A Project Report on

"The Role of Data Science in Pharmaceutical Industry"

A practice school report submitted to the Institute of Pharmacy, Bundelkhand University, Jhansi in partial fulfillment of the award of the degree of

Bachelor of Pharmacy

2024

UNDER THE SUPERVISION

Guide by
Dr. Girish Chand Soni
Associate Professor
Institute of Pharmacy,
B. U. Jhansi (U. P.)

Submitted by
Dheeraj Tripathi
B. Pharma 4th year
Institute of Pharmacy
B. U. Jhansi (U.P.)

INSTITUTE OF PHARMACY

BUNDELKHAND UNIVERSITY, JHANSI, (U.P.) INDIA

Roll No. 201251016027

BUNDELKHAND UNIVERSITY, JHANSI

Institute of pharmacy

2024

Certificate

This is to certify that the project entitled " *The Role of Data Science in Pharmaceutical industry*" submitted in partial fulfillment of the requirement for the Bachelor of Pharmacy, Institute of Pharmacy, Bundelkhand University, Jhansi is a bonafide work carried out by **Dheeraj Tripathi** during the academic session 2024.

Date:

H.O.D

Dr. Peeyush Bhardwaj
Professor and Head
Institute of Pharmacy
B. U. Jhansi (U.P.)

BUNDELKHAND UNIVERSITY, JHANSI

Institute of pharmacy

2024

Certificate

This is to certify that the project entitled " **The Role of Data Science in Pharmaceutical industry** " submitted in partial fulfillment of the requirement for the Bachelor of Pharmacy, Institute of Pharmacy, Bundelkhand University, Jhansi is a bonafide work carried out by **Dheeraj Tripathi** during the academic session 2024.

Date:

Guide

Dr. Girish chand soni
Associate Professor
Institute of Pharmacy
B. U. Jhansi (U.P.)

BUNDELKHAND UNIVERSITY, JHANSI

Institute of pharmacy

2024

Declaration

This is to certify that the project entitled "*The Role of Data Science in Pharmaceuticals Industry* " is prepared by me under the direct guidance and supervision of **Dr. Girish Chand Soni** (Associate Professor) Institute of Pharmacy, Bundelkhand University, Jhansi.

The same is submitted to Bundelkhand University, Jhansi in partial fulfillment of the requirement for the award of the degree of Bachelor of Pharmacy.

I further declare that I have not submitted this project report previously for the award of any degree or diploma to me.

Date: **Dheeraj Tripathi**

ACKNOWLEDGEMENT

Firstly, I would like to thank GOD. Most gracious and most merciful the omnipotent , the omniscient who gave me the strength to complete this work, without his generous blessing it was not possible to complete this work.

Words are indeed insufficient to express the depth of profound gratitude to Dr. Peeyush Bhardwaj Sir (H.O.D) of Institute of Pharmacy, Bundelkhand University, Jhansi (U.P.) for his valuable guidance, constructive criticism, and encouragement during the entire course.

I am extremely grateful to Dr. Girish chand soni sir Institute of Pharmacy, Bundelkhand University Jhansi (U.P) to be thoroughly involved in solving my problems during the entire academic session and for magnanimity in helping me to complete my choose project **"The Role of Data Science in Pharmaceutical Industry"** at Institute of pharmacy, Bundelkhand University; Jhansi.

Above all, I express my deep-hearted gratitude to my esteemed parents, whose emotion and moral fragrance always empowered me to carry on my studies.

Dheeraj Tripathi

Table of content

Abstract

Data science is an interdisciplinary field that extracts knowledge and insights from many structural and unstructured data,using scientific methods, data mining techniques, machine-learning algorithms, and big data. The healthcare industry generates large datasets of useful information on patient demography, treatment plans, results of medical examinations, insurance,etc. The data collected from the Internet of Things (IoT) devices attract the attention of data scientists. Data science provides aid to process, manage, analyze, and assimilate the large quantities of fragmented, structured, and unstructured data created by healthcare systems. This data requires effective management and analysis to acquire factual results. The process of data cleansing, data mining, data preparation, and data analysis used in healthcare applications is reviewed and discussed in the article. The article provides an insight into the status and prospects of big data analytics in healthcare, highlights the advantages, describes the frameworks and techniques used, briefs about the challenges faced currently, and discusses viable solutions. Data science and big data analytics can provide practical insights and aid in the decision-making of strategic decisions concerning the health system. It helps build a comprehensive view of patients, consumers, and clinicians. Data-driven decision-making opens up new possibilities to boost healthcare quality.

Applications of Data science in Pharmaceutical Industry

AI and Data Science may be used in practically every element of pharmaceutical production, from medication discovery to marketing. Incorporating AI technology into fundamental workflows can help pharma organisations run more efficiently, cost effectively, and smoothly. The best part is that AI systems are meant to improve outcomes as they learn from fresh data and experience, making them a formidable tool in pharmaceutical research and development.

Here are a few notable Data Science uses in the pharmaceutical industry:

R & D

Pharma businesses deploy powerful ML and AI algorithms to streamline drug discovery procedures. These intelligence technologies are meant to find complex patterns in huge data sets to tackle biological network problems. This is a great approach to look at disease patterns and see which treatments work best for which ailments. Thus, pharmaceutical companies can focus on developing medicines that are most likely to treat a disease or medical condition.

Drug development

AI has the potential to improve R&D. AI can accomplish everything from discover novel compounds to validate and identify targeted treatments. The MIT study found that only 13.8% of medicines pass clinical testing. A pharmaceutical (Nagaprasad et al.; JPRI, 33(46A): 6-14, 2021; Article no.JPRI.74285) 10 company must also approve a medicine after completing a clinical trial that costs between $161 million and $2 billion. Pharmaceutical companies increasingly use AI to increase new drug success rates, create more cheap ad treatments, and most importantly, lower operating expenses.

Drug Discovery:

Machine learning models aid in identifying potential drug candidates by predicting molecular properties, binding affinities, and biological activities.

Molecular Docking:
ML techniques help in predicting the binding modes and interactions between drug molecules and target proteins.

Quantitative Structure-Activity Relationship (QSAR):
ML models predict the biological activity of molecules based on their chemical structure, aiding in lead optimization and prioritization.

Pharmacokinetics and Pharmacodynamics Modeling:
ML techniques predict drug absorption, distribution, metabolism, and excretion (ADME) properties, as well as pharmacological effects.

Predictive Toxicology:
ML models predict the potential toxicity of drug candidates and help prioritize compounds for further testing.

Personalized Medicine:
ML algorithms analyze patient data to tailor treatments based on individual characteristics, such as genetics, biomarkers, and medical history.

Clinical Trial Optimization:
ML models optimize clinical trial design, patient recruitment, and monitoring, leading to more efficient and cost-effective trials.

Disease Diagnosis and Prognosis:
ML techniques analyze patient data, including medical images, genomic data, and clinical records, to assist in disease diagnosis, staging, and outcome prediction.

