

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Alfint - Aplikace pro zpětné testování akcí



Autor: Vojtěch Šíma
Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování
Třída: IT4
Školní rok: 2024/25

Poděkování

Děkuji všem, co mě psychicky podpořili, když jsem v kódu měl tisíce errorů.

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2024

.....
Podpis autora

Obsah

Úvod	6
1 Teoretická část	7
1.1 Principy zpětného testování akcí	7
1.2 Finanční ukazatele	7
1.3 Metoda zpětného testování	8
1.4 Historie zpětného testování	8
1.5 Závěr	8
2 Návrh řešení	9
2.1 Architektura aplikace	9
2.2 Použité technologie	9
2.3 Databázový model	10
2.4 Bezpečnostní a výkonnostní aspekty	10
3 Implementační část	12
3.1 Získávání dat z externích API	12
3.2 Výpočet finančních ukazatelů	12
3.3 Filtrace akcí podle kritérií	13
3.4 Vizualizace a grafy	13
4 Závěr a zhodnocení	14
4.1 Shrnutí výsledků	14
4.2 Hodnocení aplikace	14
4.3 Možnosti budoucího vývoje	14
4.4 Závěr	15

Abstrakt

Tato práce se zaměřuje na vývoj aplikace pro zpětné testování akcií na základě uživatelsky definovaných parametrů. Aplikace je implementována v jazyce Python s využitím frameworku Django a API SimFin a yfinance. Umožňuje analyzovat historická data, sdílet portfolia a vizualizovat je v přehledné podobě. Dokumentace popisuje architekturu aplikace, implementační detaily a způsoby řešení. Důraz je kladen na přesnost výpočtů a uživatelskou přívětivost.

Klíčová slova

Python, Django, SimFin, yfinance, zpětné testování, finanční analýza, akcie

SEZNAM POUŽITÝCH ZKRATEK

API	Application Programming Interface,
EBITDA	Earnings Before Interest, Taxes, Depreciation, and Amortization,
EPS	Earnings Per Share,
P/E	Price-to-Earnings Ratio,
ROE	Return on Equity,
REST	Representational State Transfer,
SQL	Structured Query Language,
UI	User Interface.

ÚVOD

Představení projektu

V současné době, kdy roste zájem o investování a analýzu akciového trhu, je zpětné testování strategií nezbytným nástrojem pro investory. Tento projekt byl vytvořen s cílem poskytnout uživatelům snadno ovladatelnou aplikaci, která umožňuje analyzovat historická data akcií, vypočítávat finanční indikátory a vytvářet přehledné vizualizace. Aplikace je implementována v Pythonu s využitím frameworku Django, API SimFin a yfinance.

Motivace

Hlavní motivací pro vytvoření této aplikace byla osobní zkušenost s nedostatky dostupných nástrojů pro zpětné testování akcií. Většina stávajících řešení postrádá možnost kombinace dat z různých zdrojů a vizualizace klíčových finančních ukazatelů. Cílem bylo vytvořit nástroj, který by nabídl komplexní analýzu a zároveň byl uživatelsky přívětivý.

Cíle projektu

Hlavním cílem bylo vytvořit aplikaci, která umožní:

- Analyzovat historická data akcií a finančních ukazatelů,
- Vypočítávat klíčové indikátory, jako jsou P/E, ROE, EPS a další,
- Vizualizovat data prostřednictvím grafů a přehledných tabulek,
- Kombinovat data z více zdrojů (SimFin, yfinance),
- Poskytnout uživatelům nástroj pro testování investičních strategií,
- Sdílet mezi sebou portfolia.

Struktura práce

Práce je rozdělena do několika hlavních částí. První kapitola se zabývá teoretickou částí, která popisuje klíčové finanční ukazatele a jejich význam při zpětném testování akcií. Druhá kapitola se věnuje návrhu aplikace, včetně použité architektury, databázového modelu a výběru technologií. Třetí kapitola popisuje implementační část, kde jsou detailně rozebrány funkce aplikace, integrace API a výpočet ukazatelů. Čtvrtá kapitola shrnuje dosažené výsledky a uzavírá práci pohledem na budoucí rozvoj projektu.

