

Anvendelse af AI i PORTMAN

Hvem er vi?



Morten Mackenhauer

Software Architect



Thomas Saaby

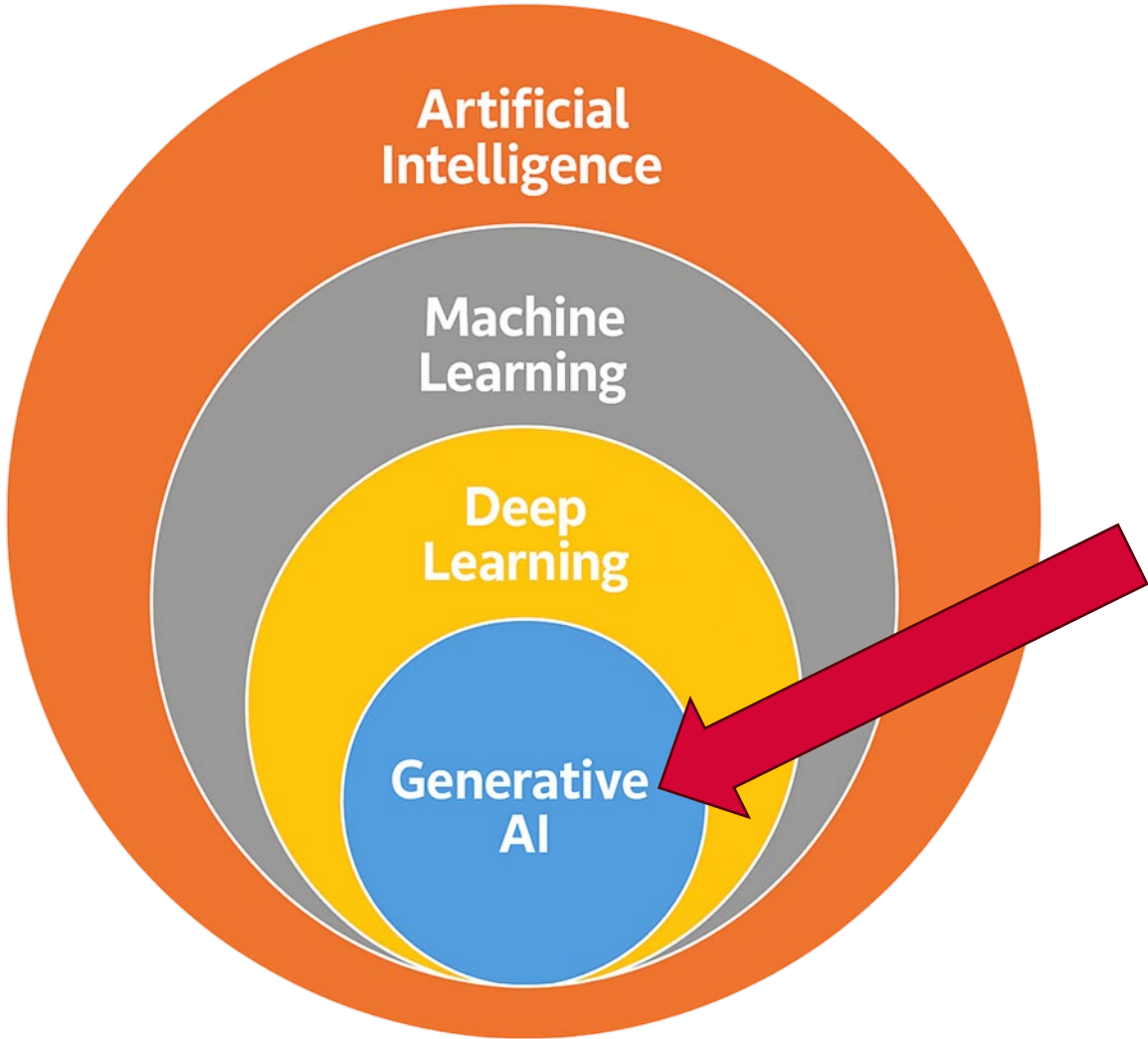
Business System Consultant



Henrik Jakobsen

Test Manager

AI I udviklingsafdelingen



- 2022
- OpenAI
- ChatGPT
- New Tech
- Generativ AI
- Data
- Large Language Model (LLM)
- PORTMAN
- Development