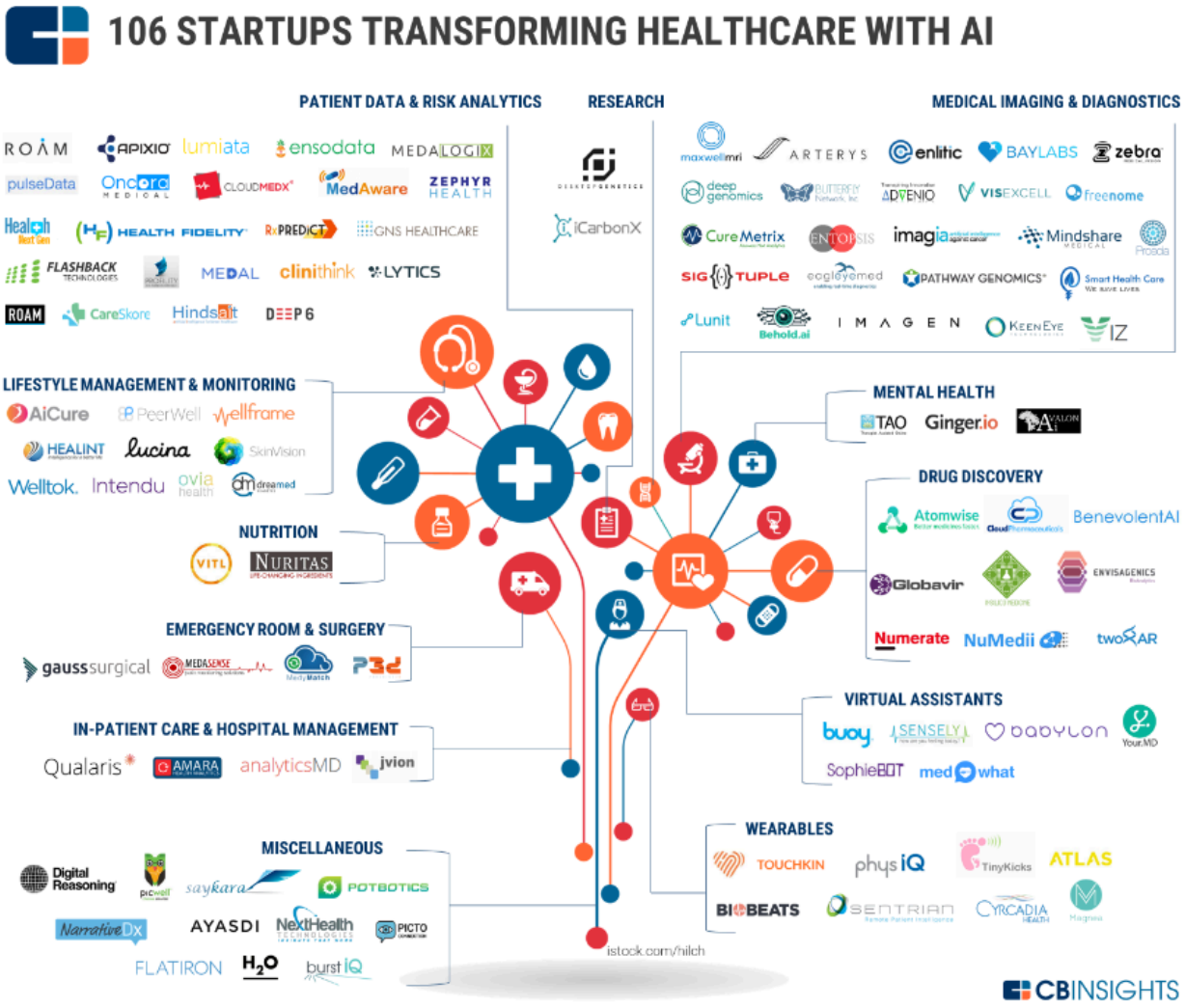
Healthcare Data Analytics:
ML algorithms analyze healthcare data to uncover patterns, trends, and insights for drug discovery, epidemiological studies, and healthcare management.

Drug Safety Surveillance:
ML algorithms monitor adverse drug reactions, drug-drug interactions, and medication errors to ensure patient safety.

Genomic and Proteomic Analysis:
ML techniques analyze omics data to identify biomarkers, drug targets, and pathways associated with disease and drug response.

Healthcare companies working in data science and health improvement

Pharmacovigilance requires tracking and identifcation of adverse drug reactions (ADRs) after launch, to guarantee patient safety. ADR events' approximate social cost per year reaches a billion dollars, showing it as a signifcant aspect of the medical care system [46]. Data mining fndings from adverse event reports (AERs) revealed that mild to lethal reactions might be caused in paclitaxel among which docetaxel is linked with the lethal reaction while the remaining 4 drugs were not associated with hypersensitivity [47] while testing ADR's "hypersensitivity" to six anticancer agents [47].



Model 1 : COVID-19, Proteins identification with Biopython.

Using Biopython for COVID-19 protein identification is a commendable approach in the realm of bioinformatics and molecular biology. Biopython offers a suite of tools and libraries that facilitate the analysis of biological data, including sequence manipulation, structure analysis, and protein identification. In the context of COVID-19, identifying and characterizing viral proteins is crucial for understanding viral pathogenesis, designing therapeutics, and developing vaccines.

The model prediction process likely involves several steps: data retrieval of COVID-19 genome sequences, translation of genomic sequences into protein sequences, annotation of protein functions and domains, and possibly structural analysis to identify potential drug targets or vaccine candidates. Biopython's rich functionalities streamline these tasks, allowing researchers to efficiently analyze large datasets and extract meaningful insights.

In [2]:
```
from Bio.SeqRecord import SeqRecord
from Bio import SeqIO
covid19 = SeqIO.read('/kaggle/input/coronavirus-genome-sequence/MN908947.fna', "fasta")
```

In [3]:
```
print(f'The genome of the virus causing Covid-19 (known as SARS-CoV-2) consists of {len(covid19)}
genetic bases or letters.')
```

```
The genome of the virus causing Covid-19 (known as SARS-CoV-2) consists of 29903 genetic bases
or letters.
```

In [4]:
```
covid_DNA= covid19.seq
print( covid_DNA[:200])
```

```
ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCAACTTTCGATCTCTTGTAGATCTGTTCTCTAAACGAACTTTAAAATCTGTGTGGCTGT
CACTCGGCTGCATGCTTAGTGCACTCACGCAGTATAATTAATAACTAATTACTGTCGTTGACAGGACACGAGTAACTCGTCTATCTTCTGCAGGC
TGCTTACGGT
```

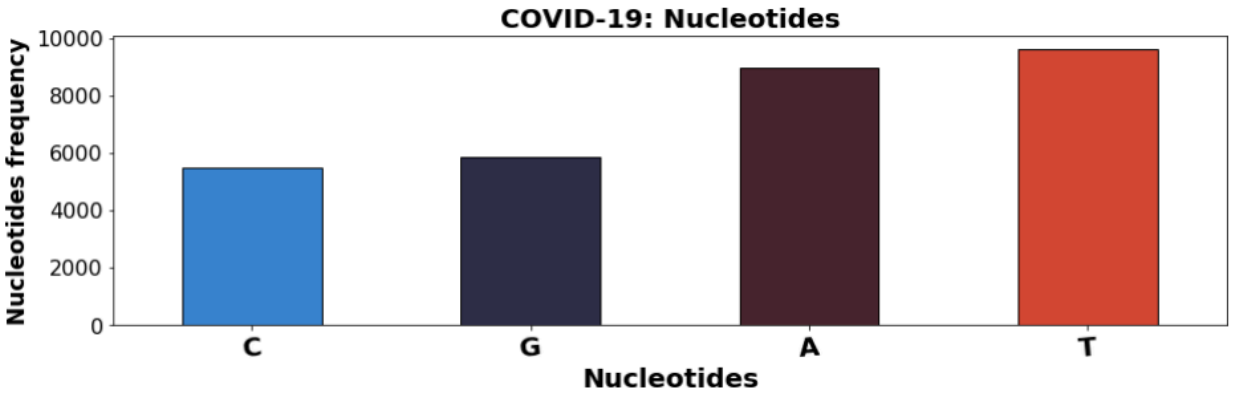
In [5]:
```
#Count the nucleotides frequency in the DNA
DNA= covid_DNA
nucleotides={}
for n in DNA:
    if n in nucleotides:
        nucleotides[n] += 1
    else:
        nucleotides[n] =  1
print(nucleotides)

#Create a dataframe
nts= pd.DataFrame(data=nucleotides, index=[0]).T.reset_index()
nts= nts.rename(columns={0: 'frequency', 'index': 'nucleotides'})
nts=nts.sort_values(by=['frequency'], ascending=True)
```

The graph representing the ratio of nucleotide sequences in model building serves as a critical visualization tool, offering insights into the composition and distribution of genetic information utilized for model development. By depicting the relative abundance of adenine (A), thymine (T), cytosine (C), and guanine (G) nucleotides within the dataset, the graph provides a snapshot of the genetic diversity and sequence characteristics underpinning the model. A balanced distribution among the nucleotides suggests a diverse and

representative dataset, facilitating robust model training and generalization. Conversely, skewness or irregular patterns in the nucleotide ratio may indicate biases or limitations in the dataset, warranting further investigation and potential adjustments to ensure model accuracy and reliability. Thus, the graph serves as a visual aid for researchers to assess the quality and suitability of the nucleotide dataset for subsequent model building and analysis, guiding informed decisions in the pursuit of scientific inquiry and discovery.

```
{'A': 8954, 'T': 9594, 'G': 5863, 'C': 5492}
```



In [7]:
```
covid_mRNA = covid_DNA.transcribe()
print(covid_mRNA[:100])
```

```
AUUAAAGGUUUAUACCUUCCCAGGUAACAAACCAACCAACUUUCGAUCUCUUGUAGAUCUGUUCUCUAAACGAACUUUAAAAUCUGUGUGGCUGU
CACUC
```

In [8]:
```
print(f'Covid-19 DNA: {covid_DNA[:50]}')
print(f'Covid-19 RNA: {covid_mRNA[:50]}')
```

```
Covid-19 DNA: ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCAACTTTCGATCTC
Covid-19 RNA: AUUAAAGGUUUAUACCUUCCCAGGUAACAAACCAACCAACUUUCGAUCUC
```

Translation :

is the process that takes the information passed from DNA as messenger RNA and turns it into a series of **amino acids**.It is essentially a translation from one code **(nucleotide A T C G sequence)** to another code **(amino acid sequence)**.How does this translation happen? As in any language, we need a dictionary for translation, in this

case the amino acid dictionary is the table below. The nucleotides are read in groups of three "AUG GCC CAG UUA ...". Each triplet is called a codon and codes for a specific amino acid.

