.کتابخانه‌های Machine Learning در زبان Rust را نام ببرید؟ یک مثال ساده بنویسید؟

کتابخانه‌های محبوب Machine Learning در Rust:

- Linfa: برای الگوریتم‌های یادگیری ماشین عمومی.
- smartcore: برای یادگیری ماشین کلاسیک.
- tract: برای کار با مدل‌های یادگیری عمیق.

مثال ساده با Linfa:

```rust
use linfa::prelude::*;
use linfa_clustering::KMeans;

fn main() {
    let data = ndarray::array![[1.0, 2.0], [1.5, 1.8], [5.0, 8.0], [8.0, 8.0]];
    let model = KMeans::params(2).fit(&data).unwrap();
    let labels = model.predict(&data);
    println!("Cluster labels: {:?}", labels);
}
```

.2برنامه‌نویسی Multi-Threading در زبان Rust را با ذکر یک مثال ساده توضیح دهید؟

در Rust از کتابخانه استاندارد std::thread برای Multi-Threading استفاده می‌شود.

```rust
use std::thread;

fn main() {
    let handles: Vec<_> = (0..5)
        .map(|i| thread::spawn(move || {
            println!("Thread {} is running", i);
        }))
        .collect();

    for handle in handles {
        handle.join().unwrap();
    }
}
```

3.برنامه‌نویسی Parallel Programming در زبان Rust را با ذکر یک مثال ساده توضیح دهید؟

برای Parallel Programming می‌توان از کتابخانه‌هایی مانند rayonاستفاده کرد.

```rust
use rayon::prelude::*;

fn main() {
    let numbers: Vec<i32> = (1..=100).collect();
    let sum: i32 = numbers.par_iter().map(|x| x * x).sum();
    println!("Sum of squares: {}", sum);
}
```

4. Lazy Loadingچیست؟ با ذکر مثال در زبان Rust توضیح دهید؟

Lazy Loadingبه بارگذاری داده‌ها یا اجرای محاسبات تنها در صورت نیاز اشاره دارد.

مثال

```rust
use once_cell::sync::Lazy;

static CONFIG: Lazy<String> = Lazy::new(|| {
    println!("Initializing...");
    "This is lazy loaded!".to_string()
});

fn main() {
    println!("Before accessing CONFIG");
    println!("CONFIG: {}", *CONFIG);
}
```

5.ساختمان داده Binary Search Tree را در زبان Rust پیاده‌سازی نمایید؟

یک پیاده‌سازی ساده:Binary Search Tree

```rust
#[derive(Debug)]
struct Node {
    value: i32,
    left: Option<Box<Node>>,
    right: Option<Box<Node>>,
}

impl Node {
    fn new(value: i32) -> Self {
        Node {
            value,
            left: None,
            right: None,
        }
    }

    fn insert(&mut self, value: i32) {
        if value < self.value {
            if let Some(ref mut left) = self.left {
                left.insert(value);
            } else {
                self.left = Some(Box::new(Node::new(value)));
            }
        } else {
            if let Some(ref mut right) = self.right {
                right.insert(value);
            } else {
                self.right = Some(Box::new(Node::new(value)));
            }
        }
    }
}

fn main() {
    let mut root = Node::new(10);
    root.insert(5);
    root.insert(15);
    println!("{:#?}", root);
}
```

