VIVA Preparation Guide - ML Project 2

Table of Contents

Project Overview

Two Classification Problems:

1. **Forest Cover Type Prediction** - Multi-class classification (7 classes)
2. **Smoking Prediction** - Binary classification (smoker/non-smoker)

Common Models Used:

- **Logistic Regression**
- **Support Vector Machine (SVM)**
- **Multi-Layer Perceptron (MLP) Neural Network**

Forest Cover Type Prediction

Dataset Details

- **Size**: 581,012 instances
- **Features**: 54 total
  - **10 Continuous**: Elevation, Aspect, Slope, Horizontal/Vertical Distance to Hydrology, Distance to Roadways/Fire Points, Hillshade (9am, Noon, 3pm)
  - **44 Binary**: 4 Wilderness Area columns + 40 Soil Type columns
- **Target**: Cover_Type (1-7)
- **Class Distribution**: Imbalanced (Class 2 most common: 283,301; Class 4 least: 2,747)

Key EDA Insights

1. **Class Imbalance**: Significant imbalance across 7 classes
2. **Feature Importance**:
   - Elevation is highly predictive
   - Hillshade features show distinct patterns per cover type
   - Soil types have strong associations with specific cover types

3. **Data Quality**: No missing values, clean dataset
4. **Skewness**: Some continuous features (like distances) are right-skewed

Preprocessing

- **Scaling**: StandardScaler applied only to 10 continuous features
- **Binary Features**: Left unchanged (already 0/1 encoded)
- **Train-Test Split**: 80-20 split with stratification
- **No Feature Engineering**: Used raw features directly

Model Results

Model	Accuracy	Key Observations
MLP Neural Network	**92.21%**	Best performer, (100, 100) hidden layers
Logistic Regression	72.34%	Good baseline, struggles with minority classes
SVM (LinearSVC)	71.14%	Similar to LR, faster training

Model Configurations

Logistic Regression

- **Solver**: LBFGS (good for multi-class)
- **Max Iterations**: 1000
- **Multi-class**: Automatic (multinomial)

SVM (LinearSVC)

- **C**: 1.0
- **Max Iterations**: 2000
- **Dual**: True

MLP Neural Network

- **Architecture**: (100, 100) hidden layers
- **Activation**: ReLU
- **Solver**: Adam
- **Learning Rate**: Adaptive (0.001 initial)
- **Regularization**: Alpha = 0.0001
- **Early Stopping**: Enabled (validation_fraction=0.1)

Why MLP Performed Best?

1. **Non-linearity**: Can capture complex interactions between features

2. **Deep Architecture**: Two hidden layers allow hierarchical feature learning
3. **Adaptive Learning**: Learning rate adaptation prevents overfitting
4. **Early Stopping**: Prevents overfitting on large dataset

Smoking Prediction

Dataset Details

- **Training Size**: 38,984 instances
- **Test Size**: 16,708 instances
- **Original Features**: 23 features
  - Demographics: age, height, weight, waist
  - Health Metrics: blood pressure, cholesterol, liver enzymes (AST, ALT, GTP)
  - Sensory: eyesight, hearing
  - Other: hemoglobin, urine protein, dental caries
- **Target**: Smoking status (0 = non-smoker, 1 = smoker)
- **Class Distribution**: 63.27% non-smokers, 36.73% smokers (slight imbalance)

Key EDA Insights

1. **Strong Predictors**:
   - **GTP** (gamma-glutamyl transpeptidase) - liver enzyme highly correlated with smoking
   - **ALT, AST** - liver enzymes elevated in smokers
   - **Triglycerides** - higher in smokers
   - **Waist circumference** - metabolic marker
   - **HDL** - lower in smokers (good cholesterol)
2. **Medical Patterns**: Clear biochemical differences between smokers and non-smokers
3. **Outliers**: Present but medically meaningful (not removed)
4. **Correlations**: Strong collinearity in liver enzymes (AST/ALT/GTP cluster)

Feature Engineering Strategy

Logistic Regression (31 features)