1 TEORETICKÁ ČÁST

1.1 Principy zpětného testování akcií

Zpětné testování (backtesting) je klíčovým nástrojem v oblasti finančního investování, který umožňuje ověřit účinnost různých investičních strategií na historických datech. Tento proces poskytuje cenné informace o tom, jak by se strategie chovala v reálných tržních podmínkách, pokud by byla aplikována v minulosti. Základním principem zpětného testování je analýza historických cen akcií, ziskovosti a dalších ukazatelů, které umožňují modelovat chování akcií a simulovat různé obchodní scénáře.

1.2 Finanční ukazatele

Finanční ukazatele jsou zásadní pro analýzu výkonnosti společnosti a rozhodování o investicích. V rámci zpětného testování se zaměříme na ukazatele, které poskytují přehled o hodnotě, stabilitě a růstovém potenciálu společnosti. Některé z nejdůležitějších ukazatelů zahrnují:

- **P/E (Price-to-Earnings Ratio)** – Poměr ceny akcie k jejímu zisku na akcii. Tento ukazatel je často používán k určení, zda je akcie přiměřeně oceněná na základě jejího zisku. Vyšší P/E může znamenat, že investoři očekávají vysoký růst společnosti, zatímco nízké P/E může naznačovat podhodnocení.
- **ROE (Return on Equity)** – Tento ukazatel ukazuje, jak efektivně společnost využívá kapitál investovaný jejími akcionáři. Vyšší ROE obvykle signalizuje silnou výkonnost společnosti, což může být pozitivním signálem pro investory.
- **EPS (Earnings per Share)** – Zisk na akcii je jedním z nejzákladnějších ukazatelů výkonnosti společnosti. Ukazuje, kolik zisku společnost vydělala na jednu akcii. Tento ukazatel je důležitý pro hodnocení profitability společnosti.
- **P/B (Price-to-Book Ratio)** – Poměr ceny akcie k její účetní hodnotě. Tento ukazatel se používá pro analýzu toho, jak je akcie oceněná ve vztahu k jejím aktivům. Poměr menší než 1 může naznačovat podhodnocení akcie.
- **PEG (Price-to-Earnings Growth Ratio)** – Tento ukazatel kombinuje P/E s růstem zisku. Pomáhá určit, zda je akcie přiměřeně oceněná na základě očekávaného růstu zisku, což poskytuje vyváženější pohled než samotný P/E.
- **Debt-to-Equity Ratio (D/E)** – Poměr dluhu k vlastnímu kapitálu ukazuje, jak společnost financuje své aktivity. Vyšší poměr znamená větší závislost na cizích zdrojích financování, což může znamenat vyšší riziko.

1.3 Metoda zpětného testování

Zpětné testování se provádí tak, že se na historických datech testuje vybraná investiční strategie. Proces zahrnuje následující kroky:

- **Výběr historických dat** – Získání historických cen akcií a dalších relevantních finančních dat pro dané období.
- **Implementace strategie** – Definování a aplikování obchodní strategie na základě vybraných ukazatelů (např. P/E, ROE).
- **Simulace obchodních rozhodnutí** – Vytváření simulace, která na základě historických dat a strategie simuluje nákup a prodeje akcií.
- **Vyhodnocení výsledků** – Analýza výsledků zpětného testování, zhodnocení výnosů a rizik spojených se strategií.

1.4 Historie zpětného testování

Historie zpětného testování sahá až do 70. let 20. století, kdy byly první pokusy o testování obchodních strategií na historických datech prováděny manuálně. S rozvojem výpočetní techniky se zpětné testování stalo mnohem dostupnější a přesnější. V současnosti se používají sofistikované algoritmy a programy, které umožňují rychlé a efektivní testování strategií na velkých historických datech.