There are 61 codons for 20 amino acids, and each of them is "read" to specify a certain amino acid out of the 20 commonly found in proteins.One codon, AUG, specifies the amino acid methionine and also acts as a start codon to signal the start of protein construction.There are three more codons that do not specify amino acids. These stop codons, UAA, UAG, and UGA, tell the cell when a polypeptide is complete. All together, this collection of codon-amino acid relationships is called the genetic code, because it lets cells “decode” an mRNA into a chain of amino acids.

Luckily, with the translate() function, python does translate the mRNA to amino acids chains. Chains are separated with a * which is the stop codon (UAA, UAG and UGA) [5].

In [9]:
```
covid_aa = covid_mRNA.translate()
print(covid_aa[:99])
```

```
IKGLYLPR*QTNQLSISCRSVL*TNFKICVAVTRLHA*CTHAV*LITNYCR*QDTSNSSIFCRLLTVSSVLQPIISTSRFRPGVTER*DGEPCPW
FQRE
```

Let's break down each set of amino acids and identify them by their one-letter abbreviations and corresponding full names:

Set	Amino Acid Sequence	Amino Acid Abbreviations	Amino Acid Names
1	IKGLYLPR	I K G L Y L P R	Isoleucine ,Lysine ,Glycine Leucine ,Tyrosine, Leucine Proline Arginine
2	QTNQLSISCRSVL	Q T N Q L S I S C R S V L	Glutamine Threonine Asparagine Glutamine Leucine Serine Isoleucine Serine Cysteine Arginine Serine Valine Leucine

3	TNFKICVAVTRLHA	T N F K I C V A V T R L H A	Threonine Asparagine Phenylalanine Lysine Isoleucine Cysteine Valine Alanine Valine Threonine Arginine Leucine Histidine Alanine
4	CTHAVLITNYCR	C T H A V L I T N Y C R	Cysteine Threonine Histidine Alanine Valine Leucine Isoleucine Threonine Asparagine Tyrosine Cysteine Arginine
5	QDTSNSSIFCRLLTVSSVLQPIISTSRFRPGVTER	Q D T S N S S I F C R L L T V S S V L Q P I I S T S R F R P G V T E R	Glutamine Aspartic Acid Threonine Serine Asparagine Serine Serine Isoleucine Phenylalanine Cysteine Arginine Leucine Leucine Threonine Valine Serine Serine Valine Leucine Glutamine Proline Isoleucine Isoleucine Serine Threonine Serine Arginine Phenylalanine Arginine Proline Glycine Valine Threonine Glutamic Acid Arginine
6	DGEPCPWFQRE	D G E P C P W F Q R E	Aspartic Acid Glycine Glutamic Acid Proline Cysteine Proline Tryptophan Phenylalanine Glutamine Arginine Glutamic Acid

It's worth to mention that not all the amino acids sequences are proteins. Only the sequences with more than 20 amino acids code for functional proteins. The short amino acid sequences are oligopeptides and have other functionalities. Here, we will focus on the chains with more than 20 amino acid chains.

Sequence with AA more than 50 are :

In [13]:
```
for i in Proteins[:]:
    if len(i) < 50:
        Proteins.remove(i)
```

In [14]:
```
print(f'We have {len(Proteins)} proteins with  more than 50 amino acids in the covid-19 genome ')
```

```
We have 5 proteins with  more than 50 amino acids in the covid-19 genome
```

In [15]:
```
proteinas=pd.DataFrame(Proteins)
proteinas['amino acid sequence'] = proteinas[0].apply(str)
proteinas['Protein length'] = proteinas[0].apply(len)
proteinas.rename(columns={0: "sequence"}, inplace=True)
pro=proteinas.drop('sequence', axis=1)
pro_= pro.sort_values(by=['Protein length'], ascending=False)
```

In [16]:
```
pd.options.display.max_colwidth = 80
import seaborn as sns
cm = sns.light_palette("green", as_cmap=True)

s = pro_.style.background_gradient(cmap=cm)
s
```

Out[16]:

In [17]:
```
pro_
```

Out[17]:

	amino acid sequence	Protein length
0	CTIVFKRVCGVSAARLTPCGTGTSTDVVYRAFDIYNDKVAGFAKFLKTNCCRFQEKDEDDNLIDSYFVVKRHTFSN...	2701
1	ASAQRSQITLHINELMDLFMRIFTIGTVTLKQGEIKDATPSDFVRATATIPIQASLPFGWLIVGVALLAVFQSASK...	290
4	TNMKIILFLALITLATCELYHYQECVRGTTVLLKEPCSSGTYEGNSPFHPLADNKFALTCFSTQFAFACPDGVKHV...	123
2	AQADEYELMYSFVSEETGTLIVNSVLLFLAFVVFLLVTLAILTALRLCAYCCNIVNVSLVKPSFYVYSRVKNLNSS...	83
3	QQMFHLVDFQVTIAEILLIIMRTFKVSIWNLDYIINLIIKNLSKSLTENKYSQLDEEQPMEID	63

Now that we have the protein sequences, we will use the BLAST search.

BLAST (basic local alignment search tool) is an algorithm and program for comparing primary biological sequence information, such as the amino-acid sequences of proteins or the nucleotides of DNA and/or RNA sequences. A BLAST search enables a researcher to compare a subject protein or nucleotide sequence (called a query) with a

library or database of sequences, and identify library sequences that resemble the query sequence above a certain threshold.

In other words, we will try to find the protein sequences already available in the databases that are the most similar to our protein sequences. *(Hint: In this case, most probably the proteins that will have the highest similarity with our Covid-19 belong to the SARS coronavirus or Bat coronavirus).*

BLAST search results: COVID-19 proteins

Out[18]:

	Protein length	DB:ID	protein	organism	match	Function
0	2701	P0C6X7	Replicase polyprotein 1ab	Human SARS coronavirus (SARS-CoV)	96%	Multifunctional protein involved in the transcription and replication of viral RNAs. Contains the proteinases responsible for the cleavages of the polyprotein.
1	290	Q0Q474	Protein 3	Bat coronavirus 279/2005 (BtCoV)	75%	Forms homotetrameric potassium sensitive ion channels (viroporin) and may modulate virus release
2	123	Q3I5J0	Protein 7a	Bat coronavirus Rp3/2004	89%	Non-structural protein which is dispensable for virus replication in cell culture.
3	83	P59637	Envelope small membrane protein	Human SARS coronavirus (SARS-CoV)	95%	Plays a central role in virus morphogenesis and assembly. Acts as a viroporin and self-assembles in host membranes forming pentameric protein-lipid pores that allow ion transport.
4	63	Q3I5J1	Non-structural protein 6	Bat coronavirus Rp3/2004	69%	Could be a determinant of virus virulence. Seems to stimulate cellular DNA synthesis in vitro (By similarity).

The table above shows the BLAST search results of the 5 amino acid chains obtained from the COVID-19 genome. As expected, all the viral proteins have high similarities with viral proteins in SARS and Bat coronaviruses.

Model 2 : " Kinds of Function Used to solve Genetics based Machine Learning model "

Selected applications of Machine learning in Genetics and Genomics.

DNA sequence	**Identify Transcription start sites , splices , exon etc.**
DNA sequence	**Identify TF binding site**
DNA sequence	**identify gene**
Gene Expression	**Predict regulatory relationship**
Gene expression data	**Identify biomarker for a disease**
DNA sequence + Gene Expression	**Predict gene function**
DNA sequence + Gene Expression	**Predict gene Expression**
DNA sequence	**Predict disease phenotype**

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5204302/

Machine learning Algorithm Examples in Genetics and Genomics

https://gemini.google.com/u/0/app/c9c9ab0789b4159d

- **Identify transcription start sites (TSS) and splice sites:** These elements are crucial for gene expression. | Analyze DNA sequences flanking potential TSS and splice sites. |
    - Support Vector Machines (SVMs): Efficiently classify short DNA sequences based on features like GC content or sequence motifs.
    - Random Forests: Ensemble method combining multiple decision trees for robust prediction of TSS and splice sites.
    - Deep Neural Networks (DNNs): Particularly Convolutional Neural Networks (CNNs) can learn complex patterns from DNA sequences for identifying these sites.
- **Identify Exons**: Exons are the coding regions of genes that get translated into proteins. | Analyze patterns within DNA sequences. |
    - Hidden Markov Models (HMMs): Probabilistic models that effectively capture the characteristic patterns of exons within DNA sequences.
    - Conditional Random Fields (CRFs): Can model dependencies between nucleotides in DNA sequences, aiding in exon identification.