AI I udviklingsafdelingen

1) Generering af kode (prosa til kode)

in C#, write a method that sorts a collection of instruments by their ISIN value

INPUT

Example Code

```
csharp Kopier

public class Instrument
{
    public string ISIN { get; set; }
    public string Name { get; set; }
}

public static class InstrumentSorter
{
    /// <summary>
    /// Sorts a collection of instruments in ascending order by ISIN.
    /// </summary>
    /// <param name="instruments">The collection of instruments to sort.</param>
    /// <returns>A new list of instruments sorted by ISIN.</returns>
    /// <exception cref="ArgumentNullException">Thrown when instrument is null.</exception>
    public static List<Instrument> SortByISIN(IEnumerable<Instrument> instruments)
    {
        if (instruments == null)
            throw new ArgumentNullException(nameof(instruments));

        return instruments
            .OrderBy(i => i.ISIN, StringComparer.OrdinalIgnoreCase) // Case-insensitive
            .ToList();
    }
}
```

OUTPUT

AI I udviklingsafdelingen

2) Fejlfinding i eksisterende kode

Find the bug in this code

```
public class OrderTotalCalculator
{
    /// <summary>
    /// Returns the final total.
    /// </summary>
    public decimal CalculateTotal(decimal subtotal, int discountPercent, int taxPercent)
    {
        var discount = subtotal * (discountPercent / 100);
        var afterDiscount = subtotal - discount;

        var tax = afterDiscount * (taxPercent / 100);
        var total = afterDiscount + tax;

        return decimal.Round(total, 2);
    }
}
```

✓ Corrected version

Use a decimal divisor (100m) so the division is done in decimal arithmetic:

csharp

Kopier Rediger

```
var discount = subtotal * (discountPercent / 100m);
var afterDiscount = subtotal - discount;

var tax = afterDiscount * (taxPercent / 100m);
var total = afterDiscount + tax;

return decimal.Round(total, 2);
```

⚠ Summary of bug:

The code uses integer division when calculating percentages, which truncates fractions and produces incorrect discount/tax values.

AI I udviklingsafdelingen

3) Læring af nye teknologier

Explain the use of XmlReader i C#

Basic Example

```
csharp

using System;
using System.Xml;

class Program
{
    static void Main()
    {
        using (XmlReader reader = XmlReader.Create("books.xml"))
        {
            while (reader.Read())
            {
                // Check if the current node is an element
                if (reader.NodeType == XmlNodeType.Element && reader.Name == "title")
                {
                    // Move to the text content of the element
                    Console.WriteLine("Book Title: " + reader.ReadElementContentAsString());
                }
            }
        }
    }
}
```

Kopier Rediger

AI I udviklingsafdelingen

4) Forståelse af kompleks kode

```
Explain => double EsgCalculator::getIndexFactor(const
instrument::Instrument& instrument, const common::Date& asOfDate)
const
{
    try
    {
        double indexFactor = 100.0;
        if (instrument.assetType() == instrument::AssetTypeEnum::Bond)
        {
            instrument::Bond bond;
            if (instrumentLookup_findBond(instrument.instrumentId(), bond))
            {
                if (bond.ch2() == 'J') // ParAmount
                {
                    indexFactor =
positionCalculatorLookup_findParAmountBondIndexFactor(asOfDate,
instrument);
                }
                else if (bond.ch2() == 'P') // PoolFactor
                {
                    instrumentLookup_findPoolFactor(instrument.instrumentId(),
asOfDate, indexFactor);
                    indexFactor = indexFactor * 100;
                }
                else // Index
                {
                    // laantype 11-15 er for indeksobligationer
                    if ((bond.laantype() == 11) || (bond.laantype() == 12) ||
(bond.laantype() == 13) || (bond.laantype() == 14) || (bond.laantype() ==
15))
                    {
                        indexFactor =
positionCalculatorLookup_findBondIndexFactor(asOfDate);
                    }
                }
            }
            return indexFactor;
        }
        catch (const std::exception& e)
        {
            RETHROW(e);
        }
    }
}
```