1.5 Závěr

Zpětné testování je neocenitelný nástroj pro každého investora, který chce ověřit efektivitu své obchodní strategie na historických datech. Kombinace správných finančních ukazatelů, analýzy historických dat a využití pokročilých technik zpětného testování pomáhá minimalizovat rizika a maximalizovat potenciální výnosy.

2 NÁVRH ŘEŠENÍ

2.1 Architektura aplikace

Architektura aplikace byla navržena s cílem zajistit modularitu, škálovatelnost a efektivní správu dat. Aplikace je postavena na frameworku Django, který je doplněn knihovnami jako yfinance pro získávání aktuálních a historických dat a SimFin pro přístup k finančním ukazatelům. Celková architektura aplikace zahrnuje následující komponenty:

- **Frontend** – Uživatelské rozhraní aplikace je navrženo tak, aby bylo přehledné a intuitivní. Umožňuje uživatelům zadávat parametry pro zpětné testování a zobrazuje výsledky ve formě tabulek a grafů.
- **Backend** – Backend je zodpovědný za získávání dat z externích API (SimFin a yfinance), výpočty finančních ukazatelů a správu databáze. Django REST framework je použit pro vytvoření API pro komunikaci mezi frontendem a backendem.
- **Databáze** – Databáze je navržena tak, aby uchovávala historická data o akcích, obchodní strategie a výsledky zpětného testování. Používá se PostgreSQL pro její robustnost a škálovatelnost.
- **Grafy a vizualizace** – Pro vizualizaci výsledků zpětného testování jsou použity knihovny jako Matplotlib a Plotly. Grafy umožňují uživatelům snadno analyzovat výsledky obchodních strategií.

2.2 Použité technologie










Aplikace využívá několik moderních technologií pro efektivní získávání dat, jejich analýzu a vizualizaci. Mezi hlavní technologie použité v aplikaci patří:

- **Django** – Framework pro vývoj webových aplikací, který umožňuje rychlý vývoj a snadnou správu backendu aplikace. Django poskytuje robustní strukturu pro zpracování požadavků, správu databáze a zabezpečení aplikace.
- **yfinance** – Knihovna pro získávání historických a aktuálních dat o akcích z Yahoo Finance. Tato knihovna je klíčová pro získání historických cen akcií a dalších relevantních dat pro zpětné testování.
- **SimFin** – API pro získávání finančních ukazatelů a fundamentálních dat o akcích. SimFin poskytuje přístup k datům, které jsou užitečné pro výpočet ukazatelů jako P/E, ROE, EPS a dalších.

- **PostgreSQL** – Relational database management system, který je použit pro správu dat o akcích, uživatelských strategiích a výsledcích testování. PostgreSQL byla zvolena pro svou spolehlivost, výkon a schopnost pracovat s velkými objemy dat.
- **Matplotlib a Plotly** – Knihovny pro tvorbu grafů a vizualizací, které jsou použity pro zobrazení výsledků zpětného testování a historických dat.

2.3 Databázový model

Databázový model aplikace je navržen tak, aby efektivně uchovával všechna potřebná data pro zpětné testování akcií a správu uživatelských portfolií. Model obsahuje několik hlavních tabulek, které uchovávají informace o akcích, jejich finančních ukazatelích a dalších relevantních údajích.

Site administration		
APLIKACE		
Balance Sheets	+ Add	 Change
Cash Flow Statements	+ Add	 Change
Income Statements	+ Add	 Change
Portfolio Stocks	+ Add	 Change
Portfolios	+ Add	 Change
Share Prices	+ Add	 Change
Stocks	+ Add	 Change
Users	+ Add	 Change
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	 Change

Tento databázový model umožňuje efektivní správu a analýzu finančních dat o akcích a uživatelských portfoliích, což je klíčové pro funkčnost aplikace pro zpětné testování akcií.

2.4 Bezpečnostní a výkonnostní aspekty

Aplikace je navržena s ohledem na bezpečnost uživatelských dat a ochranu proti běžným bezpečnostním hrozbám. Využívá bezpečnostní mechanismy, které poskytuje framework Django, včetně autentizace uživatelů, ochrany před CSRF útoky a dalších opatření.