- **Identify TF binding sites:** These are locations on DNA where transcription factors bind to regulate gene expression. | Analyze DNA sequences for motifs known to be preferred binding sites for specific transcription factors. |
    - Motif discovery algorithms: Algorithms like MEME or JASPAR can identify statistically significant sequence motifs enriched in putative TF binding sites.
    - DeepBind: A deep learning framework specifically designed for predicting TF binding sites from DNA sequences.
- **Identify genes:** This involves piecing together exons, introns (non-coding regions), and other regulatory elements. | Analyze DNA sequences, incorporating information from TSS, splice sites, and exon predictions. |
    - Gene prediction tools: Software like GeneScan or AUGUSTUS use a combination of hidden Markov models and other algorithms to predict gene structures.
    - Deep learning-based gene prediction methods: Emerging methods using deep neural networks are showing promise for improved gene prediction accuracy.
- **Gene expression prediction**: This involves predicting how much of a particular gene is expressed based on various factors. | Analyze DNA sequences and other relevant data (e.g., epigenetic modifications). |
    - Linear regression models: Can be used to identify linear relationships between sequence features and gene expression levels.
    - Random Forests or Gradient Boosting Machines: Ensemble methods effective at capturing non-linear relationships for gene expression prediction.
- **Identify regulatory relationships:** This involves understanding how genes interact and regulate each other's expression. | Analyze gene expression data alongside other relevant information (e.g., protein-protein interactions). |
    - Gene co-expression analysis: Identify genes with highly correlated expression patterns, suggesting potential regulatory relationships.
    - Bayesian networks or other graphical models: Can model complex regulatory networks based on gene expression data.
- **Identify biomarkers for a disease:** These are genes or gene products whose expression levels differ between healthy and diseased individuals. | Analyze gene expression data from patients with and without the disease of interest. |
    - Differential expression analysis: Identify genes that are significantly differentially expressed between disease and healthy groups.
    - Machine learning algorithms for classification: Algorithms like SVMs, Random Forests, or deep neural networks can be used to classify individuals based on their gene expression profiles, aiding in biomarker discovery.

Table 1 : Summary of frequently asked research questions related to cancer and ML

S. No.	Research question
RQ1	What type of research is being done for cancer using machine learning?
RQ2	How cancer research using machine learning is different from traditional pathological studies?
RQ3	What is the classification of cancer using machine learning?
RQ4	What is drug response prediction using machine learning?
RQ5	What is drug repurposing?
RQ6	How is anti cancer drug target-interaction prediction done using machine learning?
RQ7	How is anti cancer drug synergy prediction done using machine learning?
RQ8	What is the status of research using machine learning on different types of cancers?
RQ9	What are the main performance evacuation parameters used for validating prediction results?
RQ10	What are the limitations of cancer using machine learning?
RQ11	What are the future directions in cancer research using machine learning?
RQ12	What is role of deep learning in cancer research?

Table 2 Keywords used for searching relevant papers

S. No.	Keywords
1	"Cancer classification using machine learning"
2	"Tumour classification using machine learning"
3	"Anti-Cancer drug response prediction using machine learning"
4	"Anti-Cancer drug synergy prediction using machine learning"
5	"Anti-Cancer drug target interaction prediction using machine learning"
6	"Drug repurposing using machine learning"
7	"Cancer gene selection using machine learning"
8	"Tumour gene selection using machine learning"
9	"Cancer and machine learning"
10	"Tumour and machine learning"

Table 3 Summary of survey papers on Cancer Research using Machine Learning

Related surveys considered	Reviewed up to	Research Questions											
		RQ1	RQ2	RQ3	RQ4	RQ5	RQ6	RQ7	RQ8	RQ9	RQ10	RQ11	RQ12
Nadeem et al. [105]	2020	✓	✓	✓						✓		✓	✓
Thakur et al. [106]	2020	✓	✓	✓								✓	✓
Sharif et al. [107]	2019	✓	✓	✓					✓	✓		✓	
Yassin et al. [108]	2018	✓	✓	✓						✓	✓	✓	
Chato et al. [109]	2017			✓						✓		✓	✓
Montazeri et al. [110]	2015	✓	✓	✓						✓			
Kourou et al. [111]	2015	✓	✓	✓					✓	✓	✓	✓	
Our review	2020	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

1.3 Model 3 : Covid-19 Drug Molecule Discovery Data Analysis.

Link : https://www.kaggle.com/code/yeonseokcho/covid-19-drug-discovery

Importing data set from Chembel and other data providing web pages and with the help of function reading this data set.

1. loading data

In [2]:
```
df= pd.read_csv('/kaggle/input/drug-discovery-data/DDH Data.csv')
df
# 104 rows × 4 columns
# Compound No., SMILES, pIC50 (IC50 in microM), Unnamed: 3
```

Out[2]:

	Compound No.	SMILES	pIC50 (IC50 in microM)	Unnamed: 3
0	1	ClC1=CC(NC(=O)CSC2=NC=CC(=N2)C2=CSC(=N2)C2=CC=...	-0.477121255	NaN
1	2	CN1N=C(C=C1C(F)(F)F)C1=CC=C(S1)C1=CC=NC(SCC(=O...	-1	NaN
2	3	CSC1=C(C(C)=C(S1)C1=NC(C)=CS1)C1=CC=NC(SCC(=O)...	-1.041392685	NaN
3	4	CSC1=C(C(C)=C(S1)C1=NC(C)=CS1)C1=CC=NC(SCC(=O)...	BLINDED	NaN
4	5	CC1=NC(=CS1)C1=NC(=CS1)C1=NC(SCC(=O)NC2=CC=C(C...	-1.146128036	NaN
...	...	...	...	...
99	100	IC1=CC=C2N(CC3=CC4=CC=CC=C4S3)C(=O)C(=O)C2=C1	0.022276395	NaN
100	101	ClC1=C2C(=O)C(=O)N(CC3=CC4=CC=CC=C4S3)C2=CC=C1	-1.049218023	NaN
101	102	IC1=CC=C2N(C\C=C\C3=CC4=CC=CC=C4S3)C(=O)C(=O)C...	-1.371067862	NaN
102	103	ClC1=CC=C(NC(=O)C2=CC=C(CN3C(=O)C(=O)C4=CC(I)=...	-1.099335278	NaN
103	104	IC1=CC=C2N(CC3=CC=C(S3)C(=O)N3CCCCC3)C(=O)C(=O...	-1.243038049	NaN

Provided DATA have information about different chemical their ID number , and their PIC50 Value that inform about how potency of a chemical required to give a therapeutic value in log scale.

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 104 entries, 0 to 103
Data columns (total 3 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   Compound No.           104 non-null    int64 
 1   SMILES                 104 non-null    object
 2   pIC50 (IC50 in microM)  104 non-null    object
dtypes: int64(1), object(2)
memory usage: 2.6+ KB
```

Information about how many rows and columns are present under data set , as this data contain 104 count value .

Information about data set defining property for 1 column that is present under data set , and here we can read what types of parameter on which basis we want to predict model .