Perfect — let's break this down step by step so the purpose and flow of the function are clear:

Function Signature

cpp

Kopier

Rediger

double EsgCalculator::getIndexFactor(const instrument::Instrument& instrument, const common::Date& asOfDate)

Return type:

double→ The *index factor* for a given financial instrument.

Parameters:

instrument

: An object representing a financial instrument (e.g., a bond).

asOfDate

: A date indicating *when* the index factor should be calculated.

Const

: The method does not modify the state of the `EsgCalculator` instance.

Purpose

The function calculates and returns an **index factor** (a scaling or adjustment value) for a given instrument, depending on its **type and characteristics**. This is common in finance where bonds can have *par amounts*, *pool factors*, or *indexation mechanisms*.

AI I udviklingsafdelingen

5) Refaktorering af kode

```

Refactor this -> double EsgCalculator::calculateMarketValue(const
instrument::Instrument& instrument, unsigned priceHierarchy, unsigned
exchangeRateSet, const basicdata::CurrencyCode& clientBaseCcyCode,
const common::Date& asOfDate, double holdingValue,
EsgHoldingValueTypeEnum holdingValueType) const
{
    try
    {
        double marketValue = 0.0;
        switch (holdingValueType)
        {
            case EsgHoldingValueTypeEnum::Unknown:
                break;
            case EsgHoldingValueTypeEnum::MarketValueBAS:
                marketValue = holdingValue;
                break;
            case EsgHoldingValueTypeEnum::MarketValueLOC:
                marketValue = calculateMarketValueFromLOC(instrument,
holdingValue, exchangeRateSet, clientBaseCcyCode, asOfDate);
                break;
            case EsgHoldingValueTypeEnum::NominalValue:
                marketValue = calculateMarketValueFromNominal(instrument,
holdingValue, priceHierarchy, exchangeRateSet, clientBaseCcyCode,
asOfDate);
                break;
        }
        return marketValue;
    }
    catch (const std::exception& e) {
        RETHROW(e);
    }
}

```

```

cpp
Kopier Rediger

// Calculates market value based on the provided holding value type.
// Returns 0.0 for Unknown (unchanged semantics).
[[nodiscard]] double EsgCalculator::calculateMarketValue(
    const instrument::Instrument& instrument,
    unsigned priceHierarchy,
    unsigned exchangeRateSet,
    const basicdata::CurrencyCode& clientBaseCcyCode,
    const common::Date& asOfDate,
    double holdingValue,
    EsgHoldingValueTypeEnum holdingValueType) const
{
    try
    {
        switch (holdingValueType)
        {
            case EsgHoldingValueTypeEnum::MarketValueBAS:
                // Already in base currency: return as-is.
                return holdingValue;

            case EsgHoldingValueTypeEnum::MarketValueLOC:
                // Convert local-currency market value to base.
                return calculateMarketValueFromLOC(
                    instrument, holdingValue, exchangeRateSet, clientBaseCcyCode, asOfDate);

            case EsgHoldingValueTypeEnum::NominalValue:
                // Price the position from nominal using prices + FX.
                return calculateMarketValueFromNominal(
                    instrument, holdingValue, priceHierarchy, exchangeRateSet, clientBaseCcyCode,
                    asOfDate);

            case EsgHoldingValueTypeEnum::Unknown:
                // Preserve original behavior: 0.0 for unknown.
                return 0.0;
        }

        // Defensive: if a new enum value is added but not handled.
        // Keeping legacy behavior (0.0) rather than throwing to avoid breaking callers.
        return 0.0;
    }
    catch (const std::exception& e)
    {
        RETHROW(e); // preserve your existing error-enrichment/logging
    }
}

