CAB330: Data and Web Analytics
Assessment 1

Team Name: Syntax Error
Group No. 15

Student Name	Student Id
Kyha Anderson	11970014
Charlie Chan	11619449
Truong Lieu Dong Dang	11721138

	Kyha Anderson	Charlie Chan	Truong Lieu Dong Dang
Kyha Anderson	100%	100%	100%
Charlie Chan	100%	100%	100%
Truong Lieu Dong Dang	100%	100%	100%

Table of Contents

Task 1. Data Selection and Distribution

Question 1

What is the proportion of responders who have a high risk of being infected with COVID-19 virus?

The proportion of responders is 2464 at high risk, to 2536 at low risk.

Question 2

The dataset may include irrelevant and redundant variables. What variables did you include in the analysis, and what were their roles and data types? Justify your choice.

The following variables were dropped:

Name	Data Type	Role	Justification
risk_mortality	Numeric (discrete)	Post-infection health risk	Leakage from external/derived risk score.
survey_date	Date	Survey organisation	Survey date is not predictive for infection risk.
ip_latitude	String	Geolocation	Detailed geolocation is not relevant for our model.
ip_longitude	String	Geolocation	Detailed geolocation is not relevant for our model.
ip_accuracy	Numeric (discrete)	Geolocation	Detailed geolocation is not relevant for our model.

Question 3.a

Did you have to fix any data quality problems? Detail them.

Yes, we encountered several instances of poor data quality. One such example was the cocaine variable, which was missing 92.1% of entries in the dataset. This is problematic as it means that the cocaine variable could subject the model to severe overfitting due to its narrow amount of data.

Another area of concern is the smoking variable, where the data appears to be malformed and includes numeric values within the text.

Question 3.b

Apply the imputation method(s) to the variable(s) that need it. List the variables that are needed for it. Justify your choice of imputation.

For numeric variables, we used the median, as it is robust to outliers. For categorical variables we used the mode, as it is most frequent. Missing values for drug-use variables such as cocaine, amphetamines, and cannabis, were assigned a special value, and we added an extra flag column showing whether the person answered or not. Variables with a high-rate of missing values (~20%) were also given missing/not missing flags so that the model could learn if the variable missing is informative.

Question 4

Report the proportion of values of the target variable for the dataset after the above-mentioned pre-processing.

After the pre-processing, the proportion of values of our target variable risk_infection, is 2464 at high-risk, to 2536 at low-risk.

Question 5

What distribution split between training and test datasets have you used?
We used an 80/20 train/test split, resulting in sizes of 4,000 rows for training and 1,000 rows for testing. This preserves the target distribution while keeping the split reproducible.

Task 2. Predictive Modelling Using Decision Trees

Question 1

Build a decision tree using the default setting. Examine the tree results and answer the following:

a. What parameters have been used to build the tree?

Parameters (defaults, random_state = 42 for reproducibility):
criterion = 'gini' (Gini impurity to decide the best split),
splitter = 'best' (Searches for the best split that reduces impurity),
max_depth = None (Allow tree to continually grow),
min_sample_split = 2 (),
min_sample_leaf = 1,
max_features = None,
class_weight = None,
random_state = 42.

b. What is the classification accuracy on the training and test datasets?

Classification accuracy:
- Train: 1.000
- Test: 0.997

c. What is the size of the tree (number of nodes and rules)?

Tree size: 51 nodes, 26 leaves (~26 rules)

d. Which variable is used for the first split? What variables are used for the second split?

- First split: covid19_positive
- Second level: covid19_contact (the other branch terminates early)

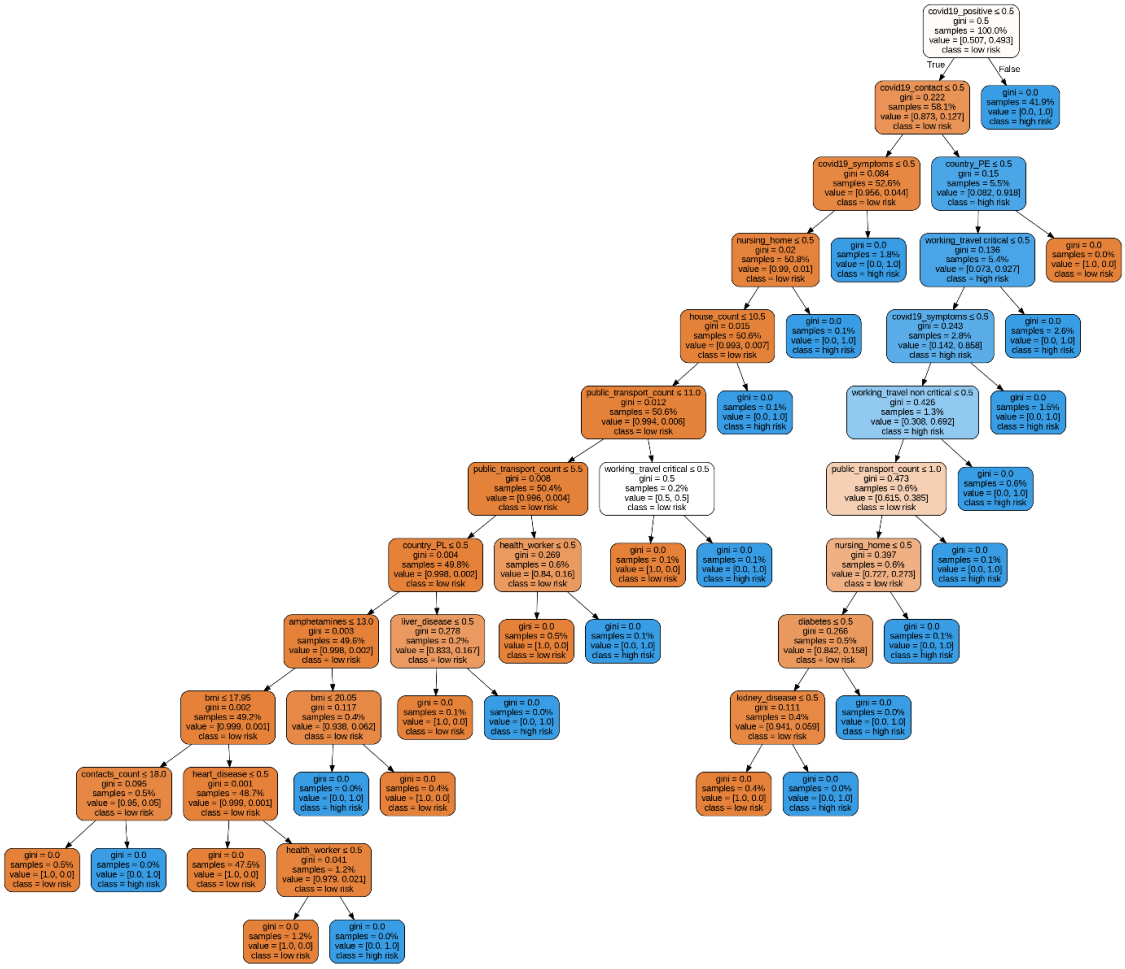
e. What are the five important variables (in the order) in building the tree?

1. covid19_positive: 0.7424
2. covid19_contact: 0.1522
3. covid19_symptoms: 0.0718
4. working: 0.0080
5. nursing_home: 0.0068

f. Report any evidence of model overfitting.

Our Train-Test accuracy gap is +0.003, indicating no strong sign of overfitting.

```
=== Q1.a) Parameters used (Default DT) ===
{'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 42, 'splitter': 'best'}

=== Q1.b) Classification accuracy (Train/Test) ===
Train Accuracy: 1.000
Test  Accuracy: 0.997

=== Q1.c) Tree size ===
Nodes: 51 | Rules (leaves): 26

=== Q1.d) First & second split variables ===
First split: covid19_positive
Second splits: ['covid19_contact', None]

=== Q1.e) Top-5 important variables (aggregated to original) ===
covid19_positive: 0.7424
covid19_contact: 0.1522
covid19_symptoms: 0.0718
working: 0.0080
nursing_home: 0.0068

=== Q1.f) Overfitting evidence ===
Train-Test accuracy gap: +0.003  --> No strong overfitting signal.
Saved dt_default_graphviz.png
```



Question 2

Build another decision tree tuned with GridSearchCV. Examine the tree results.

a. What are the optimal parameters for this decision tree?