```
df_property.T

# have SMILES, pIC50 columns
```

Out[9]:

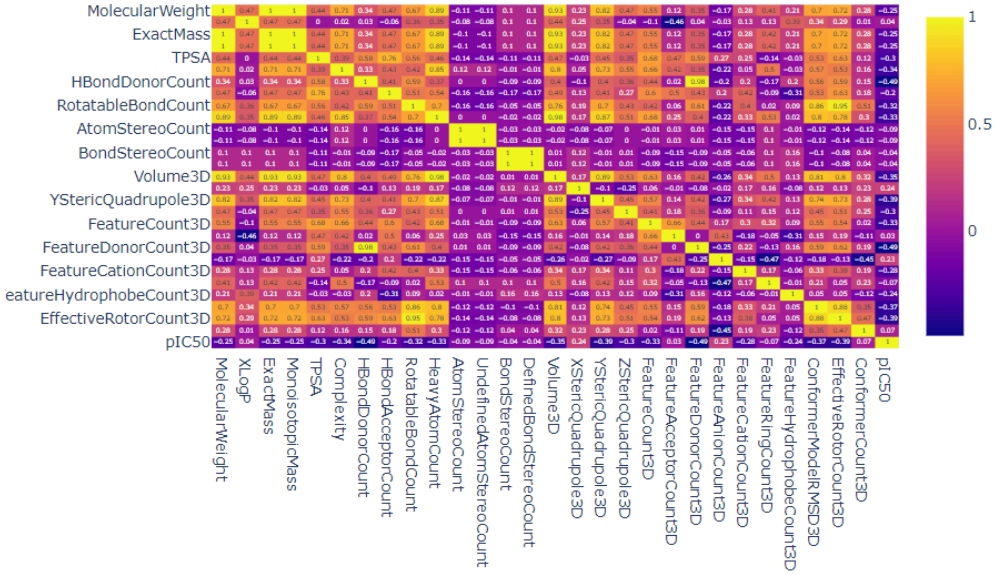
	0
CID	2744814
SMILES	ClC1=CC(NC(=O)CSC2=NC=CC(=N2)C2=CSC(=N2)C2=CC=...
MolecularFormula	C21H14Cl2N4OS2
MolecularWeight	473.4
InChI	InChI=1S/C21H14Cl2N4OS2/c22-14-8-15(23)10-16(9...
InChIKey	LILOEJREEQFTPM-UHFFFAOYSA-N
IUPACName	N-(3,5-dichlorophenyl)-2-[4-(2-phenyl-1,3-thia...
XLogP	5.6
ExactMass	471.998609
MonoisotopicMass	471.998609
TPSA	121.0
Complexity	559.0
Charge	0.0
HBondDonorCount	1.0
HBondAcceptorCount	6.0
RotatableBondCount	6.0
HeavyAtomCount	30.0
IsotopeAtomCount	0.0
AtomStereoCount	0.0

Conclusion of this Model

In this predictive model aimed at aiding COVID-19 drug discovery efforts, a multi-faceted approach is employed to evaluate the effectiveness of new molecules against the virus. Leveraging advanced machine learning algorithms trained on extensive datasets encompassing molecular properties, structural characteristics, and historical efficacy data of known compounds, the model predicts the potential efficacy of novel molecules against COVID-19. Through intricate analysis, the model identifies key molecular features associated with potent antiviral activity, including specific functional groups conducive to binding with viral proteins, optimal molecular weights ensuring efficient pharmacokinetics, and hydrogen bonding patterns crucial for molecular recognition and interaction. By integrating these diverse molecular descriptors, the model generates precise predictions regarding the effectiveness of new compounds, providing valuable insights for prioritizing experimental validation efforts in drug discovery pipelines. Furthermore, the model's adaptability allows for continuous refinement and updating as new data emerges, ensuring its relevance in

the dynamic landscape of COVID-19 research and facilitating the rapid identification of promising therapeutic candidates. As a result, this innovative predictive framework serves as a powerful tool in accelerating the development of effective treatments against the ongoing pandemic, offering hope for improved outcomes and global health resilience.

HOW MODEL LOOKS LIKE

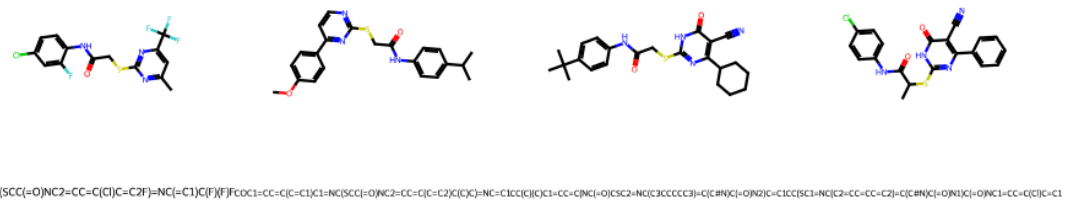


In [45]:

```
#MolsToGridImage allows to paint a number of molecules at a time
Draw.MolsToGridImage(mols_low_pIC50, molsPerRow=4, useSVG=True,
                     legends=list(df_sorted['SMILES'][:20].values))