- **Autentizace a autorizace** – Aplikace využívá vestavěnou autentizaci Django pro správu přihlašování uživatelů. Při přihlašování uživatel zadá své uživatelské jméno a heslo, které jsou ověřeny proti databázi uživatelů. Django automaticky šifruje hesla pomocí algoritmu bcrypt, což zajišťuje jejich bezpečné uchování. Pro ochranu přihlášení je použita funkce `login_required`, která zajišťuje, že pouze přihlášení uživatelé mohou přistupovat k určitým částem aplikace, jako jsou portfolia uživatelů.
- **CSRF ochrana** – Pro ochranu proti CSRF (Cross-Site Request Forgery) útokům Django automaticky přidává tokeny do každého formuláře. Tento token musí být odeslán spolu s každým požadavkem, což brání neautorizovanému zasílání požadavků na server. Formuláře v aplikaci obsahují příkaz `{ % csrf_token % }` pro generování a ověřování tokenů.
- **Šifrování dat** – Django automaticky šifruje hesla uživatelů a používá HTTPS pro šifrování komunikace mezi serverem a klientem, což zajišťuje bezpečnost přenášených dat.
- **Oprávnění a přístupová kontrola** – V aplikaci je implementována kontrola přístupu na základě uživatelských oprávnění. Příkladem je použití dekorátoru `@login_required`, který zajistí, že pouze přihlášení uživatelé mohou provádět určité akce, jako je změna stavu sdílení portfolia:

```

1 @login_required
2 def toggle_share(request, portfolio_id):
3     portfolio = get_object_or_404(Portfolio, user=request.user)
4     portfolio.is_shared = not portfolio.is_shared
5     portfolio.save()
6     return redirect('view_portfolio', portfolio_id=portfolio_id)
7

```

Kód 2.1: Login required

Tento dekorátor zajistí, že pouze uživatel, který vlastní konkrétní portfolio, může změnit jeho stav (např. sdílení portfolia).

- **Další bezpečnostní opatření** – Aplikace využívá Django zabezpečené mechanismy, jako je ochrana proti SQL injection a XSS (Cross-Site Scripting). Všechny formuláře jsou ošetřeny proti těmto útokům, což zajišťuje bezpečnost aplikace.

Výkonnost aplikace je také optimalizována pro efektivní zpracování velkých objemů dat. Použití asynchronního zpracování a cachování výsledků zaručuje, že aplikace bude reagovat rychle i při vysokém zatížení.

3 IMPLEMENTAČNÍ ČÁST

3.1 Získávání dat z externích API

Aplikace pro zpětné testování akcií používá externí API pro získávání historických dat a fundamentálních ukazatelů. Hlavními API, která jsou v aplikaci použita, jsou SimFin pro fundamentální data a yfinance pro historické ceny akcií.

Pro získávání dat z API SimFin je použita funkce `load_simfin_data()`, která načítá roční finanční údaje jako příjmy, bilanční údaje, peněžní toky a historické ceny akcií. Tato data jsou následně uložena do databáze. Příklad kódu pro načítání dat vypadá takto:

```
1
2 def load_simfin_data():
3     income_data = sf.load(dataset='income', variant='annual', market='us',
4     index=['Ticker', 'Fiscal Year'])
5     balance_data = sf.load(dataset='balance', variant='annual', market='us',
6     index=['Ticker', 'Fiscal Year'])
7     shareprices_data = sf.load(dataset='shareprices', variant='daily',
8     market='us', index=['Ticker', 'Date']))
9
```

Kód 3.1: Simfin data

Tato funkce načte data pro každý ticker (akcii) a uloží je do příslušných tabulek v databázi (např. IncomeStatement, BalanceSheet, SharePrices).

3.2 Výpočet finančních ukazatelů

Po získání dat jsou následně vypočítány klíčové finanční ukazatele, které jsou použity pro analýzu akcií a zpětné testování strategií. Výpočet těchto ukazatelů je prováděn v metodě `calculate_ratios()`, která prochází všechny akcie a na základě dat v databázi vypočítá ukazatele jako P/E, ROA, ROE, a další.