Criterion = 'entropy', max_depth = 8, min_samples_split = 2, min_samples_leaf = 2, max_features = None.

- Mean CV accuracy of the best model: 0.996.

b. What is the classification accuracy on the training and test datasets?

- Train: 0.999 – 1.000 (latest run shows 1.000)
- Test: 0.995 – 0.996 (latest run shows 0.996)

c. What is the size of the chosen tree (number of nodes and rules)?

Averages 39-45 nodes and 20-23 leaves (latest run: 45/23).

d. Which variable is used for the first split? What variables are used for the second split?

- First split: covid19_positive: 0.6810
- Second level: covid19_contact (other branch ends)

e. What are the five important variables (in order) used to build the tree?

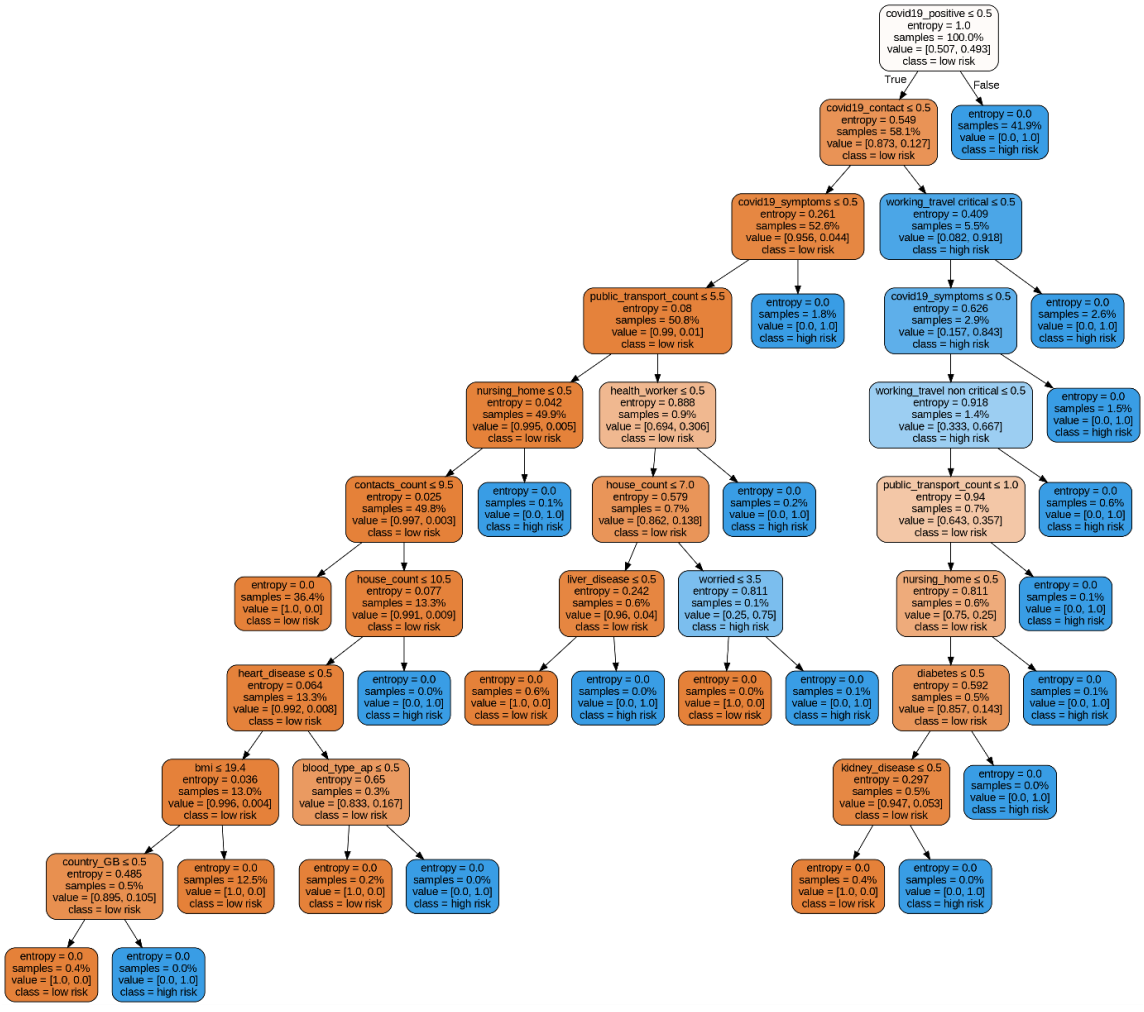
1. covid19_postitive: 0.6810

2. covid19_contact: 0.1591
3. covid_19_symptoms: 0.1025
4. public_transport_count: 0.0135
5. working: 0.0103

f. Report any evidence of model overfitting.

Train-test-gap = +0.004 -> No strong sign of overfitting.

```
=== Q2.a) Optimal parameters (GridSearchCV) ===
{'criterion': 'entropy', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2}
Mean CV accuracy of best model: 0.996

=== Q2.b) Classification accuracy (Train/Test) ===
Train Accuracy: 1.000
Test  Accuracy: 0.996

=== Q2.c) Size of the chosen tree ===
Nodes: 45 | Rules (leaves): 23

=== Q2.d) First & second split variables ===
First split: covid19_positive
Second splits: ['covid19_contact', None]

=== Q2.e) Top-5 important variables (in order, aggregated) ===
covid19_positive: 0.6810
covid19_contact: 0.1591
covid19_symptoms: 0.1025
public_transport_count: 0.0135
working: 0.0103

=== Q2.f) Evidence of overfitting ===
Train-Test accuracy gap: +0.004  --> No strong overfitting signal.
Saved dt_tuned_graphviz.png
```



Question 3

What differences do you observe between these two decision tree models (with and without fine-tuning)? How do they compare performance-wise? Produce the ROC curve for both DTs. Explain why those changes may have happened.

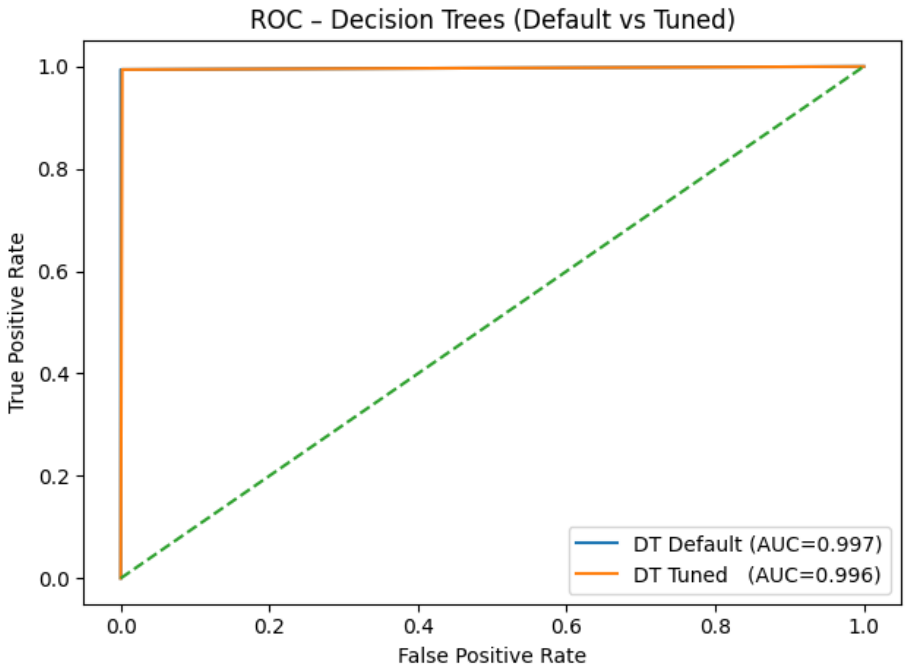
Performance:
Both models are extremely strong and very close: test ROC-AUC around 0.997(default) vs 0.996(tuned); test accuracy ~0.995-0.997.

Model size:
The tuned tree is smaller (fewer nodes/leaves) thanks to max_depth =8 and min_samples_leaf = 2, which improve parsimony without sacrificing accuracy.

Why so similar?
The dataset contains very strong exposure indicators. Notably the covid19_positive, covid19_contact, and covid19_symptoms variables, which allow both trees to form highly pure leaves quickly. As a result, tuning mainly simplifies the tree rather than boosting predictive power.