```

AI I udviklingsafdelingen

6) Generering af dokumentation

cpp

Kopier Rediger

```
// Calculates market value based on the provided holding value type.  
// Returns 0.0 for Unknown (unchanged semantics).  
[[nodiscard]] double EsgCalculator::calculateMarketValue(  
    const instrument::Instrument& instrument,  
    unsigned priceHierarchy,  
    unsigned exchangeRateSet,  
    const basicdata::CurrencyCode& clientBaseCcyCode,  
    const common::Date& asOfDate,  
    double holdingValue,  
    EsgHoldingValueTypeEnum holdingValueType) const  
{
```

AI I udviklingsafdelingen

Opsummering

- Generering af kode (prosa til kode)
- Fejlfinding i eksisterende kode
- Læring af nye teknologier
- Forståelse af kompleks kode
- Refaktorering
- Generering af dokumentation



Anvendelse af AI i testen af PORTMAN

I testgruppen er vi begyndt at anvende AI i forbindelse med:

- Forberedelse af testdata (xml og dat importer)
- Analyse af testomfang på større opgaver
- Analyse / forberedelse af testcases

Desuden har vi ambitioner om at anvende AI til at:

- Analysere testomfang og testeksempler til API test ud fra struktur i Swagger
- Oprette og vedligeholde automatiske testscripts (Selenium og PlayWrite)



Anvendelse af AI i testen af PORTMAN

Vi er ved at lære ChatGPT at kende, så vi får det optimale resultat ud af den:

- Give ChatGPT en persona:
*"Du er en advanced testanalyst.
Du behersker ISTQBs testbegreber."*
- Give ChatGPT de rigtige oplysninger:
*"Autotest toolets logging skal skrives om, for at gøre det lettere at debugge.
Acceptkriterier:
- Fejlede tests skal printe både den fulde request og fulde respons
- Fejlede tests skal også specificere hvor fejlen ligger."*
- Spørge ChatGPT på den rigtige måde:
*"Kan du give mig eksempler på testcases med varianter.
Der kan være mere end en testcase pr. Acceptkriterie"*



Anvendelse af AI i testen af PORTMAN

Eksempel: Brug i testanalyse

Prompt opsætning

“Du er en testanalytiker, som kender ISTQB's testanalyseværktøjer”

➡ Perfekt 👍 Jeg kan godt påtage mig rollen som testanalytiker med kendskab til ISTQB's testanalyseværktøjer.

Prompt med simuleret best_of = 5

“Giv mig 5 forskellige sæt testcases for validering af alder (18–65) og vælg det mest dækkende. Forestil dig du anvender best_of = 5.”

Super opgave 🧙 Vi vil validere et felt for **alder**, hvor den gyldige værdi skal ligge mellem **18 og 65 (inklusive)**.

Jeg laver 5 forskellige sæt testcases med forskellige tilgange (best_of = 5), og så vurderer jeg hvilket sæt, der giver den mest dækkende test.



Anvendelse af AI i testen af PORTMAN

Grænseværdianalyse (BVA) – klassisk

- Formål: Test af værdier lige ved grænserne.

Testcases:

- Alder = 17 (lige under minimum)
- Alder = 18 (minimum gyldig)
- Alder = 19 (lige over minimum)
- Alder = 64 (lige under maksimum)
- Alder = 65 (maksimum gyldig)
- Alder = 66 (lige over maksimum)

✓ Fordel: Meget præcis på grænserne.

✗ Ulempe: Tester ikke repræsentanter længere væk fra grænserne.



Anvendelse af AI i testen af PORTMAN

Vi er stadig ved at lære ChatGPT at kende.

Der er en masse parametre som vi kan skrue på for at sikre, at vi får det "rigtige" svar fra ChatGPT.

Vi har opsat nogle kriterier for at sikre, at vi får den rigtige kvalitet i det svar, som ChatGPT kommer med indtil at vi er tilfredse med den måde, som vi skal spørge ChatGPT på.

