Part 1: CompareMLModelsV2.Py

a- Based on accuracy which model is the best one?

The **Linear Discriminant Analysis (LDA)** achieved the highest accuracy at **98.00%**.
So, **LDA is the best-performing model** on this dataset under 2-fold cross-validation.

b- Why the other 11 models did not perform as well?

1. **Linear Regression (96.00%)**
   - Linear regression isn't designed for classification. It predicts continuous values, which we then round to classes. This introduces rounding errors and misclassifications, especially when classes are not linearly separable.
2. **Polynomial Regression (degree=2, 97.33%)**
   - Higher flexibility improves fit, but regression still isn't fundamentally a classifier. It can approximate decision boundaries but is sensitive to overfitting with polynomials, leading to slightly less generalization than LDA.
3. **Polynomial Regression (degree=3, 94.67%)**
   - Even more prone to overfitting. The added complexity doesn't help here because the iris dataset is fairly simple. Instead, the cubic decision boundary introduces noise and reduces generalization accuracy.
4. **Naive Bayes (96.67%)**
   - Assumes feature independence, which is not true for iris features (sepal and petal lengths/widths are correlated). Despite this simplification, it still performs well but not as good as LDA, which explicitly models correlations.
5. **kNN (94.67%)**
   - kNN relies on local neighborhood similarity. With k=5, boundary cases may get misclassified due to averaging across neighbors. It's also

sensitive to scaling and feature correlations.

6. **Quadratic Discriminant Analysis (97.33%)**
    - Allows each class to have its own covariance matrix. On iris, this extra flexibility can lead to small overfitting errors compared to LDA, which assumes equal covariance and matches the iris structure better.
7. **SVM (LinearSVC) (96.00%)**
    - Linear decision boundary works decently but cannot capture subtle nonlinear separations in the versicolor/virginica overlap region. LDA handled that structure better here.
8. **Decision Tree (90.67%)**
    - Decision trees can overfit small datasets. The splits capture noise, and with only 150 samples, the resulting boundaries are less generalizable compared to ensemble methods or LDA.
9. **Random Forest (95.33%)**
    - Reduces overfitting compared to a single tree, but still doesn’t model correlations as efficiently as LDA. It does well, but its randomness leads to slightly lower accuracy.
10. **Extra Trees (95.33%)**
    - Similar to Random Forest but uses more random splits. This increases diversity but reduces fine-tuned accuracy on small datasets like iris.
11. **Neural Network (MLP) (95.33%)**
    - Neural networks usually need more data to shine. With iris (150 samples), the model risks overfitting or underfitting depending on initialization. It doesn’t outperform simpler, statistically grounded models here.

Part 2:

a. Does the program use k-fold cross-validation?

- No. The program uses a single train/test split via `train_test_split(...)`, not k-fold.

b. What percentage of the dataset was used to train the DBN model?

- The code uses `test_size=0.2` in `train_test_split`.
- That means 80% training / 20% testing. Thus, 80% of the data was used to train.

c. How many samples are in the test set?

- The digits dataset has 1797 total samples.
- 20% test = `1797 × 0.2 ≈ 359.4 → 360 samples`.
- Thus, there are 360 test samples.

d. How many samples are in the training set?

- Remaining 80% = `1797 × 0.8 = 1437 samples`.
- Thus, there are 1437 training samples.

e. How many features are in the test set?

- Each digit image is 8×8 = 64 features.
- So the test set's feature matrix shape is `(360, 64)`.
- Thus, there are 64 features per sample and a total if 360*64 = 23040 test feature values.

f. How many features are in the training set?

- Same feature vector dimensionality as test set. Training set's feature matrix shape = `(1437, 64)`.

- Same as the test set, the training set has 64 features per sample and 1437*64 = 91968 training feature values

g. How many classes are there?

- Digits dataset contains numbers 0–9.
- So, there are 10 classes.

h. List the classes.

- Classes are: `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`.