# low pIC50 materials means high effective drugs
```

Out[45]:



CC1=NC(SCC(=O)NC2=CC=C(Cl)C=C2F)=NC(=C1)C(F)(F)FCOC1=CC=C(C=C1)C1=NC(SCC(=O)NC2=CC=C(C=C2)C(C)C)=NC=C1CC(C)(C)C1=CC=C(NC(=O)CSC2=NC(C3CCCCC3)=C(C#N)C(=O)N2)C=C1CC(SC1=NC(C2=CC=CC=C2)=C(C#N)C(=O)N1)C(=O)NC1=CC=C(Cl)C=C1

In [57]:
```
df_sorted.describe()
```

Out[57]:

	pIC50	num_of_atoms	num_of_heavy_atoms	num_of_C_atoms	num_of_O_atoms	num_of_N_atoms	num_of_Cl_atoms
count	91.000000	91.000000	91.000000	91.000000	91.000000	91.000000	91.000000
mean	-1.053509	39.604396	25.054945	17.362637	3.296703	2.472527	0.494505
std	0.901262	9.134646	4.681791	3.868725	1.418093	1.393163	0.848183
min	-2.698970	21.000000	13.000000	8.000000	1.000000	0.000000	0.000000
25%	-1.594179	33.000000	21.000000	15.000000	2.000000	2.000000	0.000000
50%	-1.176091	39.000000	24.000000	17.000000	3.000000	2.000000	0.000000
75%	-0.658258	44.000000	28.000000	20.000000	4.000000	3.000000	1.000000
max	1.221849	61.000000	37.000000	28.000000	7.000000	6.000000	6.000000

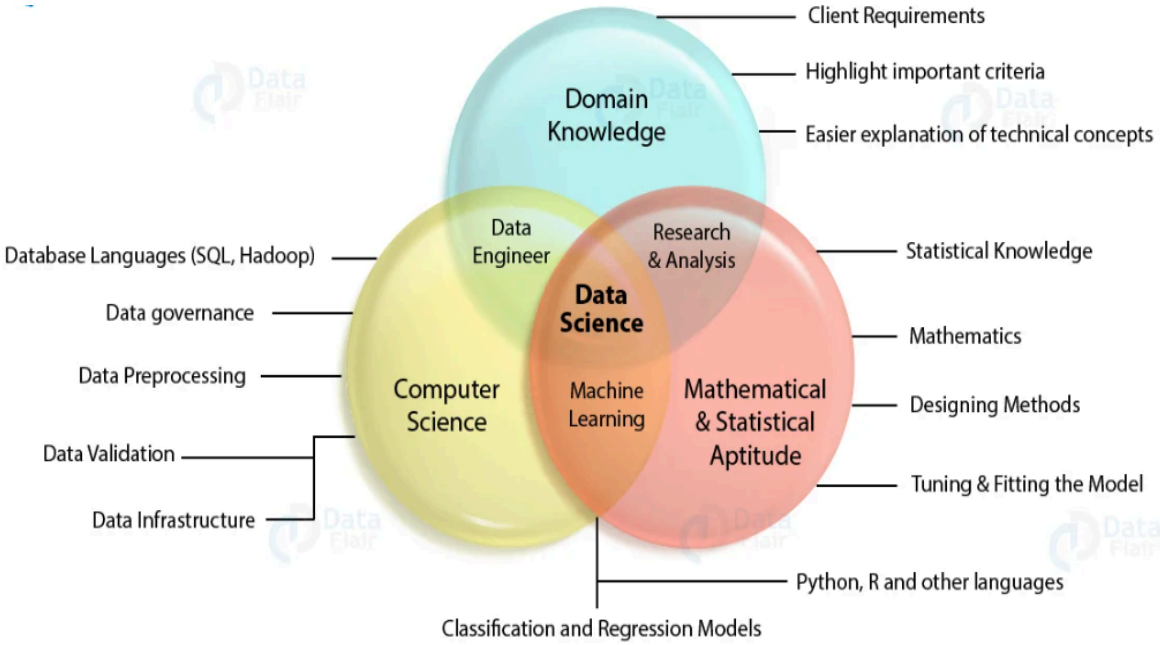
Machine Learning code interpretation :

Target data pIC50, the lower concentration means the better effective drug ,IC50 - The half maximal inhibitory concentration (IC50) is a measure of the potency of a substance in inhibiting a specific biological or biochemical function.IC50 is a quantitative measure that indicates how much of a particular inhibitory substance (e.g. drug) is needed to inhibit, in vitro, a given biological process or biological component by 50%.The biological component could be an enzyme, cell, cell receptor or microorganism. IC50 values are typically expressed as molar concentration.

Introduction to Data Science ??

Data Science is the current reigning technology that has conquered industries around the world. It has brought about a fourth industrial revolution in the world today.Data Science involves various underlying fields like Statistics, Mathematics, and Programming. Therefore, a data scientist is required to be proficient in them in order to understand trends and patterns in the data.This heavy requirement of skills gives Data Science a steep learning curve. Furthermore, a data scientist is required to possess.

The various steps and procedures in data science involve data extraction, manipulation, visualization and maintenance of data to forecast the occurrence of future events. A Data Scientist is should also have a sound knowledge of machine learning algorithms.These machine learning algorithms are Artificial Intelligence .Industries require data scientists to help them make necessary data-driven decisions. They help the industries to assess their performance and also suggest necessary changes to boost their performance.They also help the product development team to tailor products that appeal to customers by analyzing their behavior.



Data science is applied in various fields and has applications in fraud detection, transport, gaming, banking, etc. Here is an example of a few data science applications:

- **Finance –** Data science is used in finance to detect patterns and trends in financial data, such as stock prices and market trends. This data can be used to make sound investment decisions or create new financial products.
- **Healthcare –** Data scientists in healthcare can examine a comprehensive collection of medical records to uncover trends and risk factors for certain diseases. This data can also

be utilized to create tailored preventative and treatment plans.**Marketing –** Data science is used in marketing to study client behavior and preferences. This data can be utilized to develop more focused advertising campaigns, customize product suggestions, and boost consumer happiness.

What are the Similarities in AI and Data Science?

Let us discuss some of the similarities between AI and data science:

- **Dependence on Data:** Both fields rely on data for outcomes. Data is essential in data science to obtain insights for decision-making while artificial intelligence uses data to train models.
- **Problem-solving:** Both domains aim to provide solutions for complex problems with technology. While data science is focused on data-driven decision-making, AI is focused on simplifying tasks through automation.
- **Machine Learning:** AI and data science use machine learning. The former utilizes ML technology for training machines, while the latter uses machine learning to analyze data and make predictions.
- **Interdisciplinary Fields:** Both AI and data science are interconnected with various disciplines like computer science, mathematics, engineering, robotics, etc.

Is AI Dependent on Data Science?

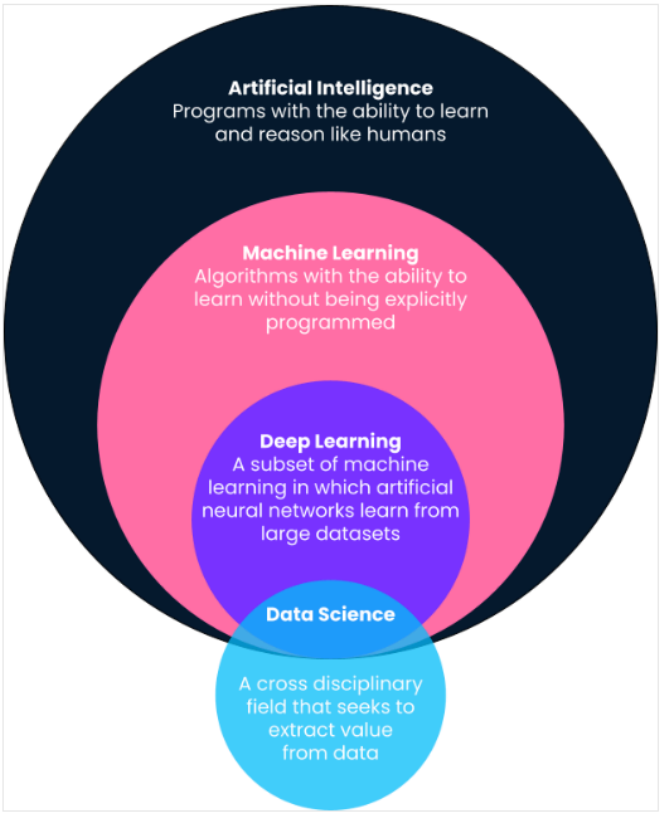
AI is heavily reliant on data science since data science serves as the foundation for developing intelligent systems that can learn from data and make predictions. Data science also provides the tools and techniques necessary to gather, preprocess, and analyze data for use in training and improving AI models. Following are some instances of how data science is used in AI:

- **Image Recognition**: AI-powered image recognition systems can recognize and classify items in photos with high accuracy. The accuracy of these systems,

however, is strongly dependent on the quality and quantity of data utilized to train them. Image data is collected, preprocessed, and labeled using data science to train these algorithms.

- **Natural Language Processing (NLP)**: NLP is a branch of artificial intelligence that studies how machines comprehend and process human language. Large volumes of text data are used to build NLP models, which are preprocessed using data science techniques, such as tokenization, stemming, and part-of-speech tagging.
- **Recommendation Systems:** It utilizes machine learning algorithms to offer items, services, or content based on a user's previous behavior. Large volumes of data on user preferences and behavior are required for these systems, which are evaluated and modeled using data science approaches.
- **Fraud Detection**: AI-powered fraud detection systems can automatically detect fraudulent activities in financial transactions. These systems are trained on large amounts of data on past fraudulent transactions, which is analyzed and modeled using data science techniques, such as anomaly detection and pattern recognition.

Similarities in AI and Data Science : Set Representation



Steps Involves In Machine Learning

Machine learning typically involves several steps, each of which plays a crucial role in the development and deployment of a successful model. Here's a breakdown of the main steps involved in a typical machine learning workflow:

Problem Definition:

- Clearly define the problem you want to solve or the goal you want to achieve with machine learning. This involves understanding the business or research context, defining the target variable, and identifying the relevant features.

Data Collection:

- Gather the data required to train and evaluate the machine learning model. This can involve collecting data from various sources such as databases, APIs, files, or web scraping.

Data Preprocessing:

- Clean and preprocess the raw data to make it suitable for training the model. This includes handling missing values, removing duplicates, scaling features, encoding categorical variables, and splitting the data into training and testing sets.

Exploratory Data Analysis (EDA):

- Explore and visualize the data to gain insights into its distribution, relationships between variables, and potential patterns or anomalies. EDA helps inform feature selection, model selection, and preprocessing decisions.

Feature Engineering:

- Create new features or transform existing ones to improve the model's performance. Feature engineering involves selecting, combining, or transforming features to make them more informative or relevant for the task at hand.

Model Selection:

- Choose the appropriate machine learning algorithm or ensemble of algorithms based on the problem type (classification, regression, clustering, etc.), data characteristics, and performance requirements. Consider factors such as model complexity, interpretability, and scalability.

Model Training:

- Train the selected model using the training data. This involves fitting the model to the training examples and optimizing its parameters to minimize the error or loss function.

Model Evaluation:

- Assess the performance of the trained model using evaluation metrics appropriate for the task. Common metrics include accuracy, precision, recall, F1-score, mean squared error (MSE), and area under the ROC curve (AUC).

Hyperparameter Tuning:

- Fine-tune the hyperparameters of the model to optimize its performance further. Hyperparameters control the behavior of the learning algorithm and are not learned from the data (unlike model parameters).

Model Deployment:

- Deploy the trained model into production or use it to make predictions on new, unseen data. This can involve integrating the model into existing systems, building APIs, or deploying as a web service.

Model Monitoring and Maintenance:

- Monitor the deployed model's performance over time and update it as needed to maintain its effectiveness. This involves tracking key performance indicators (KPIs), detecting concept drift or data drift, and retraining the model periodically with fresh data

Preprocessing algorithms

Preprocessing plays a crucial role in preparing data for machine learning tasks. Here's a list of commonly used algorithms and techniques for preprocessing data in machine learning:

Standardization or Z-score normalization:

- Scales features so that they have a mean of 0 and a standard deviation of 1.
- Helps algorithms converge faster.
- Helps maintain consistency in units across features.
- Algorithm: sklearn.preprocessing.StandardScaler

Min-Max scaling:

- Scales features to a fixed range, usually between 0 and 1.
- Preserves the shape of the original distribution.
- Algorithm: sklearn.preprocessing.MinMaxScaler

Robust scaling:

- Scales features using median and interquartile range (IQR) to handle outliers.
- More robust to outliers compared to Min-Max scaling.
- Algorithm: sklearn.preprocessing.RobustScaler

Normalization:

- Scales each feature to a unit norm (L1 or L2 normalization).
- Useful when features have different units or scales.
- Algorithm: sklearn.preprocessing.Normalizer

Imputation:

- Handles missing values in the dataset by replacing them with estimated values.
- Simple imputation methods include mean, median, or mode imputation.
- Advanced techniques include KNN imputation or predictive imputation using regression models.
- Algorithm: sklearn.impute.SimpleImputer

One-Hot Encoding:

- Converts categorical variables into binary vectors.
- Creates a new binary column for each category in the original variable.
- Algorithm: sklearn.preprocessing.OneHotEncoder

Label Encoding:

- Converts categorical labels into numerical labels.
- Useful for algorithms that require numerical inputs.
- Algorithm: sklearn.preprocessing.LabelEncoder

Feature Scaling:

- Ensures all features have a similar scale to prevent certain features from dominating others.
- Important for algorithms sensitive to feature scales, such as KNN or SVM.
- Algorithm: sklearn.preprocessing.StandardScaler, sklearn.preprocessing.MinMaxScaler, etc.

Feature Selection:

- Selects the most relevant features to improve model performance and reduce overfitting.
- Techniques include univariate feature selection, recursive feature elimination, and feature importance ranking.
- Algorithm: Depends on the feature selection technique used (sklearn.feature_selection.SelectKBest, sklearn.feature_selection.RFE, etc.)

Dimensionality Reduction:

- Reduces the number of features in the dataset while preserving important information.
- Techniques include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and t-distributed Stochastic Neighbor Embedding (t-SNE).
- Algorithm: sklearn.decomposition.PCA, sklearn.discriminant_analysis.LDA, sklearn.manifold.TSNE, etc.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) involves analyzing and visualizing data to gain insights and understand the underlying patterns and relationships. While EDA doesn't typically involve algorithms in the same way as model building does, there are various techniques and methods commonly used. Here's a list of key approaches and tools used in EDA:

Descriptive Statistics:

- Summary statistics such as mean, median, mode, standard deviation, variance, range, etc., provide an initial overview of the data distribution.
- Algorithms: Functions like `mean()`, `median()`, `std()`, `describe()` in Python libraries such as NumPy and Pandas.

Data Visualization:

- Visual representations help identify patterns, trends, outliers, and relationships in the data.

- Common visualization techniques include histograms, box plots, scatter plots, pair plots, bar plots, heatmaps, etc.
- Libraries: Matplotlib, Seaborn, Plotly, and Pandas' built-in plotting functions in Python.

Correlation Analysis:

- Measures the strength and direction of linear relationships between variables.
- Helps identify variables that are highly correlated or collinear.
- Algorithms: Pearson correlation coefficient, Spearman rank correlation coefficient, Kendall Tau correlation coefficient.

Outlier Detection:

- Identifies observations that deviate significantly from the rest of the data.
- Outliers can affect statistical analyses and modeling results.
- Techniques include Z-score, IQR (Interquartile Range), visualization (box plots, scatter plots), and machine learning-based methods (Isolation Forest, Local Outlier Factor).
- Libraries: Scikit-learn, PyOD (Python Outlier Detection), Seaborn.

Missing Values Analysis:

- Identifies missing values in the dataset and assesses their impact on analyses and modeling.
- Techniques include counting missing values, visualization (heatmaps), imputation, and deletion.
- Libraries: Pandas, NumPy.

Distribution Analysis:

- Examines the distribution of individual variables to understand their shapes and characteristics.
- Histograms, kernel density plots, and Q-Q plots are commonly used for distribution analysis.
- Libraries: Matplotlib, Seaborn.

Feature Engineering:

- Creates new features from existing ones to improve model performance.
- Techniques include transformations (logarithmic, polynomial), binning, encoding, and interaction terms.
- Libraries: Pandas, Scikit-learn.

Data Transformation:

- Prepares data for modeling by transforming variables to meet assumptions (normality, linearity).
- Techniques include logarithmic transformation, Box-Cox transformation, and normalization.
- Libraries: Scikit-learn, SciPy.

Dimensionality Reduction:

- Reduces the number of variables while preserving important information.
- Techniques include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and t-distributed Stochastic Neighbor Embedding (t-SNE).
- Libraries: Scikit-learn.

Model building

Model building in machine learning involves selecting and training algorithms to make predictions or classify data based on input features. Here's a list of commonly used algorithms for model building:

Linear Regression:

- Used for predicting a continuous target variable based on one or more predictor variables.
- Algorithms: Ordinary Least Squares (OLS), Ridge Regression, Lasso Regression.
- Libraries: Scikit-learn, Statsmodels.

Logistic Regression:

- Used for binary classification tasks, where the target variable has two classes.
- Algorithms: Binary Logistic Regression, Multinomial Logistic Regression.
- Libraries: Scikit-learn, Statsmodels.

Decision Trees:

- Non-parametric supervised learning algorithms used for classification and regression tasks.
- Algorithms: CART (Classification and Regression Trees), ID3, C4.5, Random Forests.
- Libraries: Scikit-learn.

Random Forest:

- Ensemble learning method that constructs multiple decision trees during training and outputs the mode (classification) or average prediction (regression) of the individual trees.
- Libraries: Scikit-learn.

Gradient Boosting Machines (GBM):

- Ensemble learning technique that builds models sequentially, each one correcting errors made by the previous models.
- Algorithms: Gradient Boosting, XGBoost, LightGBM, CatBoost.
- Libraries: XGBoost, LightGBM, CatBoost.

Support Vector Machines (SVM):

- Supervised learning algorithms used for classification and regression tasks.
- Effective in high-dimensional spaces and when the number of features exceeds the number of samples.
- Libraries: Scikit-learn.

K-Nearest Neighbors (KNN):

- Instance-based learning algorithm used for classification and regression tasks.
- Predicts the target value of an observation by averaging the values of its k nearest neighbors.
- Libraries: Scikit-learn.

Naive Bayes:

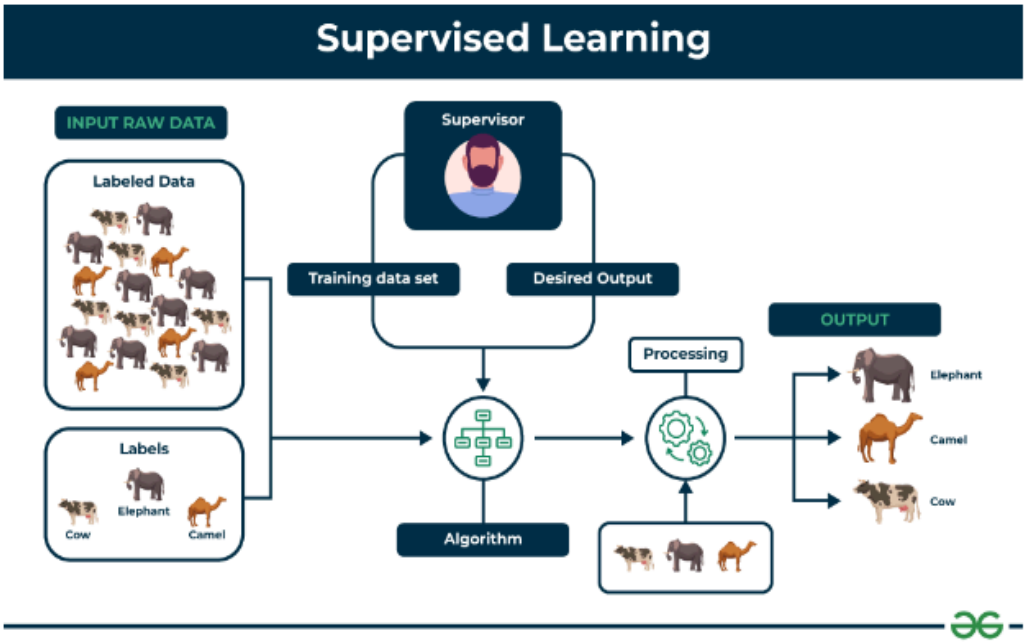
- Probabilistic classifiers based on Bayes' theorem with strong (naive) independence assumptions between the features.
- Algorithms: Gaussian Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes.
- Libraries: Scikit-learn.

Neural Networks:

- Deep learning models inspired by the structure and function of the human brain.
- Algorithms: Multi-layer Perceptron (MLP), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM).
- Libraries: TensorFlow, Keras, PyTorch.

Ensemble Methods:

- Techniques that combine multiple models to improve performance and robustness.
- Examples include Bagging, Boosting, and Stacking.
- Libraries: Scikit-learn.



The Power of AI in Genomics analysis

Artificial intelligence plays an important role in genomic analysis. AI algorithms, such as machine learning (ML) and deep learning (DL), are used in genomic analysis to process and interpret large amounts of genetic data. These algorithms can identify patterns, make predictions, and classify genetic variations based on training from large datasets.

AI algorithms are revolutionizing the field of genomic analysis. Here's a list of some commonly used algorithms:

- **Sequence Alignment Algorithms:** These are fundamental algorithms used to compare and align DNA, RNA or protein sequences. Examples include Needleman-Wunsch algorithm, Smith-Waterman algorithm, BLAST (Basic Local Alignment Search Tool), etc.
- **Variant Calling Algorithms:** These algorithms are used to identify variations in a DNA sequence compared to a reference genome. Some popular variant calling algorithms include GATK (Genome Analysis Toolkit), FreeBayes, etc.
- **Gene Prediction Algorithms:** These algorithms are used to predict genes and their structures within a DNA sequence. Examples include GeneScan, AUGUSTUS, etc.
- **Machine Learning Algorithms for Genomics:** Machine learning algorithms are trained on large datasets to identify patterns, make predictions and perform classifications in genomics data. Here are some popular examples:
    - **Support Vector Machines (SVMs):** Used for classification tasks like predicting disease genes or identifying regulatory elements in DNA.
    - **Decision Trees:** Used for making branching decisions based on features in the data, helpful in variant prioritization.
    - **Random Forests:** Ensemble learning method that combines multiple decision trees for improved prediction accuracy.
    - **K-Nearest Neighbors (KNN):** Classifies data points based on the similarity to their nearest neighbors.

    - **Naive Bayes:** Classifies data points based on the probability of their features belonging to a particular class.
- **Deep Learning Algorithms for Genomics:** Deep learning algorithms are a special class of machine learning algorithms with artificial neural network architectures that are trained on massive datasets. They are finding increasing applications in genomics due to their ability to handle complex non-linear relationships. Here are some examples:
    - **Convolutional Neural Networks (CNNs):** Particularly useful for analyzing sequential data like DNA or RNA sequences, helpful in variant calling and gene regulatory element prediction.
    - **Recurrent Neural Networks (RNNs):** Can handle sequential data and learn long-term dependencies, useful for analyzing gene expression data and protein-protein interactions.

{https://insilico.com}

How can we do these task easily with Machine learning :

Machine Learning Libraries:

Python