6.ساختمان داده AVL Tree را در زبان Rust پیاده‌سازی نمایید؟

درAVL Tree ، بالانس بودن ارتفاع زیر درخت‌ها تضمین می‌شود. پیاده‌سازی آن کمی پیچیده‌تر از Binary Search Tree است.
نمونه ساده‌ای در زیر ارائه شده است:

```
#[derive(Debug)]
struct Node {
   value: i32,
   height: i32,
   left: Option<Box<Node>>,
   right: Option<Box<Node>>,
}


impl Node {
   fn new(value: i32) -> Self {
      Node {
         value,
         height: 1,
         left: None,
         right: None,
      }
   }

   fn height(node: &Option<Box<Node>>) -> i32 {
      node.as_ref().map_or(0, |n| n.height)
   }

   fn balance_factor(node: &Option<Box<Node>>) -> i32 {
      Node::height(&node.as_ref().unwrap().left) - Node::height(&node.as_ref().unwrap().right)
   }
```

```
fn rotate_right(mut y: Box<Node>) -> Box<Node> {
    let mut x = y.left.take().unwrap();
    y.left = x.right.take();
    y.height = 1 + i32::max(Node::height(&y.left), Node::height(&y.right));
    x.right = Some(y);
    x.height = 1 + i32::max(Node::height(&x.left), Node::height(&x.right));
    x
}

fn rotate_left(mut x: Box<Node>) -> Box<Node> {
    let mut y = x.right.take().unwrap();
    x.right = y.left.take();
    x.height = 1 + i32::max(Node::height(&x.left), Node::height(&x.right));
    y.left = Some(x);
    y.height = 1 + i32::max(Node::height(&y.left), Node::height(&y.right));
    y
}

fn balance(mut node: Box<Node>) -> Box<Node> {
    node.height = 1 + i32::max(Node::height(&node.left), Node::height(&node.right));
    let balance = Node::balance_factor(&Some(node.clone()));

    if balance > 1 {
        if Node::balance_factor(&node.left) < 0 {
            node.left = Some(Node::rotate_left(node.left.take().unwrap()));
        }
        return Node::rotate_right(node);
    }
    if balance < -1 {
```

```
            if Node::balance_factor(&node.right) > 0 {
                node.right = Some(Node::rotate_right(node.right.take().unwrap()));
            }
            return Node::rotate_left(node);
        }
        node
    }

    fn insert(node: Option<Box<Node>>, value: i32) -> Box<Node> {
        if let Some(mut current) = node {
            if value < current.value {
                current.left = Some(Node::insert(current.left.take(), value));
            } else {
                current.right = Some(Node::insert(current.right.take(), value));
            }
            Node::balance(current)
        } else {
            Box::new(Node::new(value))
        }
    }
}

fn main() {
    let mut root = None;
    root = Some(Node::insert(root, 10));
    root = Some(Node::insert(root, 20));
    root = Some(Node::insert(root, 5));
    root = Some(Node::insert(root, 4));
    println!("{:#?}", root);
```

```
}
```

ساختمان داده Max Heap-Tree را در زبان Rust پیاده‌سازی نمایید؟.

```
#[derive(Debug)]
struct MaxHeap {
    data: Vec<i32>,
}

impl MaxHeap {
    fn new() -> Self {
        MaxHeap { data: Vec::new() }
    }

    fn insert(&mut self, value: i32) {
        self.data.push(value);
        let mut idx = self.data.len() - 1;

        while idx > 0 {
            let parent = (idx - 1) / 2;
            if self.data[idx] > self.data[parent] {
                self.data.swap(idx, parent);
                idx = parent;
            } else {
                break;
            }
        }
    }

    fn extract_max(&mut self) -> Option<i32> {
```

```
        if self.data.is_empty() {
            return None;
        }
        let max = self.data.swap_remove(0);
        self.heapify(0);
        Some(max)
    }

    fn heapify(&mut self, idx: usize) {
        let left = 2 * idx + 1;
        let right = 2 * idx + 2;
        let mut largest = idx;

        if left < self.data.len() && self.data[left] > self.data[largest] {
            largest = left;
        }
        if right < self.data.len() && self.data[right] > self.data[largest] {
            largest = right;
        }
        if largest != idx {
            self.data.swap(idx, largest);
            self.heapify(largest);
        }
    }
}

fn main() {
    let mut heap = MaxHeap::new();
    heap.insert(10);
```

```
    heap.insert(20);
    heap.insert(5);
    println!("{:?}", heap);
    println!("Max: {:?}", heap.extract_max());
    println!("{:?}", heap);
}
```

8.یک سرویس RESTful API جهت پردازش درخواست‌های JSON بنویسید؟

برای این کار می‌توان از کتابخانه actix-webاستفاده کرد:

```rust
use actix_web::{web, App, HttpServer, Responder};

#[derive(serde::Deserialize, serde::Serialize)]
struct MyData {
    name: String,
    age: u32,
}

async fn process_json(data: web::Json<MyData>) -> impl Responder {
    format!("Hello {}, you are {} years old!", data.name, data.age)
}

#[actix_web::main]
async fn main() -> std::io::Result<()> {
    HttpServer::new(|| {
        App::new()
            .route("/process", web::post().to(process_json))
    })
    .bind("127.0.0.1:8080")?
    .run()
    .await
}
```

9. **یک سرویس ساده جهت پردازش درخواست‌های مبتنی بر پروتکل gRPC بنویسید؟**

مراحل ایجاد یک سرویس gRPC درRust

1. **ایجاد فایل proto**

ابتدا باید فایل proto.ایجاد کنید که تعریف‌های gRPC در آن نوشته می‌شود.

service.proto:

```
syntax = "proto3";

package my_service;

service MyService {
    rpc SayHello (HelloRequest) returns (HelloResponse);
}

message HelloRequest {
    string name = 1;
}

message HelloResponse {
    string message = 1;
}
```