- **Base Features**: BMI, Waist-Height ratio, BP ratio
- **Lipid Ratios**: Chol/HDL, LDL/HDL, Trig/HDL
- **Liver Ratios**: AST/ALT
- **Sensory**: Average eyesight, hearing sum
- **Polynomial Features**: Degree 2, interaction only

SVM (45 features)

- All LR features +
- **Extended Body**: Waist/Weight ratio
- **Extended BP**: Pulse pressure, MAP (Mean Arterial Pressure)
- **Extended Lipids**: Non-HDL cholesterol, atherogenic index
- **Extended Liver**: Liver enzyme sum, GTP/ALT ratio
- **Age Interactions**: age × hemoglobin, age × BMI, age × systolic
- **Composite Scores**: Metabolic risk, cardiovascular risk

MLP (56 features)

- All SVM features +
- **Body Extended**: BSA (Body Surface Area)
- **Log Transforms**: log(GTP), log(ALT) for skewed features
- **Extended Health Scores**: Hypertension score, health score
- **Age Extended**: age^2, age × cholesterol
- **Dental**: Binary flag for dental issues

Preprocessing

- **Scaling**: StandardScaler applied to all features
- **Train-Test Split**: 80-20 with stratification
- **Feature Engineering**: Different feature sets per model

Model Results

Model	Accuracy	Precision	Recall	F1-Score	CV Score
MLP Neural Network	**75.30%**	75.10%	75.30%	75.18%	75.53%
Logistic Regression	73.52%	73.48%	73.52%	73.49%	73.89%
SVM	60.54%	60.54%	60.54%	60.54%	61.93%

Model Configurations

Logistic Regression

- **Best Parameters**:
  - C = 0.039 (ElasticNet regularization)
  - L1 ratio = 0.525
  - Polynomial features (degree 2, interaction only)
  - Solver: SAGA (supports L1, L2, ElasticNet)

SVM

- **Best Parameters**:
  - C = 0.315
  - Kernel: Sigmoid
  - Gamma = 0.085
- **Note**: Performed poorly, possibly due to kernel choice or hyperparameter space

MLP Neural Network

- **Best Architecture**: (256, 128) hidden layers
- **Activation**: Logistic (sigmoid)
- **Solver**: Adam
- **Learning Rate**: 0.0048 (adaptive)
- **Alpha**: Regularization tuned via grid search
- **Early Stopping**: Enabled

Why Different Feature Sets?

1. **Logistic Regression**: Simpler model, fewer features prevent overfitting
2. **SVM**: Can handle more features, benefits from extended interactions
3. **MLP**: Most complex, can learn from many features, benefits from comprehensive feature engineering

---

Key Concepts & Theory

1. Logistic Regression

How It Works:

- Uses **logistic function** (sigmoid) to model probability
- For multi-class: Uses **softmax** or **one-vs-rest**
- **Decision boundary**: Linear (in transformed space)

Key Parameters:

- **C**: Inverse of regularization strength (smaller C = stronger regularization)
- **Penalty**: L1 (Lasso), L2 (Ridge), or ElasticNet
- **Solver**: LBFGS (multi-class), SAGA (supports all penalties)

Advantages:

- Interpretable (coefficients show feature importance)
- Fast training
- Good baseline
- Probabilistic outputs

Disadvantages:

- Assumes linear decision boundary
- May struggle with complex non-linear patterns

2. Support Vector Machine (SVM)

How It Works:

- Finds **optimal hyperplane** that maximizes margin between classes
- Uses **kernel trick** to handle non-linear data
- **Support vectors**: Data points closest to decision boundary

Key Parameters:

- **C**: Penalty for misclassification (larger C = harder margin)
- **Kernel**: Linear, RBF, Polynomial, Sigmoid
- **Gamma**: Kernel coefficient (controls influence of individual points)

Advantages:

- Effective in high-dimensional spaces
- Memory efficient (uses support vectors only)
- Versatile (different kernels)

Disadvantages:

