Linear Algebra Background

What this is

- **recap of Linear Algebra concepts**
- **tools to apply toward ML implementation and methodology**

Linear Algebra

scalar	a **value** having **only magnitude** and **not direction** https://languages.oup.com/google-dictionary-en/
vector	a **quantity** with both **magnitude** and **direction** https://en.wikipedia.org/wiki/Vector_(mathematics_and_physics)
subspace	A subset of ***W*** of ***n-space*** is a subspace if: 1. the zero vector is in ***W*** 2. ***x + y*** is in ***W*** whenever ***x*** and ***y*** are in ***W*** 3. ***a*x*** is in ***W*** whenever ***x*** is in ***W*** and ***a*** is any scalar https://www.math.kent.edu/~reichel/glossary
basis	A basis for a **subspace *W*** is a se of **vectors** $v_1, \ldots, v_k$ in ***W*** such that: 1. $v_1, \ldots, v_k$ are linearly independent 2. $v_1, \ldots, v_k$ span ***W*** https://www.math.kent.edu/~reichel/glossary
system of equations	a ***linear system*** is a collection of two or more **linear equations** involving the **same variables**. For example: $\begin{cases} 3x + 2y - z = 1 \\ 2x - 2y + 4z = -2 \\ -x + \frac{1}{2}y - z = 0 \end{cases}$ *https://en.wikipedia.org/wiki/System_of_ 1*
vector spaces	a ***linear space*** is a set whose **elements** (i.e. **vectors**) can be **added** together and **multiplied** by **scalars** https://www.math.kent.edu/~reichel/glossary

outer product	$u \otimes v = uv^T$: the **tensor product** is the **matrix** whose entries are all **products** of an **element** in the **first vector** with an **element** in the **second** vector so that taking the **outer product** of **two vectors** of **length *n*** and ***m*** will result in an ***n* x *m* matrix**
inner product	a **generalization** of the **dot product** and is a way to **multiply vectors** together **resulting** in a **scalar** and **satisfies** the **following** properties: 1. $\langle u+v, w\rangle = \langle u, w\rangle + \langle v, w\rangle$. 2. $\langle \alpha v, w\rangle = \alpha \langle v, w\rangle$. 3. $\langle v, w\rangle = \langle w, v\rangle$. 4. $\langle v, v\rangle \geq 0$ and equal if and only if $v = 0$. *https://mathworld.wolfram.com/InnerProdu 1*
Hadamar product	the **element-wise product** of **two matrices**
matrix multiplication	if ***A*** is an ***m* x *n*** matrix and ***B*** is an ***n* x *p*** matrix, the matrix product ***C* = *AB*** is defined to be an ***m* x *p*** matrix such that: $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}$
norm	given a **vector space *X*** over a **subfield *F*** of the **complex numbers *C***, a **norm** on ***X*** is a real-valued **function *p*** : ***X* -> *R*** with the **following** properties: (where |s| denotes the usual absolute value of a scalar s) 1. $p(x+y) \leq p(x) + p(y)$ for all $x, y \in X$. 2. $p(sx) = \|s\|p(x)$ for all $x \in X$ and all scalars s. 3. positive definiteness for all $x \in X$, if $p(x) = 0$, then $x = 0$. *https://en.wikipedia.org/wiki/Norm_(math 1*
transpose	an operator that **flips** a **matrix** over its **diagonal** denoted $\mathbf{A}^T$

	https://en.wikipedia.org/wiki/Transpose
Eigenvalue	an **eigenvalue** of a ***n-by-n*** **matrix** A is a **scalar *c*** such that ***A*x* = *c*x*** holds for some nonzero **vector *x*** (where ***x*** is an ***n-tuple***) https://www.math.kent.edu/~reichel/glossary
Eigenvector	an **eigenvector** of an ***n-by-b*** **matrix *A*** is a nonzero **vector *x*** such that ***A*x* = *c*x*** holds for some **scalar** https://www.math.kent.edu/~reichel/glossary
Eigendecomposition	the **factorization** of a **matrix** into a canonical form, whereby the **matrix** is represented in **terms** of its **eigenvalues** and **eigenvectors** https://www.math.kent.edu/~reichel/glossary
trace	the **sum** of its **eigenvalues** counted with multiplicities such that: 1. ***tr(AB) = tr(BA)*** for any same-sized **matrices *A*** and ***B*** 2. thus, **similar matrices** have the **same trace** 3. $\operatorname{tr}(\mathbf{A}) = \sum_{i=1}^{n} a_{ii} = a_{11} + a_{22} + \cdots + a_{nn}$ *https://en.wikipedia.org/wiki/Trace_(lin 1*
norm to distance	*d(x, y)* = **||*x* – *y*||**, where '|| . ||' denotes **magnitude** and ' – ' denotes **difference**
Euclidean Distance	$d(p, q)^2 = (q_1 - p_1)^2 + (q_2 - p_2)^2$ $\|x - y\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$ $\|\|x - y\|\|_2^2 = (x-y)^T(x-y)$ $= x^T x - 2x^T y - y^T y$
Holder's Inequality	$\|\|\mathbf{x}\|\|_p := \left(\sum_{i=1}^{n} \|x_i\|^p \right)^{1/p}$ $p \geq 1$
vector space axioms	**associativity** ***u + (v + w) = (u + v) + w***

