

POLITECNICO DI TORINO

Dipartimento di Ingegneria Gestionale e della Produzione

Corso di Laurea in Ingegneria Gestionale

Classe L-8 Ingegneria dell'Informazione



Ottimizzazione del portafoglio finanziario

Relatore

Prof. Fulvio Corno

Candidato

Pizzato Chiara (s219345)

Anno Accademico 2017/2018

Indice

| | |
|---|----|
| Proposta di progetto | 3 |
| Descrizione dettagliata del problema | 5 |
| Descrizione del data-set utilizzato per l'analisi..... | 9 |
| Descrizione ad alto livello delle strutture dati e degli algoritmi utilizzati | 10 |
| Diagramma delle classi delle parti principali dell'applicazione | 14 |
| Alcune videate dell'applicazione | 16 |
| Tabelle con risultati sperimentali ottenuti | 19 |
| Valutazione risultati ottenuti e conclusioni | 21 |

Proposta di progetto

Studente proponente

s219345 Pizzato Chiara

Titolo della proposta

Ottimizzazione del portafoglio finanziario

Descrizione del problema proposto

L'applicazione desktop si propone di risolvere il problema relativo alla selezione degli investimenti per la composizione del portafoglio sulla base di determinati parametri fissati dall'utente (budget, rating minimo, rendimento e durata). Si tratta di decidere in quali titoli e con quali quote investire i capitali disponibili in modo da massimizzare il ricavo atteso entro una determinata soglia di rischio oppure minimizzare il rischio entro una determinata soglia di rendimento minima.

Descrizione della rilevanza gestionale del problema

La definizione degli investimenti può riguardare sia un soggetto imprenditoriale nell'ambito della sua attività di impresa relativamente alla componente di liquidità, sia più in generale un privato cittadino nel miglioramento della sua qualità di vita.

L'analisi dei possibili investimenti e le capacità di gestione sono fondamentali per la valutazione del portafoglio finanziario. Il problema affrontato dall'applicazione permetterà di creare un modello di simulazione di un sistema reale, raccogliere dati per la sperimentazione, analizzare i risultati in modo critico e valutare i flussi di investimento in possibili scenari alternativi.

L'obiettivo principale è fornire all'utente uno strumento efficace per poter gestire il proprio capitale in base alle proprie esigenze ed ai propri obiettivi di guadagno e di possibile reinvestimento in attività imprenditoriali.

L'applicazione risponderà a tutte le attività che normalmente richiedono le capacità proprie dell'ingegnere gestionale nell'ambito dell'analisi degli investimenti: capacità di individuare, valutare e gestire i principali rischi finanziari; capacità di comprendere l'interazione tra l'andamento dei mercati finanziari, le opportunità di investimento e le scelte di struttura del capitale; capacità di analizzare le performance economico-finanziarie dell'investimento e capacità di definire il portafoglio ottimo.

Descrizione dei data-set per la valutazione

L'applicazione utilizzerà un data-set contenente un sottoinsieme di titoli obbligazionari a tasso fisso del mercato obbligazionario mondiale. In particolare il data-set farà riferimento a titoli obbligazionari di tipologia Corporate e Government. Ogni obbligazione sarà caratterizzata principalmente da un nome, un tasso d'interesse, una scadenza, un indice di rating assegnato dall'agenzia Standard & Poor's e un prezzo di mercato. Tutte le informazioni presenti nel database faranno riferimento a dati storici del mercato obbligazionario dell'anno corrente.

Descrizione preliminare degli algoritmi coinvolti

L'applicazione permetterà di simulare la creazione di un portafoglio finanziario sulla base dei dati forniti in input dall'utente (capitale iniziale, rendimento, rating minimo e durata). Per la definizione

del portafoglio verrà implementato un algoritmo ricorsivo che terrà conto dei vari vincoli alla base del problema. Dato l'elevato numero di titoli finanziari presenti nel database, sarà necessario ottimizzare al meglio l'algoritmo in modo da ridurre le tempistiche di calcolo del piano d'investimento ottimale (ad esempio verrà fissata una percentuale minima di possesso per ogni titolo in portafoglio e verranno introdotte altre ipotesi semplificative).