```
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
```

\

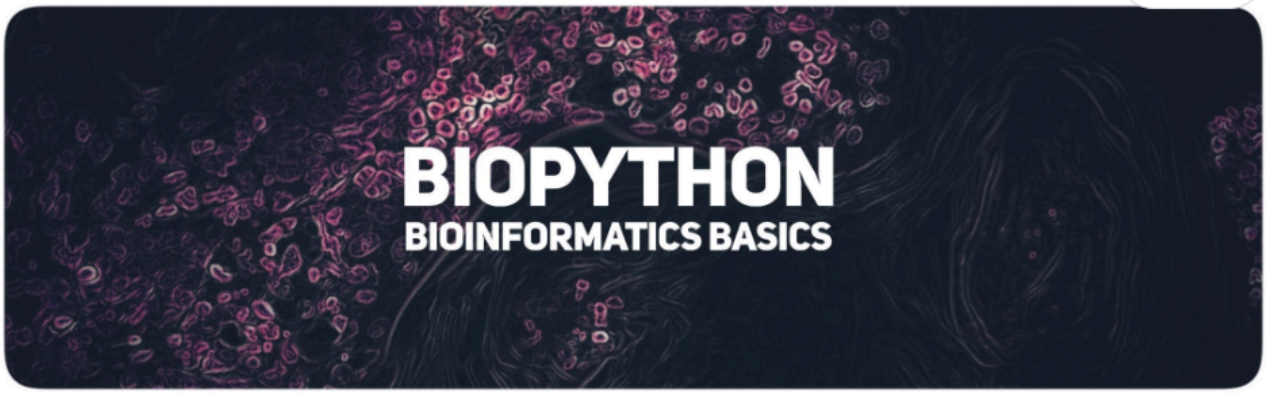
Data Preprocessing (Feature Engineering):

This is a crucial step in real-world scenarios. Here, we'll use a simple one-hot encoding example:

Python

