LDA Classifier Using Maple

1. Load the Linear Algebra package.

```
> with(LinearAlgebra)
```

(1)

```
[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace,
 CompanionMatrix, CompressedSparseForm, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension,
 Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, FromCompressedSparseForm, FromSplitForm, GaussianElimination, GenerateEquations,
 GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix,
 IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUDecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd,
 MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm,
 Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, ProjectionMatrix, QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension,
 RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, SplitForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve,
 ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]
```

(above are all the commands that support linear algebra)

2. Create your matrices

```
> X1 := Matrix([[1, 2], [4, 6], [5, 8], [7, 2], [5, 5], [6, 8], [9, 1], [3, 3]])
```

$$X1 := \begin{bmatrix} 1 & 2 \\ 4 & 6 \\ 5 & 8 \\ 7 & 2 \\ 5 & 5 \\ 6 & 8 \\ 9 & 1 \\ 3 & 3 \end{bmatrix} \quad (2)$$

```
> X2 := Matrix([[-1,-3], [-4,-5], [-5,-8], [-7,-2], [-5,-6], [-6,-8], [-9,-1]])
```

$$X2 := \begin{bmatrix} -1 & -3 \\ -4 & -5 \\ -5 & -8 \\ -7 & -2 \\ -5 & -6 \\ -6 & -8 \\ -9 & -1 \end{bmatrix} \quad (3)$$

3. Calculate the mean of each class

```
> mu1 := Matrix([[(1+4+5+7+5+6+9+3)/8, (2+6+8+2+5+8+1+3)/8]])
```

$$\mu 1 := \begin{bmatrix} 5 & \frac{35}{8} \end{bmatrix} \quad (4)$$

```
> mu2 := Matrix([[(-1-4-5-7-5-6-9)/7, (-3-5-8-2-6-8-1)/7]])
```

$$\mu 2 := \begin{bmatrix} -\frac{37}{7} & -\frac{33}{7} \end{bmatrix} \quad (5)$$

4. Find the overall mean

```
8/15*mu1[1,1] + 7/15*mu2[1,1]
```

$$\frac{1}{5} \quad (6)$$

```
> 8/15*mu1[1,2] + 7/15*mu2[1,2]
```

$$\frac{2}{15} \quad (7)$$

```
> mu := Matrix([[1/5, 2/15]])
```

$$\mu := \begin{bmatrix} \frac{1}{5} & \frac{2}{15} \end{bmatrix} \quad (8)$$

5. Normalize the data by subtracting the mean from each.

```
> X1o := Matrix([[1 -mu[1, 1] , 2-mu[1, 2]], [4-mu[1, 1], 6-mu[1, 2]], [5-mu[1, 1], 8-mu[1, 2]], [7-mu[1, 1], 2-mu[1, 2]], [5-mu[1, 1], 5-mu[1, 2]], [6-mu[1, 1], 8-mu[1, 2]], [9-mu[1, 1], 1-mu[1, 2]], [3-mu[1, 1], 3
  -mu[1, 2]] ])
```

$$X1o := \begin{bmatrix} \frac{4}{5} & \frac{28}{15} \\ \frac{19}{5} & \frac{88}{15} \\ \frac{24}{5} & \frac{118}{15} \\ \frac{34}{5} & \frac{28}{15} \\ \frac{24}{5} & \frac{73}{15} \\ \frac{29}{5} & \frac{118}{15} \\ \frac{44}{5} & \frac{13}{15} \\ \frac{14}{5} & \frac{43}{15} \end{bmatrix} \quad (9)$$

```
> X2o := Matrix([[-1-mu[1, 1],-3-mu[1, 2]], [-4-mu[1, 1],-5-mu[1, 2]], [-5-mu[1, 1],-8-mu[1, 2]], [-7-mu[1, 1],-2-mu[1, 2]], [-5-mu[1, 1],-6-mu[1, 2]], [-6-mu[1, 1],-8-mu[1, 2]], [-9-mu[1, 1],-1-mu[1, 2]]])
```

$$X2o := \begin{bmatrix} -\frac{6}{5} & -\frac{47}{15} \\ -\frac{21}{5} & -\frac{77}{15} \\ -\frac{26}{5} & -\frac{122}{15} \\ -\frac{36}{5} & -\frac{32}{15} \\ -\frac{26}{5} & -\frac{92}{15} \\ -\frac{31}{5} & -\frac{122}{15} \\ -\frac{46}{5} & -\frac{17}{15} \end{bmatrix} \quad (10)$$