- Slow on large datasets
- Sensitive to hyperparameters
- Less interpretable
- Doesn't provide probability estimates directly

3. Multi-Layer Perceptron (MLP)

How It Works:

- **Feedforward neural network** with multiple layers
- **Forward pass**: Data flows through layers
- **Backpropagation**: Updates weights using gradient descent
- **Activation functions**: ReLU, tanh, sigmoid introduce non-linearity

Architecture:

- **Input Layer**: Number of features
- **Hidden Layers**: Learn hierarchical representations
- **Output Layer**: Number of classes (with softmax for multi-class)

Key Parameters:

- **Hidden Layer Sizes**: (100, 100) means 2 layers with 100 neurons each
- **Activation**: ReLU (most common), tanh, logistic
- **Solver**: Adam (adaptive), SGD (stochastic gradient descent)
- **Learning Rate**: How fast weights update
- **Alpha**: L2 regularization strength
- **Early Stopping**: Prevents overfitting

Advantages:

- Can learn complex non-linear patterns
- Universal function approximator
- Handles large feature spaces well

Disadvantages:

- Black box (less interpretable)
- Requires more hyperparameter tuning
- Longer training time
- Risk of overfitting

4. Feature Engineering

Why Important?

- **Domain Knowledge**: Medical ratios (Chol/HDL) are clinically meaningful
- **Non-linearity**: Polynomial features capture interactions
- **Normalization**: Log transforms handle skewness
- **Composite Scores**: Combine multiple features into meaningful metrics

Common Techniques Used:

1. **Ratios**: Chol/HDL, AST/ALT (capture relationships)
2. **Polynomial Features**: Interaction terms (age × BMI)
3. **Log Transforms**: log(GTP) for right-skewed data
4. **Composite Scores**: Metabolic risk, CV risk
5. **Averaging**: eyesight_avg = (left + right) / 2

5. Evaluation Metrics

Accuracy:

- Overall correctness: (TP + TN) / Total
- **Limitation**: Misleading with imbalanced classes

Precision:

- Of predicted positives, how many are actually positive: TP / (TP + FP)
- **Use case**: When false positives are costly

Recall:

- Of actual positives, how many were found: TP / (TP + FN)
- **Use case**: When false negatives are costly

F1-Score:

- Harmonic mean of precision and recall: 2 × (Precision × Recall) / (Precision + Recall)
- **Use case**: Balanced metric when both precision and recall matter

Confusion Matrix:

- Shows TP, TN, FP, FN for each class
- **Use case**: Understand where model makes mistakes

6. Preprocessing

StandardScaler:

- **Formula**: (x - mean) / std
- **Why**: Features on different scales can bias models
- **When**: Always for SVM and MLP, optional for Logistic Regression

Stratified Split:

- Maintains class distribution in train/test sets
- **Why**: Important for imbalanced datasets
- **When**: Always use for classification

7. Hyperparameter Tuning

GridSearchCV vs RandomizedSearchCV:

- **GridSearch**: Exhaustive search over all combinations (slow)
- **RandomizedSearch**: Random sampling (faster, often finds good solutions)
- **Used**: RandomizedSearchCV for efficiency

Cross-Validation:

- **K-fold CV**: Split data into k folds, train on k-1, test on 1, repeat k times
- **Why**: Better estimate of model performance, reduces overfitting
- **Used**: 5-fold CV in this project

Expected VIVA Questions

General Questions

Q1: Why did you choose these three models? - **Answer**: - **Logistic Regression**: Interpretable baseline, fast, good for understanding feature importance - **SVM**: Can handle non-linear patterns with kernels, effective in high dimensions - **MLP**: Most flexible, can learn complex patterns, best for non-linear relationships

Q2: What is the difference between classification and regression? - **Answer**: - **Classification**: Predicts discrete categories (e.g., cover type 1-7, smoker/non-smoker) - **Regression**: Predicts continuous values (e.g., price, temperature) - This project uses **classification** (both multi-class and binary)