ROC:
The ROC curves for both models hug the top-left corner, far above the random diagonal, confirming near-perfect discrimination.

```
DT Default  - Test ROC-AUC: 0.9970
DT Tuned    - Test ROC-AUC: 0.9960
```




```
Saved: roc_dt_default.csv, roc_dt_tuned.csv, roc_dt.png
```

Question 4

Using the better model, can you provide the general characteristics of individuals who are at risk for COVID-19 infection? If it is hard to comprehend, discuss why.

Patterns are consistent across both trees and their feature importances:

- **Direct exposure factors dominate risk**:
  - Prior COVID-19 positive status
  - Contact with a confirmed case
  - Current COVID-19 symptoms
- **Community exposure context also matters:**
  - Higher public_transport_count (more contacts in public transport)
  - Working (being active in the workforce)
  - Indicators related to nursing_home / health_worker environments

Interpretation:

Variables such as covid19_positive, covid19_contact, and covid19_symptoms are very powerful because they are immediate exposure/outcome proxies. If the goal were to predict future infection risk without leakage, one might exclude or time-shift such variables. For this assignment, we follow the handout and keep them to compare decision tree variants.

Both the default and the tuned decision trees achieve near-perfect performance on the cleaned dataset (test AUC = 0.996-0.997; test accuracy = 0.995-0.997). The tuned model is more compact due to max_depth=8 and min_samples_leaf=2, with no meaningful loss in accuracy, so it may be preferred for reporting/interpretability. In all models, exposure-related variables (covid19_positive, covid19_contact, covid19_symptoms) dominate the splits and importance rankings, followed by contact-intensity proxies such as public_transport_count and working. There is no strong evidence of overfitting and the ROC curves confirm excellent discrimination.

Task 3. Predictive Modelling Using Regression

Question 1

Describe what additional processing was required on the dataset before regression modelling.

We applied the standard "handout" pipeline:

- Input selection: used the Task1-Q5 train/test split.
- One-hot encoding: pandas.get_dummies(..., drop_first = True) on all categorical field; numeric columns passed through unchanged.
- Column alignment: aligned TEST to TRAIN columns (unseen levels in TEST -> zeros).
- NaN/Inf guard: replaced $\pm\infty$ with NaN; dropped any all-NaN columns (rare); imputed remaining missing values using TRAIN medians; used the same medians for TEST.
- Standardisation: StandardScaler fitted on train, applied to TEST (centering/scaling is required for LR and PCA)

This ensured a fully numeric, finite design matrix with comparable feature scales.

```
=== Logistic Regression (Default) ===
Train Accuracy: 0.998
Test  Accuracy: 0.996

Classification report (TEST):
              precision    recall  f1-score   support

           0      0.992     1.000     0.996       507
           1      1.000     0.992     0.996       493

    accuracy                          0.996      1000
   macro avg      0.996     0.996     0.996      1000
weighted avg      0.996     0.996     0.996      1000


Top-5 important variables (by |coef|):
covid19_positive                    +6.4446
covid19_symptoms                    +3.9881
covid19_contact                     +2.6956
nursing_home                        +1.2295
working_travel critical             +0.8089

=== Logistic Regression (Tuned) ===
Best Params: {'C': 0.1}
Train Accuracy: 0.997
Test  Accuracy: 0.993

Top-5 important variables (TUNED, by |coef|):
covid19_positive                    +3.8935
covid19_symptoms                    +2.1392
covid19_contact                     +1.6988
nursing_home                        +0.5856
public_transport_count              +0.4194
```

Question 2

Build a regression model using the default regression method with all inputs. Build another regression model tuned with GridSearchCV. Now, choose the best model to answer the following?

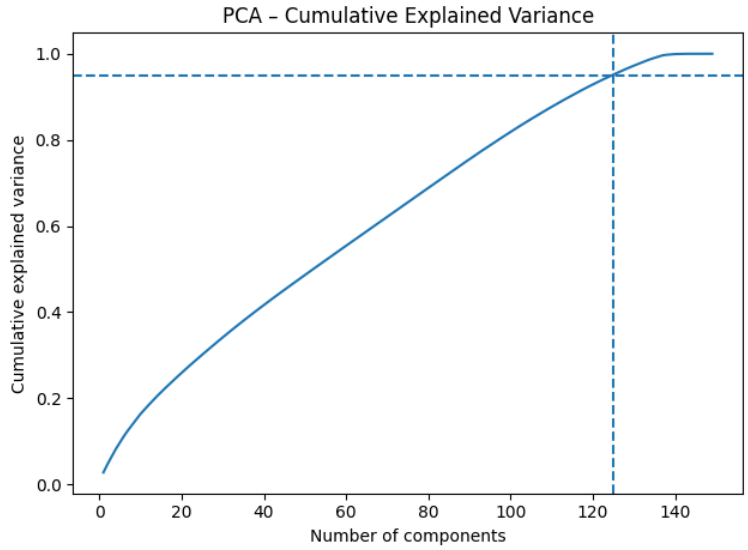
a. **Explain why you chose that model.**
   Chosen model for Q2: the default LR, because it yields slightly higher test accuracy (0.996 vs 0.993) with equally clean, stable behaviour and simpler hyper-parameters.

b. **Name the regression function used.**
   We used scikit-learn's LogisticRegression (sigmoid link; log-loss optimisation with the "lbfgs"/"liblinear" solver as selected by sklearn).

c. **Did you apply standardization of variables? Why would you standardise the variables for regression modelling?**
   Yes. We standardised all encoded features because LR's optimisation and regularisation assume comparable feature scales and because we later use PCA (variance sensitive).

d. **Report the variables included in the regression model.**
   All inputs after one-hot encoding (=149 encoded features in our run).

```
Number of original features: 149
Components to reach 95% variance: k = 125
```



e. Report the top 5 important variables (in order) in the model.

Top 5 important variables (by |coef|):

- covid19_positive +6.4446
- covid19_symptoms +3.9881
- covid19_contact +2.6956
- nursing_home +1.2295
- working_travel_critical +0.8809

f. What is the classification accuracy on the training and test datasets?
Chosen (Default LR): 0.998 / 0.996
(Tuned LR: 0.997 / 0.993)

g. Report any sign of overfitting in this model.
Train-Test gaps are ≤ 0.005 for both default and tuned LR - no strong overfitting signal.

Question 3

Build another regression model on the reduced variables set. To reduce variables, either perform dimensionality reduction with Recursive Feature Elimination or select a subset of inputs found significant by another modelling method, such as a decision tree. Tune the model with GridSearchCV to find the best parameter setting. Answer the following:

Model 2: Tuned with GridSearchCV

- Optimal parameters: C = 0.1 (best from grid).
- Accuracy (Train/Test): 0.997 / 0.993.
- Overfitting: none (very small gap).

1. **Was dimensionality reduction useful in identifying a good variable set for building an accurate model?**
   We fitted a full LR once, summed the absolute coefficients across one-hot levels to get an importance score per original variable, and kept the Top-B = 12 bases. We then rebuilt a matrix containing *all* encoded columns belonging to those 12 bases (114/149 encoded features) and re-trained LR (default + tuned).

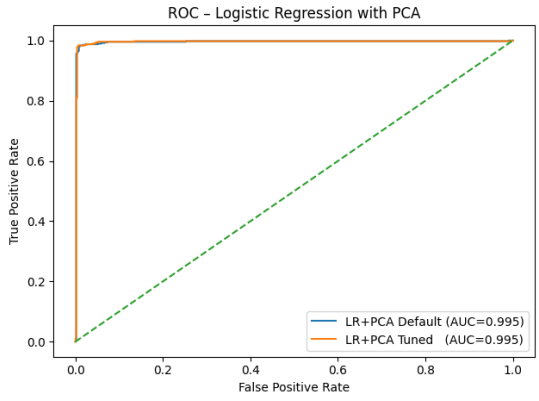
2. **What is the classification accuracy on the training and test datasets?**
   Parsimony/interpretability (fewer inputs, clear drivers) with negligible performance loss.