Například výpočet P/E poměru probíhá následovně:

```
1 def calculate_ratios():
2     stocks = Stock.objects.all()
3     for stock in stocks:
4         income_statement = IncomeStatement.objects.filter(stock=stock).first()
5         if stock.market_cap and income_statement.net_income != 0:
6             stock.pe_ratio = round(stock.market_cap / income_statement.net_income, 2)}
```

Kód 3.2: Ratios

Tato funkce pro každou akcii vypočítá P/E poměr na základě tržní kapitalizace a čistého zisku z posledního dostupného výkazu zisků a ztrát.

3.3 Filtrace akcí podle kritérií

Pro efektivní analýzu a zpětné testování je v aplikaci implementována filtrace akcí podle různých kritérií, jako jsou sektor, průmysl, a finanční ukazatele. Filtry jsou uloženy v uživatelských preferencích a aplikovány na dotazy v databázi. Kód pro filtrování akcí podle vybraných parametrů vypadá takto:

```
1 filters = request.session.get('filters', [])
2 q_object = Q()
3 for f in filters:
4     field = f.get('field')
5     value = f.get('value')
6     q_object &= Q(**{f"field
```

Kód 3.3: Filtr

Tento filtr umožňuje uživatelům vyhledávat akcie na základě různých parametrů, jako jsou sector nebo industry, a následně je použít pro zpětné testování.

3.4 Vizualizace a grafy

Aplikace umožňuje vizualizaci historických dat akcí a výsledků zpětného testování. Pro tento účel je použita knihovna Plotly, která generuje interaktivní grafy. Jeden z grafů, který zobrazuje vývoj ceny akcie, je generován takto:

```
1 <h2 class="subtitle">Graf vývoje ceny akcie</h2>
2 <div id="interactive-chart"></div>
3 <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
4 <script>
5     const chartData = [{ chart_data|safe }];
6     Plotly.newPlot('interactive-chart', chartData.data, chartData.layout);
```

Kód 3.4: Grafy

Tento kód využívá Plotly pro vykreslení interaktivního grafu, který zobrazuje data o vývoji ceny akcie na základě historických údajů. Grafy jsou dynamické a umožňují uživatelům interaktivně prozkoumávat data.

4 ZÁVĚR A ZHODNOCENÍ

4.1 Shrnutí výsledků

Cílem této práce bylo vyvinout aplikaci pro zpětné testování akcií, která by uživatelům umožnila testovat jejich investiční strategie na historických datech a analyzovat výkonnost akcií na základě různých finančních ukazatelů. Po realizaci a testování aplikace lze shrnout následující výsledky:

- Aplikace úspěšně integruje dvě klíčová externí API (SimFin a yfinance) pro získávání historických dat a finančních ukazatelů.
- Finanční ukazatele, jako jsou P/E, ROE, ROA, a další, jsou správně vypočítávány a uloženy do databáze pro následné použití v analýzách.
- Uživatelé mají možnost definovat vlastní investiční strategie a provádět zpětné testování na historických datech.
- Výsledky zpětného testování jsou generovány a vizualizovány pomocí interaktivních grafů, což zlepšuje analýzu a porozumění výsledkům.
- Aplikace umožňuje uživatelům sdílet portfolia.

Aplikace tedy splňuje své základní cíle a poskytuje funkční nástroj pro zpětné testování akcií.

4.2 Hodnocení aplikace

Aplikace byla vyvinuta s důrazem na uživatelskou přívětivost, bezpečnost a výkonnost. Testování aplikace ukázalo, že všechny základní funkce, jako je získávání dat z externích API, výpočet finančních ukazatelů a zpětné testování strategií, fungují správně. Dále byly provedeny testy na výkonnost aplikace, které potvrdily, že i při vyšším zatížení dokáže aplikace správně a efektivně zpracovávat velké objemy dat.

Jedním z hlavních přínosů aplikace je její flexibilita, která umožňuje uživatelům definovat vlastní investiční strategie a analyzovat je na historických datech. Výsledky zpětného testování jsou prezentovány v přehledné formě, která umožňuje uživatelům rychlou analýzu výkonnosti jejich strategií.