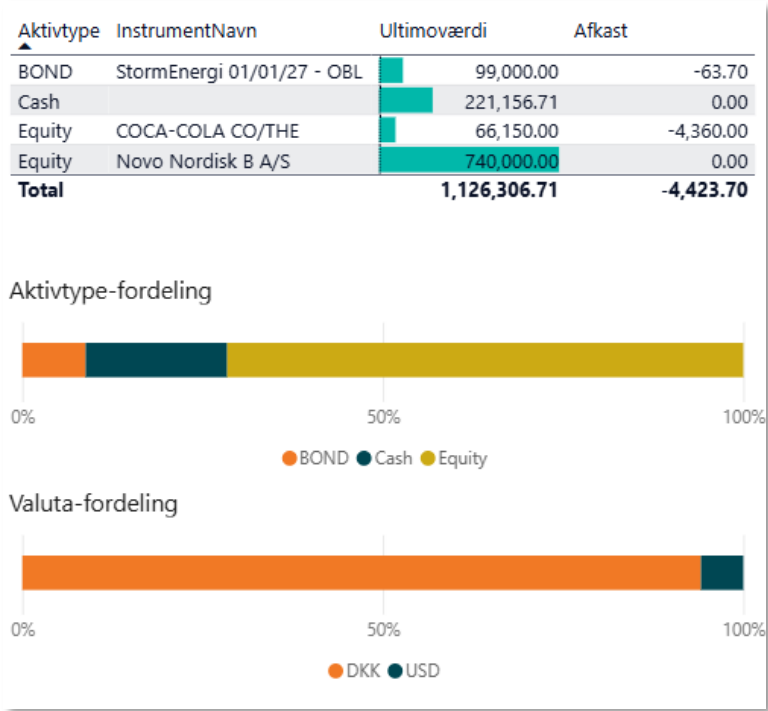
Hvad skal vi afprøve	Hvad vil vi have ud af det	Hvad er forudsætningen
Nedbryde PBler til testcases	Give titles og descriptions til testcases	PBler skal være testbare (fornuftige beskrivelser) - SMART
	Finde varianter	

Vi udvikler opskrifter (Pixi-bøger) efterhånden som vi finder den rigtige måde at spørge ChatGPT på i en given kontekst.



Forretningsbrug af ChatGPT

et Power BI eksempel



”Som en del af vores løbende arbejde med at modernisere og effektivisere vores API-løsning, vil vi **udfase understøttelsen af XML-formatet** fra og med **PORTMAN version 7.48**, så Aplet derfra udelukkende understøtter JSON-formatet.”



Power BI report

Queries [2]

	1.2 EopBasHoldingValueAtAmvPrice	ABC 123 GroupId1	ABC 123 GroupId2	ABC 123 InstrumentName	1.2 PeriodBasReturnAtAmvPrice
Holding_XML	1	99000 BOND	DKK	StormEnergi 01/01/27 - OBL	-63,69863014
Holding_JSON_ChatGPT	2	221156,71 Cash	DKK	null	0
	3	740000 Equity	DKK	Novo Nordisk B A/S	0
	4	66150 Equity	USD	COCA-COLA CO/THE	-4360

Fra XML ...

Query Settings

- PROPERTIES
- APPLIED STEPS
 - Source *
 - Navigation *
 - Expanded CalculatedHolding *
 - Changed Type with Locale *
 - Expanded InstrumentName

The screenshot displays the Power BI Desktop interface. On the left, the 'Queries' pane shows two queries: 'Holding_XML' and 'Holding_JSON_ChatGPT'. The main area shows a table with columns: '1.2 EopBasHoldingValueAtMarketPrice', 'ABC 123 GroupId1', 'ABC 123 GroupId2', 'ABC 123 InstrumentName', and '1.2 PeriodBasReturnAtAmvPrice'. The table contains four rows of data. On the right, the 'Query Settings' pane is open, showing the 'PROPERTIES' tab. Under 'APPLIED STEPS', the steps are listed: 'Source', 'Holdings', 'Expanded CalculatedHolding', 'Changed Type with Locale', and 'Expanded InstrumentName' (which is currently selected and highlighted).