10. یک سرویس ساده جهت پردازش درخواست‌های مبتنی بر Web Assembly بنویسید؟

برای ساخت برنامه‌های WebAssembly در Rust می‌توان از کتابخانه wasm-bindgenاستفاده کرد.

```rust
use wasm_bindgen::prelude::*;

#[wasm_bindgen]
pub fn greet(name: &str) -> String {
    format!("Hello, {}! Welcome to WebAssembly.", name)
}
```

Socket Programming .در زبان Rust را به همراه یک مثال بیان کنید؟

مثال ساده TCP Server :و:Client

Server:

```rust
use std::io::prelude::*;
use std::net::TcpListener;

fn main() {
    let listener = TcpListener::bind("127.0.0.1:8080").unwrap();
    println!("Server running on port 8080");

    for stream in listener.incoming() {
        let mut stream = stream.unwrap();
        let mut buffer = [0; 1024];
        stream.read(&mut buffer).unwrap();
        println!("Received: {}", String::from_utf8_lossy(&buffer));
        stream.write(b"Hello from server!").unwrap();
    }
}
```

Client

```rust
use std::io::prelude::*;
use std::net::TcpStream;

fn main() {
    let mut stream = TcpStream::connect("127.0.0.1:8080").unwrap();
    stream.write(b"Hello server!").unwrap();

    let mut buffer = [0; 1024];
    stream.read(&mut buffer).unwrap();
    println!("Received: {}", String::from_utf8_lossy(&buffer));
}
```

12.برنامه‌ای برای عملیات CRUD بر روی پایگاه داده در Rust بنویسید؟

برای انجام عملیات CRUD از کتابخانه sqlxاستفاده می‌کنیم:

```
use sqlx::mysql::MySqlPool;


#[tokio::main]
async fn main() -> Result<(), sqlx::Error> {
   let pool = MySqlPool::connect("mysql://user:password@localhost/database").await?;


   // Create
   sqlx::query("INSERT INTO users (name, age) VALUES (?, ?)")
      .bind("Alice")
      .bind(30)
      .execute(&pool)
      .await?;


   // Read
   let row: (String, i32) = sqlx::query_as("SELECT name, age FROM users WHERE name = ?")
      .bind("Alice")
```

```
        .fetch_one(&pool)
        .await?;
    println!("User: {} is {} years old", row.0, row.1);

    // Update
    sqlx::query("UPDATE users SET age = ? WHERE name = ?")
        .bind(31)
        .bind("Alice")
        .execute(&pool)
        .await?;

    // Delete
    sqlx::query("DELETE FROM users WHERE name = ?")
        .bind("Alice")
        .execute(&pool)
        .await?;

    Ok(())
}
```

13.برنامه‌ای برای خواندن و نوشتن یک فایل CSV ساده بنویسید؟

با استفاده از کتابخانه :csv

```rust
use csv::Writer;

fn main() -> Result<(), Box<dyn std::error::Error>> {
    let mut wtr = Writer::from_path("data.csv")?;
    wtr.write_record(&["name", "age"])?;
    wtr.write_record(&["Alice", "30"])?;
    wtr.write_record(&["Bob", "25"])?;
    wtr.flush()?;

    let mut rdr = csv::Reader::from_path("data.csv")?;
    for result in rdr.records() {
        let record = result?;
        println!("{:?}", record);
    }

    Ok(())
}
```

14.مدل MVC را در قالب یک برنامه پیاده‌سازی کنید؟

مدل MVC (Model-View-Controller) به صورت مفهومی اجرا می‌شود. در Rust می‌توان از فریمورک‌هایی مثل Actix-web برای این ساختار استفاده کرد.

ساختار پیشنهادی:

- **Model:** مسئول داده‌ها (ارتباط با پایگاه داده).
- **View:** مسئول ارائه پاسخ‌ها.(HTML/JSON)
- **Controller:** مدیریت درخواست‌ها.

```rust
// Model
struct User {
    name: String,
    age: u32,
}

// Controller
async fn get_user() -> impl Responder {
    let user = User {
        name: "Alice".to_string(),
        age: 30,
    };
    web::Json(user)
}

// View
#[actix_web::main]
async fn main() -> std::io::Result<()> {
    HttpServer::new(|| {
        App::new().route("/user", web::get().to(get_user))
    })
    .bind("127.0.0.1:8080")?
    .run()
    .await
}
```

15.اصول SOLID را در زبان Rust پیاده‌سازی نمایید؟

اصول SOLID شامل موارد زیر است:

1. **Single Responsibility Principle (SRP):** هر ماژول باید تنها یک مسئولیت داشته باشد.

```rust
struct FileLogger;
impl FileLogger {
    fn log(&self, message: &str) {
        println!("Log to file: {}", message);
    }
}
```