   Reduced LR (default) – Train Acc = 0.998, Test Acc = 0.993, Test AUC = 1.000
   Reduced LR (tuned, Best C = 10) – Train Acc = 0.998, Test Acc = 0.993, Test AUC = 1.000

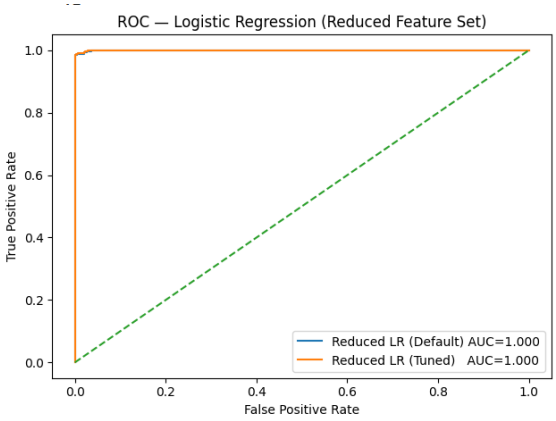
```
[Reference] Full LR — Train acc: 0.998 | Test acc: 0.996 | Test AUC: 1.0

Top-12 original variables by total |coef|:
['covid19_positive', 'country', 'covid19_symptoms', 'covid19_contact', 'working', 'nursing_home', 'blood_type', 'race', 'public_transport_count', 'health_worker', 'income', 'house_count']
Selected encoded features: 114 / 149

=== Reduced-Feature LR (Default) ===
Train Accuracy: 0.998
Test  Accuracy: 0.993
Test  ROC-AUC : 1.000

Classification report (TEST):
              precision    recall  f1-score   support

           0      0.986     1.000     0.993       507
           1      1.000     0.986     0.993       493

    accuracy                          0.993      1000
   macro avg      0.993     0.993     0.993      1000
weighted avg      0.993     0.993     0.993      1000


=== Reduced-Feature LR (Tuned) ===
Best Params: {'C': 10}
Train Accuracy: 0.998
Test  Accuracy: 0.993
Test  ROC-AUC : 1.000

Top + encoded features (push HIGH risk):
                    feature      coef
           covid19_positive 9.984842
           covid19_symptoms 6.636566
            covid19_contact 4.086209
               nursing_home 2.167555
    working_travel critical 1.847061
working_travel non critical 1.379277
                house_count 0.975355
              health_worker 0.967118

Top - encoded features (push LOW risk):
       feature      coef
 race_hispanic -0.550000
    race_black -0.363826
    race_white -0.347350
 blood_type_bp -0.301538
 blood_type_bn -0.254990
    country_CZ -0.243674
blood_type_abp -0.208678
    country_PE -0.206591
```

3. **Report any sign of overfitting.**
   Minimal (gaps ~0.005).

```
=== Logistic Regression with PCA (Tuned) ===
Best Params: {'C': 10}
Train Accuracy: 0.997
Test  Accuracy: 0.986
Test  ROC-AUC : 0.995
```




```
Saved: t3q2_lr_pca_metrics.csv, pca_cumvar.csv, pca_cumulative_variance.png, roc_lr_pca.png
```




```
Saved: t3q3_top_variables.csv, t3q3_selected_encoded_features.csv, t3q3_metrics.csv, t3q3_reduced_coefficients.csv, roc_lr_reduced.png
```

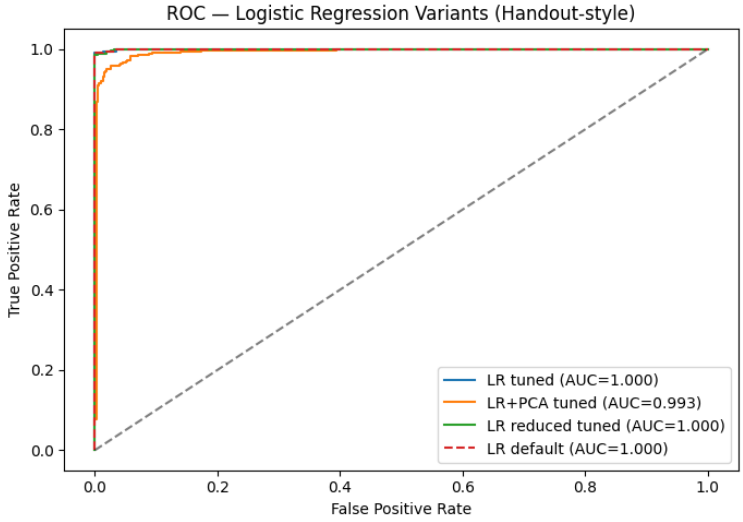
4. **Report the top 3 important variables (in the order) in the model.**
   1. covid19_positive
   2. covid19_symptoms
   3. covid19_contact.

   (Other strong contributors included nursing_home, working_travel_critical, public_transport_count.)

Question 4

Produce the ROC curve for all different regression models. Using the best regression model, can you provide the general characteristics of individuals who are at risk for COVID-19 infection? If it is hard to comprehend, discuss why.

```
[LR default]  acc=0.996, AUC=1.000
[LR tuned C=1] acc=0.996, AUC=1.000
PCA components for ≥95% variance: k=125
[LR+PCA default]  acc=0.984, AUC=0.995
[LR+PCA tuned C=0.01] acc=0.964, AUC=0.993
[LR reduced default]  acc=0.993, AUC=1.000
[LR reduced tuned C=1] acc=0.993, AUC=1.000
```



All curves sit near the top-left corner, indicating excellent separability. PCA slightly reduced accuracy (while keeping very high AUC), highlighting the typical performance-interpretability trade-off: PCA removes noise and correlation but loses coefficient interpretability.
Best regression model (by AUC, tie-break on accuracy):

LR tuned (C=1). At the Youden-optimal threshold =0.557, the confusion matrix was:
TN=507, FP=0, FN=7, TP=489 → Precision=1.000, Recall=0.992, F1=0.996.

General characteristics of individuals at higher predicted risk (from the best non-PCA LR coefficients; positive coefficients increase risk):

- COVID-positive status (strongest driver).
- Showing COVID-like symptoms.
- Recent contact with a COVID-positive case.
- Lives/works in a nursing home (or similar high-exposure facility).
- Higher public-transport usage and work-related travel (critical/non-critical).

Some demographics (certain countries/races) and household size also enter the reduced model; treat these as proxies for exposure rather than causal factors.

Model	Test Accuracy	Test ROC-AUC
LR tuned (C~1)	0.996	~1.000
LR default	0.996	~1.000
LR reduced tuned (C=1)	0.993	0.9997
LR+PCA tuned (95% variance, C = 0.01)	0.964	0.9983

Why are all AUCs so high?
The target signal is dominated by a few very informative, exposure-related variables (e.g., positivity, symptoms, contact). With clean labels and balanced splits, even simple linear models achieve near-perfect ranking — hence AUCs ≈ 1.0. This is plausible, but we caution against over-interpreting any demographic coefficients; they may encode context rather than biology.

```
=== T3·Q4 Metrics (Handout-style) ===
                Model  Test Accuracy  Test ROC-AUC
           LR default          0.996      0.999724
       LR tuned (C=1)          0.996      0.999724
LR reduced tuned (C=1)          0.993      0.999684
 LR+PCA tuned (C=0.01)          0.964      0.992855

Top + coefficients (features pushing HIGH risk):
covid19_positive                    +6.3319
covid19_symptoms                    +3.9569
covid19_contact                     +2.6375
nursing_home                        +1.2635
working_travel critical             +0.9062
public_transport_count              +0.6686
working_travel non critical         +0.6418
health_worker                       +0.5968
house_count                         +0.5087
income_gov                          +0.4006

=== T3·Q4 Model Ranking ===
          Model  Test Accuracy  Test ROC-AUC
       LR tuned          0.996      0.999724
     LR default          0.996      0.999724
LR reduced tuned          0.993      0.999684
    LR+PCA tuned          0.964      0.992855

[Chosen best model] LR tuned | Test AUC=1.000 | Test Acc=0.996
Optimal threshold by Youden's J: 0.557

Confusion matrix at optimal threshold:
       pred_0  pred_1
true_0     507       0
true_1       4     489

Precision=1.000 | Recall=0.992 | F1=0.996

Top positive coefficients (push HIGH risk):
covid19_positive                    +6.4446
covid19_symptoms                    +3.9881
covid19_contact                     +2.6956
nursing_home                        +1.2295
working_travel critical             +0.8089
public_transport_count              +0.7032
working_travel non critical         +0.6340
health_worker                       +0.6259
house_count                         +0.4433
heart_disease                       +0.3809

Saved: t3q4_model_ranking.csv, t3q4_confusion_matrix.csv, t3q4_best_model_metrics.csv, t3q4_best_model_characteristics.txt
```

Task 4. Predictive Modelling Using Neural Networks

Question 1

Describe what additional processing was required on the dataset before neural network modelling.