4.3 Možnosti budoucího vývoje

Vzhledem k tomu, že aplikace byla navržena s ohledem na rozšiřitelnost a flexibilitu, existuje několik možností pro její budoucí vylepšení. K těmto možnostem patří:

- **Predikce pomocí AI a strojového učení** – Integrace modelů strojového učení pro predikci budoucího vývoje cen akcií na základě historických dat, klíčových finančních ukazatelů a dalších relevantních faktorů. To by uživatelům pomohlo lépe odhadnout potenciální výnosy a rizika.
- **Kalendář ekonomických událostí** – Přidání interaktivního kalendáře obsahujícího významné ekonomické události, jako jsou zveřejnění výsledků společností, makroekonomická data nebo rozhodnutí centrálních bank. Tato funkcionality by uživatelům poskytla lepší přehled o možných rizicích a příležitostech.
- **Individuální zprávy pro každou akcii** – Zahrnutí sekce se souhrnem aktuálních zpráv a novinek souvisejících s každou akcií. Tímto způsobem by uživatelé mohli získat rychlý přehled o faktorech, které mohou ovlivnit cenu konkrétní akcie.
- **Vážení akcií v portfoliu** – Možnost vážit jednotlivé akcie v portfoliu podle uživatelem zadaných procent. To by umožnilo simulovat reálnější investiční strategie, kde mají různé akcie různou důležitost.
- **Pokročilé analýzy portfolia** – Zavedení nástrojů pro analýzu portfolia, jako je sledování celkové volatility, Sharpeho poměru nebo dalších ukazatelů, které by pomohly uživatelům optimalizovat jejich investice.
- **Sledování sentimentu trhu** – Možnost analyzovat sentiment trhu pomocí zpracování textů (např. z finančních článků a zpráv). Tato funkcionality by uživatelům pomohla identifikovat, zda převládá pozitivní, neutrální nebo negativní nálada ohledně konkrétní akcie či trhu obecně.
- **Komunitní diskuse a hodnocení** – Integrované fórum nebo sekce pro komentáře by umožnily uživatelům diskutovat o investičních strategiích, ekonomických událostech nebo konkrétních akciích. Možnost hodnotit portfolia a strategie by dále podpořila interakci mezi uživateli.

Aplikace má také potenciál pro komerční využití, zejména pokud bude rozšířena o pokročilé funkce analýzy a optimalizace investičních strategií.

4.4 Závěr

Tato práce se zaměřila na vytvoření aplikace pro zpětné testování akcií, která by poskytovala nástroje pro analýzu akciových trhů na základě historických dat a finančních ukazatelů. Aplikace byla úspěšně navržena, implementována a otestována. Testování prokázalo správnost výpočtů a funkčnost aplikace i při velkém objemu dat.

Díky implementaci flexibilních analytických nástrojů a vizualizací má aplikace vysoký potenciál pro využití v praxi a pro rozšíření o nové funkce v budoucnu. Tento projekt tak představuje silný základ pro další vývoj nástrojů pro analýzu a testování akciových strategií.

Zdrojový kód projektu je dostupný na GitHubu (<https://github.com/Vojtik1/Alfint>)

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- **Django Documentation** – Django Software Foundation. Dostupné z: <https://www.djangoproject.com/>
- **yfinance Documentation** – Yahoo Finance API for Python. Dostupné z: <https://pypi.org/project/yfinance/>
- **SimFin API Documentation** – SimFin. Dostupné z: <https://simfin.com/docs>
- **The Intelligent Investor** – Benjamin Graham, HarperBusiness, 2003.
- **Plotly: A Comprehensive Guide** – Plotly Technologies Inc. Dostupné z: <https://plotly.com/python/>
- **ChatGPT** - ChatGPT. Dostupné z: <https://chatgpt.com/>
- **Stack Overflow** - Stack Overflow. Dostupné z: <https://stackoverflow.com/>