Q3: What is overfitting and how did you prevent it? - **Answer**: - **Overfitting**: Model learns training data too well, performs poorly on new data - **Prevention**: - Cross-validation for hyperparameter tuning - Early stopping in MLP - Regularization (L1/L2 in Logistic Regression, alpha in MLP) - Train-test split to evaluate generalization

Forest Cover Type Questions

Q4: Why did MLP perform best for Forest Cover Type? - **Answer**: - 7 classes with complex non-linear relationships - 54 features with interactions (elevation × soil type, hillshade patterns) - MLP's deep architecture (100, 100) can capture these complex patterns - Logistic Regression and SVM are more linear, struggle with class boundaries

Q5: How did you handle the class imbalance? - **Answer**: - Used **stratified train-test split** to maintain class distribution - Considered class weights but didn't implement (could mention as future work) - MLP's early stopping helps prevent bias toward majority classes

Q6: Why did you scale only continuous features? - **Answer**: - Binary features (Wilderness_Area, Soil_Type) are already 0/1, scaling unnecessary - Continuous features (Elevation, Slope) have different scales (elevation ~2000-4000, slope ~0-60) - Scaling ensures all features contribute equally to distance calculations (SVM) and gradient updates (MLP)

Q7: What features were most important? - **Answer**: - **Elevation**: Strongest predictor (different cover types at different elevations) - **Hillshade features**: Show sun exposure patterns, vary by cover type - **Soil types**: Specific soil types associated with specific cover types - **Distance to hydrology**: Water availability affects vegetation

Smoking Prediction Questions

Q8: Why did you use different feature sets for each model? - **Answer**: - **Logistic Regression**: Simpler model, fewer features prevent overfitting, polynomial features capture interactions - **SVM**: Can handle more features, extended features help find better decision boundaries - **MLP**: Most complex, can learn from many features, comprehensive feature engineering maximizes its potential

Q9: What medical insights did you discover? - **Answer**: - **Liver enzymes** (GTP, ALT, AST) are strong predictors (smoking affects liver) - **HDL cholesterol** lower in smokers (smoking reduces good cholesterol) - **Triglycerides** higher in smokers (metabolic impact) - **Waist circumference** correlates with smoking (lifestyle factor)

Q10: Why did SVM perform poorly for Smoking Prediction? - **Answer**: - Only 1 iteration in RandomizedSearchCV (limited hyperparameter exploration) - Sigmoid kernel may not be optimal for this data - Feature space (45 features) might need different kernel or better tuning - Could benefit from more extensive hyperparameter search

Q11: Explain your feature engineering approach. - **Answer**: - **Domain Knowledge**: Medical ratios (Chol/HDL, AST/ALT) are clinically meaningful - **Polynomial Features**: Capture interactions (age × BMI) - **Log Transforms**: Handle right-skewed liver enzymes - **Composite Scores**: Combine multiple risk factors (metabolic risk, CV risk) - **Averaging**: Combine bilateral measurements (eyesight, hearing)

Q12: How did you handle the class imbalance in Smoking dataset? - **Answer**: - Used **stratified split** to maintain 63:37 ratio - Considered class weights in hyperparameter tuning (Logistic Regression had 'balanced' option) - Model evaluation uses weighted metrics (precision, recall, F1)

Technical Questions

Q13: What is the difference between L1 and L2 regularization? - **Answer**: - **L1 (Lasso)**: Adds |weights| to loss, can zero out features (feature selection) - **L2 (Ridge)**: Adds weights^2 to loss, shrinks weights but doesn't eliminate - **ElasticNet**: Combines both (L1 + L2) - Used ElasticNet in Logistic Regression for smoking prediction

Q14: Explain backpropagation. - **Answer**: - Algorithm to train neural networks - **Forward pass**: Calculate predictions and loss - **Backward pass**: Calculate gradients using chain rule - **Update**: Adjust weights using gradient descent - Used in MLP training

Q15: What is early stopping? - **Answer**: - Monitors validation performance during training - Stops when validation score doesn't improve for N iterations - Prevents overfitting - Used in MLP (n_iter_no_change=10)