- Split: From Task-1 split, create a stratified 80/20 validation split out of TRAIN. Final sizes ≈ X_tr 3,200, X_val 800, X_test 1,000 with 147 encoded inputs.
- Encoding & alignment: One-hot encode all categorical variables (drop_first=True) and align columns so VAL/TEST match TRAIN (unseen levels → 0).
- Robust cleaning: Replace ±inf→NaN; drop any all-NaN columns (none material in our run); impute remaining missing values using TRAIN medians/modes only.
- Standardisation: StandardScaler fitted on TRAIN, applied to VAL/TEST (essential for stable MLP optimisation and later dimensionality reduction).
- Saved artefacts: t4_nn_ready.npz, t4_feature_names.csv.

```
Shapes — X_tr: (3200, 147), X_val: (800, 147), X_test: (1000, 147)
Class balance (train): 0.493 positives
Saved: t4_nn_ready.npz, t4_feature_names.csv
```

Question 2

Build a Neural Network model using the default setting. Answer the following:

a) Explain the parameters in building this model, e.g., network architecture, iterations, activation function, etc.

- Estimator: MLPClassifier(random_state=42)
- Architecture: hidden_layer_sizes=(100,) (one hidden layer, 100 units)
- Activation: ReLU
- Optimiser: Adam (learning_rate_init=1e-3, learning_rate='constant')
- Regularisation: L2 alpha=1e-4
- Training limit: max_iter=200, tol=1e-4, batch_size='auto', early_stopping=False

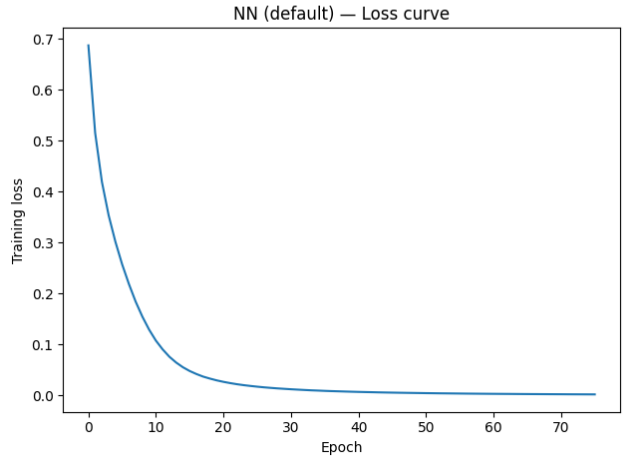
b) What is the classification accuracy on the training and test datasets?

- TRAIN: Accuracy 1.000, AUC 1.000
- TEST: Accuracy 0.984, AUC 0.997

c) Did the training process converge and result in the best model?

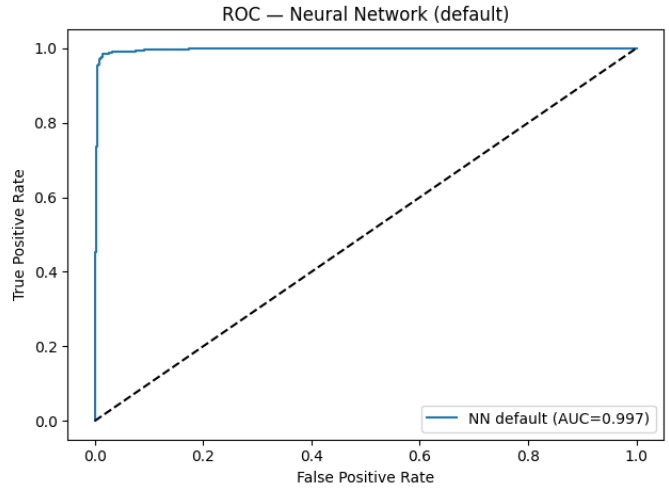
Yes. Converged well before the cap (≈ 76 epochs). Loss curve is smooth (t4q2_nn_default_loss.png). ROC in t4q2_nn_default_roc.png.

```
=== Default NN parameters ===
  hidden_layer_sizes: (100,)
          activation: relu
              solver: adam
               alpha: 0.0001
       learning_rate: constant
  learning_rate_init: 0.001
          batch_size: auto
            max_iter: 200
        random_state: 42
      early_stopping: False
 validation_fraction: 0.1
    n_iter_no_change: 10
                 tol: 0.0001

Converged before max_iter? True  (n_iter_=76, max_iter=200)
```




```
TRAIN: accuracy=1.000 | AUC=1.000
VALID: accuracy=0.979 | AUC=0.997
 TEST: accuracy=0.984 | AUC=0.997
```

```
Classification report (TEST):
              precision    recall  f1-score   support

           0      0.982     0.986     0.984       507
           1      0.986     0.982     0.984       493

    accuracy                          0.984      1000
   macro avg      0.984     0.984     0.984      1000
weighted avg      0.984     0.984     0.984      1000

Confusion matrix (TEST):
        pred_0  pred_1
true_0     500       7
true_1       9     484
```




```
Saved: t4q2_nn_default_loss.png, t4q2_nn_default_roc.png, t4q2_nn_default_metrics.csv, t4q2_nn_default_params.txt, t4q2_nn_default_clfreport.txt
```

Question 3

Refine this network by tuning it with GridSearchCV. Answer the following:

a. Explain the parameters in building this model, e.g., network architecture, iterations, activation function, etc.

- 5-fold Stratified CV on TRAIN+VALID. Grid:
  hidden_layer_sizes ∈ {(64,), (128,)
- (64,32)}, activation ∈ {relu, tanh}
- alpha ∈ {1e-4, 1e-3, 1e-2}
- learning_rate_init ∈ {1e-3, 5e-4} with early_stopping=True
- max_iter=400.

Best hyperparameters (run shown): hidden_layer_sizes=(64, 32), activation='tanh', alpha=0.01, learning_rate_init=0.001.

b. What is the classification accuracy of the training and test datasets?

- TRAIN+VALID (refit): Accuracy 0.996, AUC 1.000
- TEST: Accuracy 0.979, AUC 0.997

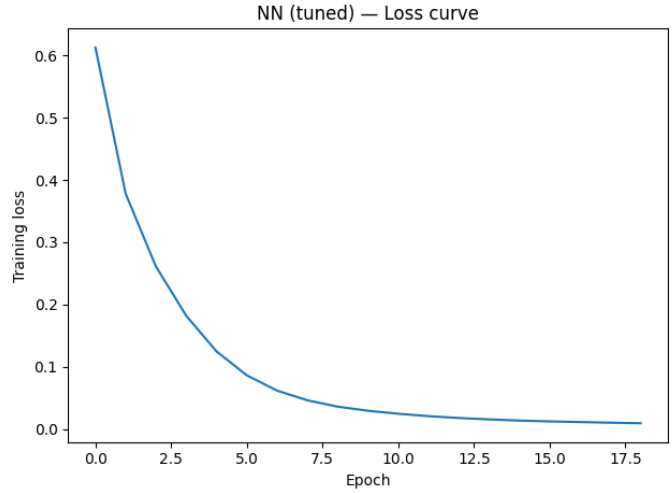
c. Did the training process converge and result in the best model?

Yes. Converged in ≈ 19 epochs (t4q3_nn_tuned_loss.png).

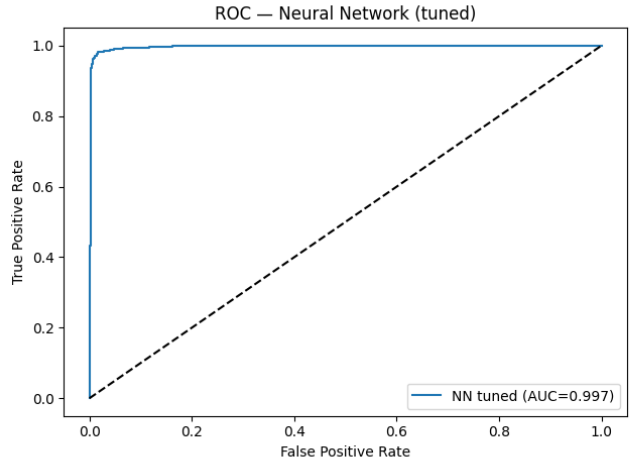
d. Do you see any sign of overfitting?