	1.2 EopBasHoldingValueAtMarketPrice	ABC 123 GroupId1	ABC 123 GroupId2	ABC 123 InstrumentName	1.2 PeriodBasReturnAtAmvPrice
1	99000	BOND	DKK	StormEnergi 01/01/27 - OBL	-63,69863014
2	221156,71	Cash	DKK		0
3	740000	Equity	DKK	Novo Nordisk B A/S	0
4	66150	Equity	USD	COCA-COLA CO/THE	-4360

Forretnings-brug af ChatGPT

PORTMAN API – udfasning af XML

What can I help with?

Du er Power Query M-ekspert.

Du skal konvertere en query, der henter XML som default, til at hente og behandle JSON fra samme API.

Output (kolonner, typer, filtrering/sortering) skal være funktionelt identisk med originalen.

...

...

[Indsæt XML-script]

+



Tanker om brug af AI i PORTMAN



Ny PORTMAN hjælp

- PORTMAN-GPT



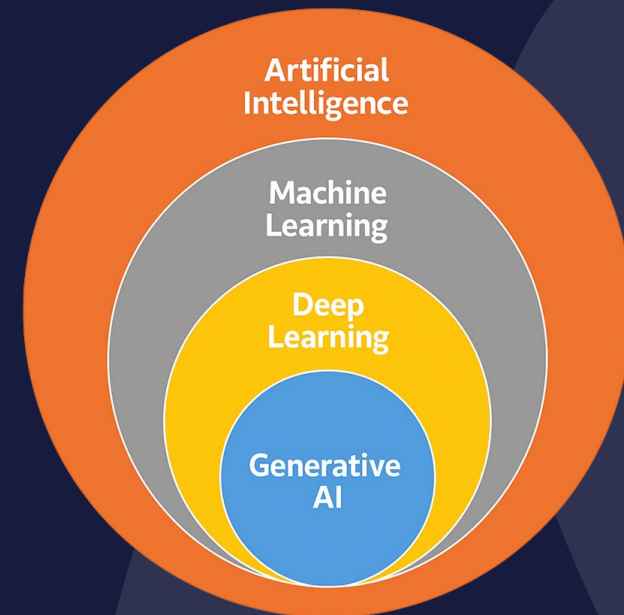
Automatiseret datakontrol

- Finde afvigere fra normalen



Simplificér afstemning

- Samle ens differencer i grupper, der kan håndteres samlet



Vitec Alocs vision med AI

Vision

Hos Vitec Aloc ser vi generativ AI som en drivkraft til at løfte effektivitet, kvalitet og kreativitet til et nyt niveau – både i og omkring PORTMAN.



- Processer
- Innovation
- Fokus
- Udbredelse
- Vision
- Effektivitet
- Kvalitet
- Kreativitet
- Time to market

Disclaimer

Dette materiale er udarbejdet af Vitec Aloc.

Materialet er udarbejdet alene til orientering for kunder eller andre, som materialet er udleveret til.

Materialet er informationsmateriale, og udgør ikke et tilbud fra Vitec. Moduler eller features nævnt i præsentationen er kun indeholdt i den endelige leverance såfremt disse er nævnt i den endelige kontrakt.

Alle råd og informationer i materialet er baseret på kilder som Vitec vurderer er pålidelige, men Vitec er hverken ansvarlig for fuldstændigheden eller nøjagtigheden af oplysningerne eller for direkte eller indirekte tab, som kan opstå på grund af fejl eller mangler heri.

Materialet bliver ikke opdateret, når først det er printet ud eller afleveret elektronisk.

Vitec har ophavsretten til materialet, som er udleveret til kundens personlige brug, og ikke til offentliggørelse noget sted.

Vitec Aloc A/S, (CVR nr. 14788484), Cortex Park Vest 3, 5230 Odense M