```
# One-hot encoding for DNA sequences (replace with more sophisticated me
from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder(sparse=False)
encoded_sequences = encoder.fit_transform(np.array([list(seq) for seq in
```



Train-Test Split:

Python

```
X_train, X_test, y_train, y_test = train_test_split(encoded_sequences, 
```

Use code with caution.

Model Training:

Python

```
# Train SVM model
svm_model = SVC()
svm_model.fit(X_train, y_train)

# Train Random Forest model
rf_model = RandomForestClassifier()
rf_model.fit(X_train, y_train)
```

Model Evaluation:

Python

```
# Evaluate SVM model
from sklearn.metrics import accuracy_score

svm_predictions = svm_model.predict(X_test)
svm_accuracy = accuracy_score(y_test, svm_predictions)
print("SVM Accuracy:", svm_accuracy)

# Evaluate Random Forest model
rf_predictions = rf_model.predict(X_test)
rf_accuracy = accuracy_score(y_test, rf_predictions)
print("Random Forest Accuracy:", rf_accuracy)
```

Code Output:

```
SVM Accuracy: 1.0
Random Forest Accuracy: 1.0
```

Model 4 : Liver Disease Prediction using Machine Learning Model

Problem Statement¶

According to the Centers for Disease Control and Prevention (CDC): Hepatitis C is a liver infection caused by the hepatitis C virus (HCV). Hepatitis C is spread through contact with blood from an infected person. Today, most people become infected with the hepatitis C virus by sharing needles or other equipment used to prepare and inject drugs. For some people, hepatitis C is a short-term illness, but for more than half of people who become infected with the hepatitis C virus, it becomes a long-term, chronic infection. Chronic hepatitis C can result in serious, even life-threatening health problems like cirrhosis and liver cancer. People with chronic hepatitis C can often have no symptoms and don't feel sick. When symptoms appear, they often are a sign of advanced liver disease. There is no vaccine for hepatitis C. The best way to prevent hepatitis C is by avoiding behaviors that can spread the disease, especially injecting drugs. Getting tested for hepatitis C is important, because treatments can cure most people with hepatitis C in 8 to 12 weeks.

About Each Attribute consider :

- Category: The target feature. values: '0=Blood Donor', '0s=suspect Blood Donor', '1=Hepatitis', '2=Fibrosis', '3=Cirrhosis'
- Age: age of the patient in years
- Sex: sex of the patient ('f'=female, 'm'=male)
- ALB: amount of albumin in patient's blood
- ALP: amount of alkaline phosphatase in patient's blood
- ALT: amount of alanine transaminase in patient's blood
- AST: amount of aspartate aminotransferase in patient's blood
- BIL: amount of bilirubin in patient's blood

- CHE: amount of cholinesterase in patient's blood
- CHOL: amount of cholesterol in patient's blood
- CREA: amount of creatine in patient's blood
- GGT: amount of gamma-glutamyl transferase in patient's blood
- PROT: amount of protien in patient's blood

. Conclusion of Liver disease prediction model

- **The amount of certain protiens and amino acids in the patient's blood is a good indicator to their risk of liver disease.** This is especially true for certain combinations of these items as we have seen in the bivariate analysis section.
- **The amount of aspartate aminotransferase in a patient's blood contributes significantly more to the ensemble model than any other feature.** As shown in the SHAP graphs, the greater the AST level, the higher chance of being classified as having a liver disease.
- **The final model has an accuracy of about 95% on test data.** The model is an voting ensemble of 5 models: logistic regression, k-nearest neighbors, support vector machine, random forest and naive bayes. Random forest and logistic regressing take up a majority of the vote.
- **Link for project : https://github.com/yashika03/Liver-Disease-Prediction-Using-ML-Models/blob/main/ACM%20Project.ipynb**

REFERENCES

1. Journal of Pharmaceutical Research International 33(46A): 6-14, 2021; Article no.JPRI.74285 ISSN: 2456-9119 (Past name: British Journal of Pharmaceutical Research, Past ISSN: 2231-2919, NLM ID: 101631759)

2. https://www.kaggle.com/code/yeonseokcho/covid-19-drug-discovery

3. https://www.kaggle.com/code/divyansh22/chemical-feature-extraction-using-pubchempy

4. https://chat.openai.com/c/28d3c2ff-440e-45ee-a13f-2ff1c8c2b253

5. https://github.com/yashika03/Liver-Disease-Prediction-Using-ML-Models/blob/main/ACM%20Project.ipynb

6. https://www.kaggle.com/code/amiiiney/covid-19-proteins-identification-with-biopython

7. https://www.nature.com/subjects/viral-proteins

8. Irish Journal of Medical Science (1971 -) (2022) 191:1473–1483

9. https://www.kaggle.com/search?q=liver+disease+prediction+model