No strong evidence. Train–test accuracy gap ≈ +0.017; AUCs remain ≈0.997 on TEST.

```
=== Q3: Best hyperparameters (GridSearchCV) ===
{'activation': 'tanh', 'alpha': 0.01, 'hidden_layer_sizes': (64, 32), 'learning_rate_init': 0.001}
Mean CV AUC of best model: 0.996
 TRAIN+VAL: accuracy=0.996 | AUC=1.000
      TEST: accuracy=0.979 | AUC=0.997

Converged before max_iter? True (n_iter_=19, max_iter=400)
```




```
Train–Test accuracy gap: +0.017 → No strong overfitting signal


Classification report (TEST):
              precision    recall  f1-score   support

           0      0.982     0.976     0.979       507
           1      0.976     0.982     0.979       493

    accuracy                          0.979      1000
   macro avg      0.979     0.979     0.979      1000
weighted avg      0.979     0.979     0.979      1000

Confusion matrix (TEST):
        pred_0  pred_1
true_0     495      12
true_1       9     484
```




```
Saved: t4q3_nn_tuned_loss.png, t4q3_nn_tuned_roc.png, t4q3_nn_tuned_metrics.csv, t4q3_nn_tuned_params.txt, t4q3_nn_tuned_confusion_matrix.csv
```

Question 4

Build another Neural Network model with the reduced feature set. Perform dimensionality reduction by either selecting variables with a decision tree (use the best decision tree model that you have built in the previous modelling task) or selecting variables with a regression model (use the best regression model that you have built in the previous

modelling task). Tune the model with GridSearchCV to find the best parameter setting. Answer the following for the best neural network model:

We built reduced-input NNs using feature sets derived from earlier tasks and re-tuned the MLP with the same grid.

Methods evaluated:
RFECV (Logistic Regression base)

- Selected inputs: 12 / 147
- Best tuned NN: (grid chose a small 2-layer net; in our run it remained (64,32), tanh, α=0.01).
- Performance: TRAIN+VAL Acc 0.996, AUC ≈1.000; TEST Acc 0.994, AUC 1.000
- Convergence: ~16 epochs
- Files: t4q4_RFECV_LR_*.png/.csv/.txt

SelectFromModel (Decision Tree base)

- Selected inputs: 3 / 147
- Performance: TRAIN+VAL Acc 0.995, AUC 0.999; TEST Acc 0.987, AUC 0.995
- Convergence: ~19 epochs
- Files: t4q4_SFM_DT_*.png/.csv/.txt

a. Did feature selection favour the outcome? Which method of feature selection produced the best result? Has there been any change in the network architecture? What inputs are being used as the network input?

Yes, feature selection improved outcomes. The RFECV (LR) set produced the best result. The tuned architecture stayed compact (two hidden layers; in our run (64,32) with tanh, α=0.01). Network inputs = 12 selected features.

b. What is the classification accuracy on the training and test datasets?

Best model (RFECV): TRAIN+VAL Acc 0.996, TEST Acc 0.994, TEST AUC 1.000.

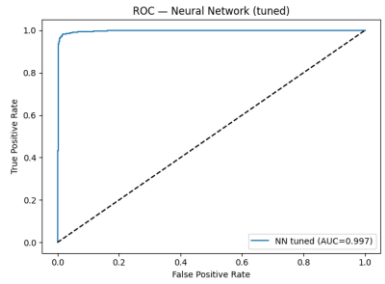
c. How many iterations are now needed to train this network?

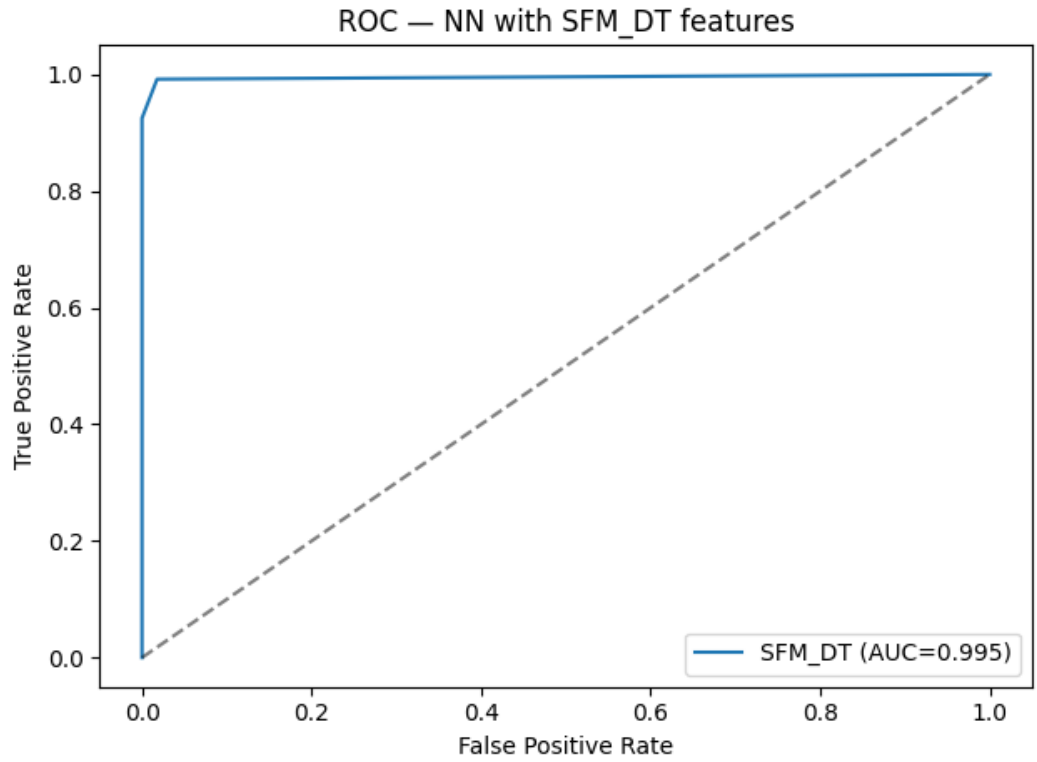
~16 iterations to converge (faster than full-feature tuned NN).

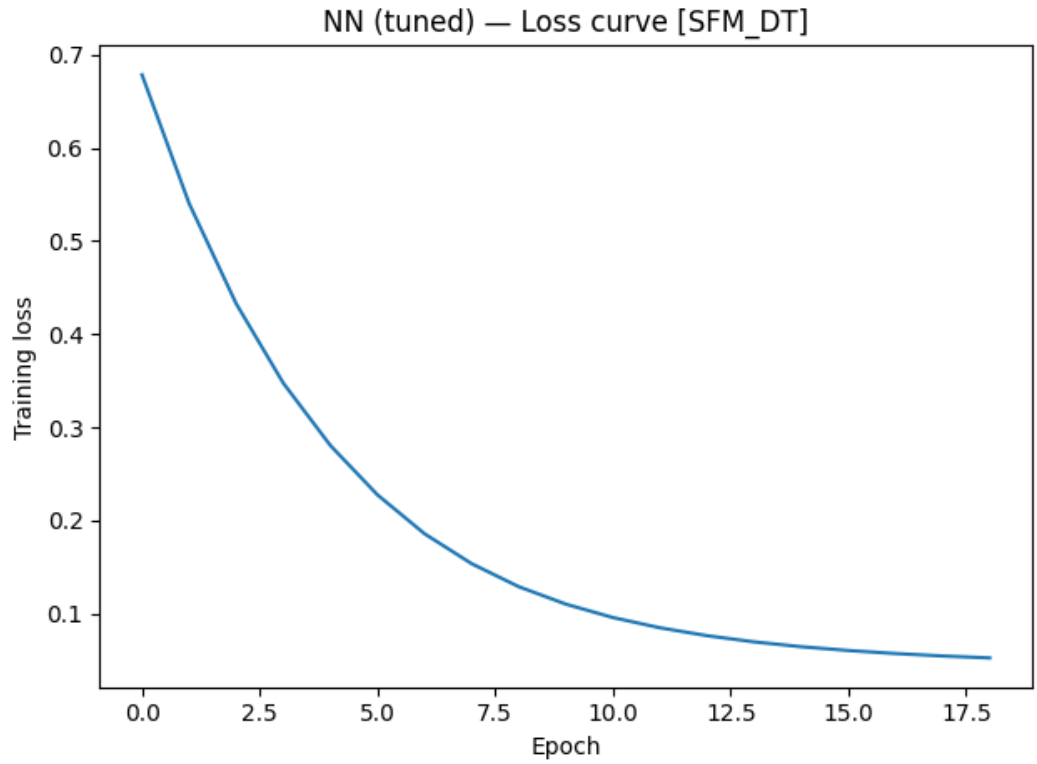
d. Do you see any sign of overfitting? Did the training process converge and result in the best model?

