.8%	4.2%	180
Baseline NN	L2 (λ=0.001)	90.1%	1.8%	190
Baseline NN	Dropout (50%)	89.5%	1.5%	200
Logistic Regression	L2 (α=0.1)	83.5%	0.9%	10
Logistic Regression	L1 (α=0.1)	82.7%	0.7%	12
SVM (RBF)	C=0.1 (Strong Regularization)	87.2%	0.5%	110
SVM (RBF)	C=10 (Weak Regularization)	88.7%	2.1%	95

Key Findings from Regularization Study

1. **Neural Networks**:
   a. **L2 Regularization** (λ=0.001) improved accuracy by 0.3% and reduced overfitting gap from 4.2% to 1.8%.
   b. **Dropout** slightly reduced accuracy but minimized overfitting.
2. **Logistic Regression**:
   a. **L2** outperformed L1 due to correlated features in PCA-transformed data.
3. **SVM**:
   a. Strong regularization (`C=0.1`) reduced overfitting but sacrificed accuracy.

100-Mark Rubric

Criteria	Sub-Criteria	Marks
Data Preprocessing	Correct normalization, splitting, flattening.	10
Dimensionality Reduction	PCA (95%, 98%), UMAP, t-SNE visualizations.	15

Model Implementation	All 8 models coded correctly (with/without PCA).	20
Regularization Study	Applied ≥2 regularization methods per model.	20
Hyperparameter Tuning	Optimized PCA variance, NN/SVM regularization.	15
Visualization	Explained variance, PCA/UMAP components, loss curves.	10
Analysis & Comparison	Impact of PCA, regularization, model performance.	15
Deliverables	Jupyter notebook + slides (code, results, clarity).	10
Bonus	Advanced techniques (e.g., CNN, hyperparameter search).	+5

Grading Notes

- **Regularization Study (20 marks)**:
  - 5 marks: Tested L1/L2 for Logistic Regression.
  - 5 marks: Compared dropout vs L2 for NN.
  - 5 marks: Analyzed SVM's `C` impact.
  - 5 marks: Discussed trade-offs (accuracy vs overfitting).
- **Penalties**:
  - -5 marks for missing PCA variance optimization.
  - -10 marks for incomplete regularization implementation.

Conclusion should look like:

- **Best Regularization**: L2 (λ=0.001) for NN and L2 for Logistic Regression.
- **Worst Performer**: Strong SVM regularization (`C=0.1`) reduced accuracy by 1.5%.
- **Key Takeaway**: Regularization improves generalization but requires careful tuning.