Q16: Why use cross-validation? - **Answer**: - Better estimate of model performance than single train-test split - Reduces variance in performance estimates - Helps select best hyperparameters - Used 5-fold CV for hyperparameter tuning

Q17: What is the difference between train, validation, and test sets? - **Answer**: - **Train**: Used to learn model parameters - **Validation**: Used to tune hyperparameters (early stopping, CV) - **Test**: Used for final evaluation (never used during training) - In this project: Train-test split, validation from training set (early stopping)

Model Comparison Questions

Q18: Which model would you use in production and why? - **Answer**: - **Forest Cover**: MLP (92% accuracy, significant improvement) - **Smoking**: MLP (75% accuracy, best performance) - **Considerations**: - Accuracy is best with MLP - If interpretability needed: Logistic Regression - If speed critical: Logistic Regression or SVM

Q19: How would you improve the models? - **Answer**: - **More hyperparameter tuning**: Especially for SVM - **Ensemble methods**: Combine predictions from multiple models - **Feature selection**: Remove redundant features - **Class balancing**: SMOTE or class weights for imbalanced classes - **More data**: If available - **Deep learning**: Try deeper networks or CNNs if applicable

Q20: What challenges did you face? - **Answer**: - **Class imbalance**: Handled with stratified splits - **Feature engineering**: Required domain knowledge for medical features - **Hyperparameter tuning**: Time-consuming, used RandomizedSearchCV - **SVM performance**: Poor results, needed more tuning - **Large dataset**: Forest Cover (581K instances) required efficient algorithms

---

Quick Reference Summary

Forest Cover Type Prediction

Aspect	Details
Problem Type	Multi-class classification (7 classes)
Dataset Size	581,012 instances, 54 features
Best Model	MLP Neural Network (92.21% accuracy)
Key Features	Elevation, Hillshade, Soil Types
Preprocessing	StandardScaler on 10 continuous features
Class Imbalance	Yes (Class 2: 283K, Class 4: 2.7K)

Smoking Prediction

Aspect	Details
Problem Type	Binary classification
Dataset Size	38,984 train, 16,708 test, 23 original features
Best Model	MLP Neural Network (75.30% accuracy)
Key Features	GTP, ALT, AST, Triglycerides, HDL, Waist
Feature Engineering	Extensive (31-56 features per model)
Class Imbalance	Slight (63% non-smokers, 37% smokers)

Model Performance Comparison

Forest Cover Type

1. **MLP**: 92.21%
2. **Logistic Regression**: 72.34%
3. **SVM**: 71.14%

Smoking Prediction

1. **MLP**: 75.30%
2. **Logistic Regression**: 73.52%
3. **SVM**: 60.54%

Key Takeaways for VIVA

1. **MLP performed best** in both projects due to non-linear pattern learning
2. **Feature engineering** was crucial for smoking prediction (medical domain knowledge)
3. **Preprocessing** matters: scaling for continuous features, stratified splits for imbalance
4. **Hyperparameter tuning** improved performance significantly
5. **Domain knowledge** helped in feature engineering (medical ratios, composite scores)
6. **Class imbalance** handled with stratified splits
7. **Cross-validation** used for reliable performance estimates

---

Final Tips for VIVA

1. **Be Confident**: You've done comprehensive work on both projects
2. **Explain Your Choices**: Why you chose specific models, features, parameters
3. **Acknowledge Limitations**: Mention what could be improved (SVM tuning, ensemble methods)

4. **Connect Theory to Practice**: Explain concepts when discussing your implementation
5. **Show Understanding**: Demonstrate you understand why models performed as they did
6. **Be Honest**: If you don't know something, say so and explain what you would do to find out

Key Points to Emphasize:

- Comprehensive EDA with visualizations
- Multiple models compared systematically
- Proper preprocessing and feature engineering
- Hyperparameter tuning with cross-validation
- Domain knowledge applied (medical features)
- Best practices followed (stratified splits, early stopping)

Good Luck with your VIVA!