	commutativity	***u + v = v + u***
	identity element	there exists an element ***0 ∈ V,*** called the ***zero vector*** such that ***v + 0 = v*** for all ***v ∈ V***
	inverse elements	for every ***v ∈ V*** there **exists** an element ***-v ∈ V***, called the ***additive inverse*** of ***v***, such that ***v + (-v) = 0***
	scalar-multiplication / field-multiplication compatibility	***a(bv) = (ab)v***
	identity element of scalar multiplication	***I · v = v***, where ***I*** denotes the **multiplicative identity** in ***F***
	distributivity	***a(u + v) = au + av***
spatial vectors	**vectors** in an **n-dimensional vector space**	
vectorization	turns a **matrix** into a **vector** so that an ***n x m*** **matrix** will **produce** a ***n*m*** length **vector**	
submatrix	a grouped **subset** of a **matrix**	
block matrix	a subset of **non-overlapping submatrices**	
determinant	the product of the eigenvectors of a matrix $$\det(A) = \prod_{i=1}^{n} \lambda_i = \lambda_1 \lambda_2 \cdots \lambda_n$$	

neural networks	$\text{Layer } \ell - 1 \mapsto \text{Layer } \ell$ $\vec{\varphi}^{(\ell)} = \sigma\left(W^{(\ell)}\vec{\varphi}^{(\ell-1)} + \vec{b}^{(\ell)}\right)$
losses	$\lVert X\vec{\beta} - Y \rVert^2$
multivariate normal pdf	$(2\pi)^{-k/2} \det(\boldsymbol{\Sigma})^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$
dimensionality reduction	simplifies **complex**, **high-dimensional** data by **transforming** it into a **lower-dimensional** space
span of set	all the **vectors** obtained by **linearly combining** a set of vectors ***S = {v₁, v₂, ..., vₙ},*** such that $\operatorname{span}(S) = \left\{ \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \;\middle|\; \lambda_i \in \mathbb{R} \right\}$
column span	***colsp(A)*** = ***span({v₁, v₂, ..., aₙ})***
column rank	***rank(A)*** = ***dim(colsp(A))***
null space	**set** of all **vectors *x*** of a **matrix *A*** for which ***Ax = 0***
nullity	the **dimension** of the **null space**
rank-nullity relationship	for **matrix *A*** with ***n* columns**: ***rank(A) + nullity(A) = n***
orthonormality	a set of vectors {u₁, u₂, ..., un} is orthonormal iff: $\forall i, j : \langle u_i, u_j \rangle = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$
Kronecker delta	δ_{ij} is a mathematical **function** that acts as a discrete "**switch**," returning **1** if its two indices ***i*** and ***j*** are the **same**, and ***0*** if they are **different**
Gram-Schmidt theorem	if ***{v₁, v₂, ..., vₙ}*** is a **linearly independent** list of **vectors** in an **inner-product space *V*,** then there exists an **orthonormal list *{e₁, e₂, ..., eₙ}*** of vectors ***V*** such that **span(*e₁, e₂, ..., eₙ*)** = **span(*v₁, v₂, ..., v₃*)**
matrix inverse	an ***n-by-n*** square **matrix *A*** is **invertible**, if there exists an ***n-by-n*** square **matrix *B*** such that ***AB = BA = Iₙ*** where In denotes the **n-by-n** identity matrix