Descrizione preliminare delle funzionalità previste per l'applicazione software

Tramite l'applicazione l'utente potrà definire i parametri desiderati per la definizione del portafoglio finanziario. In particolare l'utente potrà scegliere il capitale iniziale disponibile, il rendimento atteso dall'investimento, il livello di rischio assunto (rating minimo), l'arco temporale e l'obiettivo dell'investimento (massimizzazione dei ricavi considerato il rischio fornito come livello massimo o minimizzazione del rischio complessivo considerato il rendimento fornito come valore minimo). L'applicazione elaborerà i dati forniti in input dall'utente per restituire l'allocazione ottimale finanziaria. L'utente potrà così ottenere informazioni sulla distribuzione nel tempo e sulla diversificazione degli investimenti del portafoglio finanziario, informazioni che vengono illustrate anche tramite grafici (un grafico relativo alla diversificazione del rischio ed un grafico relativo alla diversificazione dell'investimento).

Descrizione dettagliata del problema

L'applicazione affronta il problema relativo alla scelta del portafoglio finanziario ottimo, cioè l'insieme degli investimenti in titoli che permette all'investitore di ottenere il guadagno futuro desiderato a partire dalla ricchezza iniziale allocata per l'investimento. Infatti il termine investimento identifica l'attività finanziaria di un soggetto finalizzata all'acquisizione o creazione di nuove risorse. La definizione degli investimenti può riguardare sia un soggetto imprenditoriale nell'ambito della sua attività di impresa relativamente alla componente di liquidità, sia in senso più ampio un privato cittadino nel miglioramento della sua qualità di vita.

Un investimento è caratterizzato dal capitale iniziale (budget), dal rendimento, dalla durata e dal rischio relativo al mancato raggiungimento degli obiettivi finanziari prestabiliti. Ogni investimento comporta un'assunzione di rischio da parte dell'investitore. Questo rischio rappresenta il grado di incertezza che il mercato attribuisce all'effettiva realizzazione dei rendimenti attesi. In genere l'investitore, di fronte alla presenza di un possibile rischio, tende a diversificare i propri investimenti per ridurre tale rischio a parità di rendimento atteso. La diversificazione del rischio e dell'investimento si concretizza nella scelta di titoli differenti per società e settore di riferimento, per mercato di appartenenza, per tipologia e per durata.

L'insieme delle attività finanziarie possedute da un soggetto a seguito di un investimento costituisce un portafoglio finanziario. L'attività di investimento in un portafoglio di titoli è meno rischiosa rispetto all'allocazione dell'intero capitale su un'unica tipologia di titolo. Infatti tramite l'operazione di diversificazione degli investimenti si cerca di ridurre il più possibile il rischio di perdite future dovuto all'andamento negativo di un singolo titolo (*principio di diversificazione del portafoglio*). L'ottimizzazione del portafoglio finanziario consiste nel trovare la migliore strategia d'investimento che permette di massimizzare il guadagno ed allo stesso tempo minimizzare il rischio: un portafoglio "ottimizzato" permette di ottenere un rendimento maggiore a parità di rischio oppure un rischio minore a parità di rendimento. Quindi il problema di ottimizzazione del portafoglio consiste nel decidere in quali titoli e con quali quote investire i capitali disponibili in modo da massimizzare il ricavo atteso entro una determinata soglia di rischio oppure minimizzare il rischio entro una determinata soglia di rendimento minima.

La teoria moderna di allocazione delle risorse finanziarie è stata introdotta dall'economista statunitense Harry Markowitz nell'articolo "*Portfolio Selection*" pubblicato sul *Journal of Finance* nell'anno 1952. Markowitz ha condiviso il premio Nobel per l'economia con Merton Miller e William Sharpe "*per i contributi pionieristici nell'ambito dell'economia finanziaria*".

Markowitz sostiene che, per ottimizzare il rendimento degli investimenti, è necessario realizzare un portafoglio diversificato. Il principio di diversificazione del portafoglio permette di eliminare la componente di rischio specifico legata alla singola impresa, mantenendo la componente di rischio sistematico legata all'economia nel suo complesso. La teoria del portafoglio, basandosi sull'approccio media-varianza, mostra come combinare tra loro i diversi strumenti finanziari in un unico portafoglio per ottenere il rendimento massimo per un dato livello di rischio. Per poter costruire un portafoglio efficiente è necessario individuare una combinazione di titoli tale da minimizzare il rischio e massimizzare il rendimento complessivo, compensando i diversi andamenti dei singoli titoli.

La teoria di Markowitz si basa sull'ipotesi che gli investitori vogliono massimizzare la propria ricchezza finale e sono avversi al rischio. Inoltre gli unici parametri che possono indirizzare l'investitore nella scelta dei titoli da inserire in portafoglio sono il valore atteso e la deviazione

standard.