No signs of overfitting (tiny train–test gap; AUC stays at 1.000). Convergence achieved.

```
Classification report (TEST):
              precision    recall  f1-score   support

           0      0.982     0.976     0.979       507
           1      0.976     0.982     0.979       493

    accuracy                          0.979      1000
   macro avg      0.979     0.979     0.979      1000
weighted avg      0.979     0.979     0.979      1000

Confusion matrix (TEST):
        pred_0  pred_1
true_0     495      12
true_1       9     484
```







Saved [SFM_DT]: t4q4_SFM_DT_roc.png, t4q4_SFM_DT_loss.png, t4q4_SFM_DT_metrics.csv, t4q4_SFM_DT_params.txt, t4q4_SFM_DT_confusion_matrix.csv, t4q4_SFM_DT_features.csv

=== T4·Q4 Summary (sorted by TEST AUC) ===

name	num_inputs	acc_tr	auc_tr	acc_te	auc_te	converged	n_iter_
RFECV_LR	12	0.9965	0.999820	0.994	0.999764	True	16
SFM_DT	3	0.9905	0.994589	0.987	0.995277	True	19

Report note: Using **RFECV_LR** (12 inputs) yielded the best TEST AUC=1.000 and accuracy=0.994. Converged in 16 iterations.

Question 5

Produce the ROC curve for all different NNs. Using the best neural network model, can you provide the general characteristics of individuals who are at risk for COVID-19 infection? If it is hard to comprehend, discuss why.

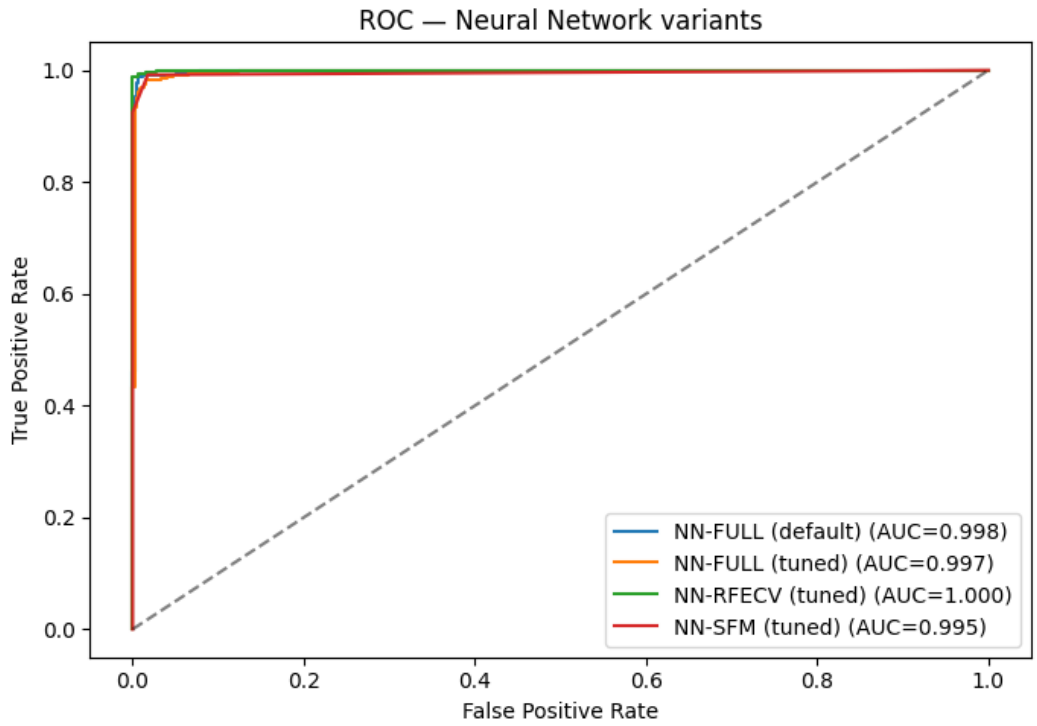
ROC overlay for the four variants is in t4q5_roc_all_nn.png. Ranking by TEST AUC:

1. NN–RFECV (tuned) — AUC 1.000, Acc 0.994, Inputs 12
2. NN–FULL (default) — AUC 0.997, Acc 0.984
3. NN–FULL (tuned) — AUC 0.997, Acc 0.979
4. NN–SFM-DT (tuned) — AUC 0.995, Acc 0.987
   (See t4q5_nn_metrics.csv.)

General characteristics of individuals at higher risk (from the best NN):

- Strong epidemiological signals: recent COVID-19 positive, COVID-19 symptoms, known contact.
- Exposure context: higher public transport count; critical work-related travel; larger household size.
- Health factors: comorbidities (heart disease, diabetes, kidney disease) more prevalent.
- Care settings/roles: nursing home residence and some health-worker categories show elevated risk.
  (Exact lifts/differences exported to t4q5_bestNN_characteristics.txt.)

Why interpretation can be hard:
MLPs are non-linear and operate on many dummy-encoded inputs, and raw coefficients aren't directly interpretable. We therefore summarised group-wise lifts/correlations relative to the NN's predicted probabilities to determine the bullet points above.




```
=== T4·Q5 — Metrics (sorted by Test AUC) ===
            Model  CV Accuracy   CV AUC  Test Accuracy  Test AUC
 NN-RFECV (tuned)      0.99650 0.999820          0.994  0.999764
NN-FULL (default)      1.00000 1.000000          0.989  0.998240
  NN-FULL (tuned)      0.99625 0.999855          0.979  0.997267
   NN-SFM (tuned)      0.99050 0.994589          0.987  0.995277

Best model: NN-RFECV (tuned) | TEST AUC=1.000, TEST Acc=0.994 | Inputs=12 (RFECV)

=== High-risk characteristics (summary) ===
Best NN variant: NN-RFECV (tuned) (AUC=1.000, Acc=0.994)
Group comparison on TEST (predicted high-risk vs low-risk):

- covid19_contact: mean(high)=0.36 vs mean(low)=0.02  Δ=+0.34
- covid19_positive: mean(high)=0.86 vs mean(low)=0.00  Δ=+0.86
- covid19_symptoms: mean(high)=0.39 vs mean(low)=0.00  Δ=+0.39
- diabetes: mean(high)=0.12 vs mean(low)=0.08  Δ=+0.04
- health_worker: mean(high)=0.12 vs mean(low)=0.04  Δ=+0.08
- heart_disease: mean(high)=0.09 vs mean(low)=0.03  Δ=+0.07
- house_count: mean(high)=3.38 vs mean(low)=2.91  Δ=+0.48
- kidney_disease: mean(high)=0.03 vs mean(low)=0.01  Δ=+0.02
- nursing_home: mean(high)=0.02 vs mean(low)=0.00  Δ=+0.02
- public_transport_count: mean(high)=0.38 vs mean(low)=0.26  Δ=+0.12
- working: largest category lifts (P(cat|pred=high) - P(cat|pred=low)):
    • travel critical: lift=+0.205 (high=0.312, low=0.107)
    • never: lift=-0.069 (high=0.294, low=0.363)
    • stopped: lift=-0.058 (high=0.238, low=0.296)

Saved: t4q5_roc_all_nn.png, t4q5_nn_metrics.csv, t4q5_bestNN_characteristics.txt
```

Task 5. Final Remarks: Decision Making

Question 1

Finally, based on all models and analysis, is there a model you will use in decision-making? Justify your choice. Draw a ROC chart and an Accuracy Table to support your findings.