6. Make the covariance matrix for X1. Note that X1^+ is the shortcut for transpose and X1.X2 is the shortcut for matrix multiplication.

> $Cov1 := \frac{X1o^{+}.X1o}{8}$

$$Cov1 := \begin{bmatrix} \frac{2829}{100} & \frac{993}{50} \\ \frac{993}{50} & \frac{44507}{1800} \end{bmatrix} \quad (11)$$

> $Cov2 := \frac{X2o^{+}.X2o}{7}$

$$Cov2 := \begin{bmatrix} \frac{886}{25} & \frac{4393}{175} \\ \frac{4393}{175} & \frac{47683}{1575} \end{bmatrix} \quad (12)$$

7. Add the two covariance matrices

> $Cov := \frac{8}{15} \cdot Cov1 + \frac{7}{15} \cdot Cov2$

$$Cov := \begin{bmatrix} \frac{2372}{75} & \frac{1673}{75} \\ \frac{1673}{75} & \frac{6146}{225} \end{bmatrix} \quad (13)$$

8. Take the inverse of the Covariance

> $Covinv := (Cov)^{-1}$

$$Covinv := \begin{bmatrix} \frac{2634}{35323} & -\frac{2151}{35323} \\ -\frac{2151}{35323} & \frac{21348}{247261} \end{bmatrix} \quad (14)$$

9. Now build and use your classifier.

```
> x1 := Matrix([3, 5]);
```

$$x1 := \begin{bmatrix} 3 & 5 \end{bmatrix} \quad (15)$$

```
> f1 := mu1 . Covinv . x1^+ - 1/2 . mu1.Covinv.mu1^+ + evalf(ln(8/15));
```

$$f1 := \begin{bmatrix} -0.369371583090861 \end{bmatrix} \quad (16)$$

```
> f2 := mu2 . Covinv . x1^+ - 1/2 . mu2.Covinv.mu2^+ + evalf(ln(7/15));
```

$$f2 := \begin{bmatrix} -1.99277802365830 \end{bmatrix} \quad (17)$$

10. Have fun!

```
> x1 := Matrix([-4, -7])
```

$$x1 := \begin{bmatrix} -4 & -7 \end{bmatrix} \quad (18)$$

```
> f1 := mu1 . Covinv . x1^+ - 1/2 . mu1.Covinv.mu1^+ + evalf(ln(8/15));
```

$$f1 := \begin{bmatrix} -1.99340195989917 \end{bmatrix} \quad (19)$$

```
> f2 := mu2 . Covinv . x1^+ - 1/2 . mu2.Covinv.mu2^+ + evalf(ln(7/15));
```

$$f2 := \begin{bmatrix} -0.221500478299930 \end{bmatrix} \quad (20)$$

```
> x1 := Matrix([-4, 3])
```

$$x1 := \begin{bmatrix} -4 & 3 \end{bmatrix} \quad (21)$$

```
> f1 := mu1 . Covinv . x1^+ - 1/2 . mu1.Covinv.mu1^+ + evalf(ln(8/15));
```

$$f1 := \begin{bmatrix} -1.26087640997420 \end{bmatrix} \quad (22)$$

```
> f2 := mu2 . Covinv . x1^+ - 1/2 . mu2.Covinv.mu2^+ + evalf(ln(7/15));
```

$$f2 := \begin{bmatrix} -1.07297205807076 \end{bmatrix} \quad (23)$$

```
>
```