matrix inverse equivalent statements	Let ***A*** be a square **matrix** • there is an ***n-by-n*** matrix **B** such that $AB = I_n = BA$ • matrix **A** has a **left inverse** and a **right inverse**, in which case **both** left and right **inverses exist** and $B = C = A^{-1}$ • A has full rand; that is, rank A = n • A is invertible, that is, A has an inverse, is nonsingular, and is nondegenerate
Probability and Statistics	
discrete distribution	describes the **probabilities** of **outcomes** for **discrete** random **variables** where each **outcome** has a specific **probability** between ***0* and *1*** and all probabilities **sum** to ***1*** https://en.wikipedia.org/wiki/Probability_distribution
continuous distribution	describes **probability** for **variables** that can take **any value** within a **range** https://en.wikipedia.org/wiki/Probability_distribution
discrete random variable	a random **variable** that has a **countable** range and **assumes** each **value** in this **range** with a **positive probability** https://gwthomas.github.io/docs/math4ml.pdf
continuous random variable	a random **variable** that has an **uncountable** range and **assumes** each **value** in this **range** with **probability zero** https://gwthomas.github.io/docs/math4ml.pdf
probability mass function	gives the **probability** that a **discrete** random **variable** is exactly **equal** to a specific **value** $x \in \Omega \quad \text{discrete } \Omega$ $p_X(x) = P(X = x) \qquad \sum_x p_X(x) = 1 \qquad p_X(x) \geq 0$ https://en.wikipedia.org/wiki/Probability_mass_function#:~:text=In%20and%20statistics%2C%20a,mass%20is%20called%20the%20mode.
probability density function	describes the **likelihood** of a **continuous** random **variable** falling **within** specific **range**, represented as a **curve** where the total **area** under it **equals** ***1***, and the **area** over an interval **gives** the **probability** https://en.wikipedia.org/wiki/Probability_density_function#:~:text=In%20probability%20theory%2C%20a%20probability,possible%20values%20to%20begin%20with.

joint distribution	a **distribution** over some **combination** of several random **variables** https://gwthomas.github.io/docs/math4ml.pdf
independence	the likelihood of one random variable X_i is not a condition of X_j $X_i \perp\!\!\!\perp X_j$ $p(x_i, x_j) = p(x_i)p(x_j)$
Bayes Rule	connects **conditionals** in one **direction** to **conditionals** in another **direction** $p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}$
Bayes Theorem	The first step into solving Bayes' theorem problems is to assign letters to events: • A = chance of having the faulty gene. That was given in the question as 1%. That also means the probability of *not* having the gene (~A) is 99%. • X = A positive test result. So: $p(A \mid X) = \frac{p(X \mid A)p(A)}{p(X \mid A)p(A) + p(X \mid \neg A)p(\neg A)}$ 1. P(A|X) = Probability of having the gene given a positive test result. 2. P(X|A) = Chance of a positive test result given that the person actually has the gene. That was given in the question as 90%. 3. p(X|~A) = Chance of a positive test if the person *doesn't* have the gene. That was given in the question as 9.6%
conditional **likelihood**	the **probability** that ***Y***, given ***X* = *x*** $p(y \mid x) = \frac{p(x, y)}{p(x)}$
conditional **independence**	shows that X_i is independent of X_j given X_k $X_i \perp\!\!\!\perp X_j \mid X_k$ $p(x_i, x_j \mid x_k) = p(x_i \mid x_k)p(x_j \mid x_k)$
parameters	list of specific values needed to calculate a parametric distribution
parametric **distributions**	calculated probability given a list of parametric values

	$p(x) \equiv p_\theta(x) \equiv p(x \mid \theta)$
parametric families	all the possible distributions that can be calculated by adjusting parametric values. some examples of parametric families: normal (Gaussian): $p(x \mid \theta) \equiv p(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ truncated normal: $p(x \mid \theta) \equiv p(x \mid \mu, \sigma, a, b) = \frac{\phi(\frac{x-\mu}{\sigma})}{\sigma\left(\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})\right)}$ multivariate normal: $p(x \mid \theta) \equiv p(x \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp\left(-\frac{1}{2}(x-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k \lvert\boldsymbol{\Sigma}\rvert}}$ logistic: $p(x \mid \theta) \equiv p(x \mid \mu, s) = \frac{e^{-\frac{x-\mu}{s}}}{s\left(1+e^{-\frac{x-\mu}{s}}\right)^2}$ beta: $p(x \mid \theta) \equiv p(x \mid \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$ binomial (parametric pmf): $\Pr(k; n, p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$
conditional distribution	show the **probability** of **outcomes** for one **variable**, given that another **variable** is **fixed** at a specific **value** or falls within a certain **category**, essentially focusing on a **sub-population** https://www.khanacademy.org/math/ap-statistics/analyzing-categorical-ap/distributions-two-way-tables/v/marginal-distribution-and-conditional-distribution#:~:text=On%20the%20other%20hand%2C%20the,of%20car%20origin%20and%20color.
marginalization	the **probability** of the **variables** contained in a **subset** of a **collection** of random **variables** is **determined** by **integrating** out additional variables https://en.wikipedia.org/wiki/Marginal_distribution
statistical expectations	***E[X]*** is the long-run **average outcome** of a random **variable**, calculated as **weighted average** of its **possible values** where each **value** is **weighted** by its **probability**
expected value	an **expected** value $\boldsymbol{E_X}$ from a random **distribution *p*** taken as a function ***f(X)*** $\mathbb{E}_{X \sim p}\left[f(X)\right] = \int_\Omega f(x)p(x)\mathrm{d}x$
KL Divergence	a **Kullback–Leibler divergence** is a statistical **distance**: a measure of how much an **approximating** probability **distribution Q** differs from the **true value** **KL properties**:

	1. **measure** of how **different** two probability **distributions** are 2. ***D(p||q)* ≥ *0; D(p||q)*** = 0 iff ***p = q*** 3. **not** a **metric**; not **commutative**, does not **satisfy triangle equality** 4. the **average** number of **bits** that are **wasted** by **encoding events** from a **distribution *p*** with a code based on a **not-quite-right distribution** of ***q*** $$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$$ https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence
estimating expectations	given samples $\{x_1, x_2, .., x_n\}$ of ***p***, an **empirical average** of ***f(x)*** is calculated to **approximate** the true **value** of ***f(x)*** $$\mathbb{E}_{X \sim p}[f(X)] \approx \frac{1}{n} \sum_{i=1}^{n} f(x_i)$$ $$x_1 \sim p, x_2 \sim p, \ldots, x_n \sim p$$

Live Session	
Introduction	**12 Jan 2026**
Instructor	Rei Sanchez-Arias
Email	reisanar@unc.edu
Website	https://www.reisanar.com/
Office Hours	Mondays 12:00 pm to 1:00 pm
Live Session	Monday 6:00 pm to 7:30 pm
Linear Algebra and Probability	
Look up Terms	see below

Look up Terms

- **Hessian matrix**
  - the **Hessian** matrix of a **scalar function** of several **variables** describes the **curvature** of that function
  - by taking the **determinant** of the **Hessian** matrix at a critical point, we can **test** whether that **point** is a local **min**, **max**, or **saddle** point

$$\text{Hessian}(f(x,y)) = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix}$$

https://www.mit.edu/~ashrstnv/hessian-ma 1

- **Jacobian matrix**
  - a matrix of all the **first-order** partial **derivatives** of a **vector-valued** function, acting as a **multivariable** function's **derivative**, showing how **changes** in input **variables** affect **output** variables locally
  - Essential for **gradient descent** in **training** neural **networks**, calculating **sensitivity**, and **backpropagation**

$$J(u,v) = \begin{bmatrix} x_u & x_v \\ y_u & y_v \end{bmatrix}$$

https://math.etsu.edu/multicalc/prealpha 1

- **covariant** matrix
  - a **square matrix** giving the **covariance** between each **pair** of **elements** of a given **random** vector
  - in the **matrix diagonal** there are **variances**, i.e., the **covariance** of each **element** with **itself**
  - **generalizes** the notion of **variance** to **multiple dimensions**

    https://ise.ncsu.edu/wp-content/uploads/sites/9/2022/01/Covariance-matrix-Wikipedia-1.pdf

	• **gradient**
	  ○ **(∇f)** a **vector** that points in the **direction** of the function's **steepest increase** and whose **magnitude** represents the **rate** of that **increase**
	  ○ calculated by **collecting partial derivatives** into a **vector**
	$$grad\ f(x,y,z) = \nabla f(x,y,z) = \frac{\partial f}{\partial x} i + \frac{\partial f}{\partial y} j + \frac{\partial f}{\partial z} k$$
	https://byjus.com/maths/gradient/#:~:tex 1

Look up on SciKitLearn	• **QuadraticDiscriminantAnalysis**
	  ○ Quadratic Discriminant Analysis.
	  ○ A **classifier** with a **quadratic decision boundary**, generated by **fitting** class **conditional densities** to the **data** and using **Bayes's rule**.
	  ○ The model **fits** a **Gaussian density** to each **class.**
	• **BernoulliNB**
	  ○ Naive **Bayes** classifier for **multivariate Bebrnoulli models**.
	  ○ Like **MultinomialNB**, this classifier is **suitable** for **discrete data**. The **difference** is that while **MultinomialNB** works with occurrence counts, **BernoulliNB** is designed for **binbary/boolean** features.
	• **MultinomialNB**
	  ○ Naive **Bayes** classifier for **multinomial models**.
	  ○ The **multinomial** Naive **Bayes** classifier is suitable for **classification** with **discrete** features (e.g., **word counts** for text **classification**). The **multinomial distribution** normally **requires** integer feature **counts**. However, in practice, **fractional counts** such as **tf-idf** may also work.

Class Work Exercises	
Class Work Notebooks	CW-DATA780-unit_02.ipynb CW-DATA780-unit_02-stats.ipynb CW-tensorflow_basics_practice.ipynb
CW Notebook Solutions	DATA780-unit_02-lin_alg-1-1.html DATA780-unit_02-stats.html tensorflow_basics_practice.html
Solution	**DATA780\Week2\<fname_lname>_DATA780_week2_ProbStatML.docx"**