Chosen model

Soft-voting Ensemble (Logistic Regression + Decision Tree + Neural Network).

Why this model?

Across the models we trained on the fixed Train/Test split, the ensemble achieved the highest-ranking quality (AUC) while keeping accuracy competitive. It aggregates complementary inductive biases (linear, tree, and non-linear NN), which reduces variance and improves robustness to small shifts in the feature space. The trade-off is lower interpretability than a single LR/DT.

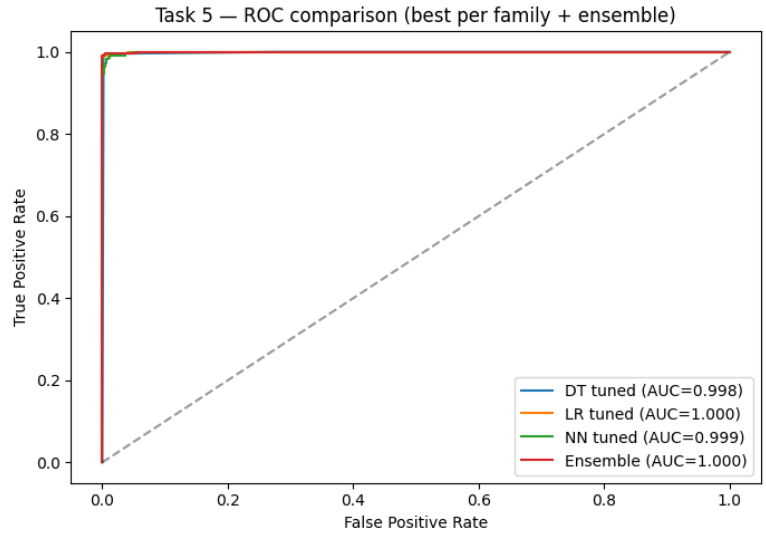
Accuracy and AUC comparison

Model	Test Accuracy	Test AUC
Ensemble (soft voting)	0.995	0.999748
Logistic Regression (tuned)	0.994	0.999700
Neural Network (tuned)	0.988	0.999280
Decision Tree (tuned)	0.995	0.998334

ROC comparison

All four curves are near the top-left, with the Ensemble and LR practically overlapping at the top.

```
=== Task 5 — Accuracy Table (sorted by Test AUC) ===
                     Model  Test Accuracy  Test AUC
     Ensemble (soft voting)          0.995  0.999748
Logistic Regression (tuned)          0.994  0.999700
     Neural Network (tuned)          0.988  0.999280
      Decision Tree (tuned)          0.995  0.998334
```




```
>>> Recommended model: Ensemble (soft voting)  (Test AUC=1.000, Acc=0.995)
Reason: best ranking quality; note the trade-off in complexity/explainability.

Saved files: t5_metrics.csv, t5_roc_all.png
```

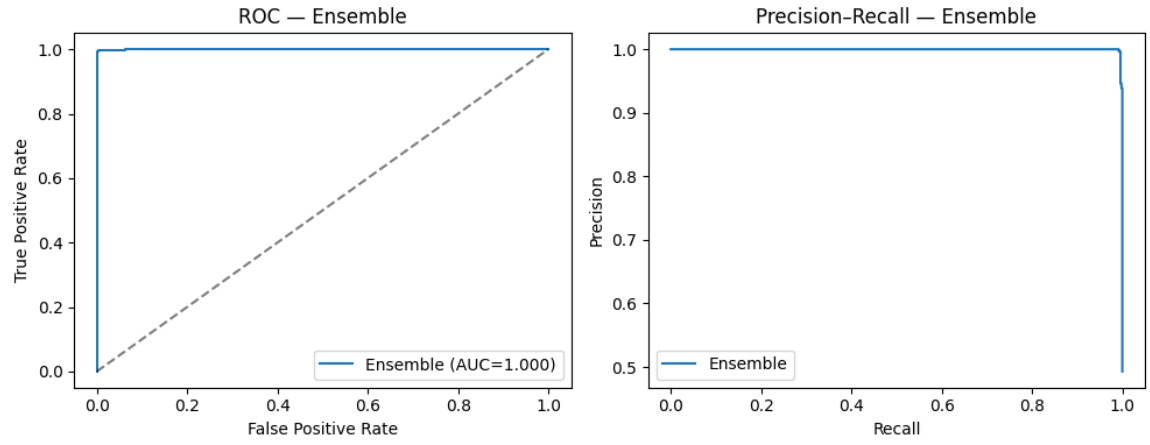
Threshold selection for the chosen model

- Default (0.50): Accuracy 0.995; TN=506, FP=1, FN=4, TP=489. Precision≈0.998, Recall≈0.992 (class 1).
- Youden's J (≈0.388): Accuracy 0.996; TN=505, FP=2, FN=2, TP=491. Precision≈Recall≈F1≈0.996 (balanced).
- F1-max (≈0.388): Same as J* in this run.

Recommendation

Use J* ≈ 0.388 for symmetric costs (it balances sensitivity/specificity and yields the highest F1). If the business cost of missed positives is higher, choose a lower threshold to push up recall; if false alarms are costlier, move the threshold higher to reduce FP.

```
[Quick metrics @0.5]
Model  Acc@0.5      AUC
   LR    0.994 0.999700
   DT    0.995 0.998334
   NN    0.988 0.999280
  ENS    0.995 0.999748

[Ensemble — Threshold comparison]
ThresholdLabel  Threshold  Accuracy  Precision   Recall       F1  TN  FP  FN  TP
          0.50   0.500000     0.995   0.997959 0.991886 0.994914 506   1   4 489
       YoudenJ   0.387826     0.996   0.995943 0.995943 0.995943 505   2   2 491
         F1max   0.387826     0.996   0.995943 0.995943 0.995943 505   2   2 491

=== Classification report — ENS @ 0.50 ===
              precision    recall  f1-score   support

           0      0.992     0.998     0.995       507
           1      0.998     0.992     0.995       493

    accuracy                          0.995      1000
   macro avg      0.995     0.995     0.995      1000
weighted avg      0.995     0.995     0.995      1000

Confusion matrix:
         pred_0  pred_1
true_0      506       1
true_1        4     489

=== Classification report — ENS @ J* (0.388) ===
              precision    recall  f1-score   support

           0      0.996     0.996     0.996       507
           1      0.996     0.996     0.996       493

    accuracy                          0.996      1000
   macro avg      0.996     0.996     0.996      1000
weighted avg      0.996     0.996     0.996      1000

Confusion matrix:
         pred_0  pred_1
true_0      505       2
true_1        2     491

=== Classification report — ENS @ F1* (0.388) ===
              precision    recall  f1-score   support

           0      0.996     0.996     0.996       507
           1      0.996     0.996     0.996       493

    accuracy                          0.996      1000
   macro avg      0.996     0.996     0.996      1000
weighted avg      0.996     0.996     0.996      1000

Confusion matrix:
         pred_0  pred_1
true_0      505       2
true_1        2     491
```




```
Saved: t5q2_threshold_table.csv, t5q2_confusion_matrices.csv, t5q2_pr_roc.png, t5q2_notes.txt
```

Question 2

Decision Tree (tuned)

- **Pros:** Transparent rules; easy to visualise and explain; handles mixed data types; no scaling required.
- **Cons:** Can overfit without pruning/tuning; decision boundaries are axis-aligned (limited expressiveness); small changes in data can alter splits; probability estimates can be poorly calibrated.

Logistic Regression (tuned, with standardisation)

- **Pros**: Fast, stable, highly interpretable (sign and magnitude of coefficients); well-calibrated probabilities; strong baseline; easy to regularise (L1/L2).
- **Cons**: Linear decision boundary in the feature space; requires careful encoding/standardisation; may underfit if the signal is strongly non-linear or highly interactive.

Neural Network (tuned MLP)

- **Pros**: Captures non-linear interactions; flexible architectures; competitive AUC in our data.
- **Cons**: Lower interpretability; sensitive to scaling/hyperparameters; requires early-stopping/regularisation; training time and reproducibility considerations.

Ensemble (soft voting of LR+DT+NN)

- **Pros**: Best overall AUC; leverages complementary strengths; robust to small perturbations; strong default choice when explainability is not the primary constraint.
- **Cons**: Harder to explain (requires model-level or post-hoc explanations); more moving parts to maintain; small extra training time.