Il rendimento r_t di un titolo è una variabile casuale caratterizzata dal valore medio μ_r , che misura il rendimento atteso, e dalla varianza σ_r^2 , che misura l'incertezza relativa al conseguimento del valore atteso. Nel caso in cui si abbia a disposizione un campione di T osservazioni sotto forma di serie storica dei rendimenti, è possibile calcolare il rendimento atteso μ_r come

$$\bar{r} = \frac{1}{T} \sum_{t=1}^T r_t$$

e la varianza σ_r^2 dei rendimenti come

$$s_r^2 = \frac{1}{T-1} \sum_{t=1}^T (r_t - \bar{r})^2$$

La varianza rappresenta il rischio dell'attività finanziaria, cioè la dispersione rispetto al valore atteso (un'attività finanziaria è tanto più rischiosa quanto più elevata è la probabilità che i rendimenti futuri si disperdano rispetto al valore medio stimato). Nel modello di Markowitz la distribuzione delle probabilità relative alla definizione dei rendimenti è di tipo gaussiano; in particolare una tale variabile, distribuita normalmente, è descritta in modo completo tramite la sua media e la sua varianza.

Il calcolo del rischio (rappresentato dalla varianza) e del rendimento (rappresentato dalla media) di un portafoglio finanziario costituito da N titoli si basa sulle seguenti formule

$$E(r_p) = \mu_p = \sum_{i=1}^N \mu_i \omega_i$$
$$VAR(r_p) = \sigma_p^2 = \sum_{i=1}^N \omega_i^2 \sigma_i^2 + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \omega_i \omega_j \sigma_i \sigma_j \rho_{ij}$$

dove ω_i è la quantità di ricchezza investita nell' i -esimo titolo e ρ_{ij} è il coefficiente di correlazione tra il titolo i -esimo e il titolo j -esimo.

Il problema dell'investitore consiste nel decidere come allocare la propria ricchezza lungo un determinato orizzonte temporale per massimizzare la ricchezza futura minimizzando il rischio, caratteristica che viene riassunta tramite la funzione di utilità attesa. La funzione di utilità attesa viene quindi massimizzata nel rispetto dei vincoli definiti per il problema (ad esempio vincolo di budget, vincolo di rischio minimo, vincolo di rendimento atteso). La soluzione ottenuta dal problema di ottimizzazione (un problema di ottimizzazione si basa sulla ricerca della soluzione migliore tra tutte le possibili soluzioni) permette di massimizzare il rendimento atteso fissato il livello di rischio oppure minimizzare il rischio per un dato livello di rendimento atteso.

L'applicazione realizzata gestisce il sotto problema di definizione di un portafoglio ottimo di titoli obbligazionari a tasso fisso.

Un'obbligazione (in inglese *bond*) è un titolo di debito emesso da società o enti pubblici che attribuisce al suo possessore il diritto al rimborso a scadenza del capitale prestato all'emittente più un interesse su tale somma. Essa rappresenta una forma di investimento per il detentore ed uno strumento di reperimento di liquidità per l'emittente. L'obbligazione è considerata una forma di investimento a rischio finanziario minore rispetto all'azione, in quanto non soggetta a quotazione sul mercato azionario e con tasso d'interesse generalmente fisso a seguito della stipulazione del contratto di vendita. Il rimborso del capitale al possessore del titolo da parte dell'emittente avviene alla scadenza in un'unica soluzione, mentre gli interessi sono liquidati periodicamente. L'interesse

corrisposto periodicamente è detto *cedola* perché in passato per riscuoterlo si doveva staccare il tagliando numerato dal certificato dell'obbligazione.

Il prezzo di un'obbligazione è determinato dal tasso d'interesse di mercato, dalla cedola, dalla scadenza dell'obbligazione e dalla rischiosità. Esso deve considerare il rischio di insolvenza dell'emittente, il rischio di tasso d'interesse, il rischio di valuta e il rischio di liquidità. Dopo l'emissione, il prezzo di un'obbligazione può variare solo in seguito a una variazione dei tassi di interesse o della rischiosità dell'emittente, di conseguenza le obbligazioni hanno un valore stabile nel tempo e sono tendenzialmente poco sensibili alle oscillazioni del mercato.

La qualità dell'emittente di un'obbligazione viene espressa tramite la misura rating definita sulla base di determinate caratteristiche e criteri. Diverse agenzie di rating (ad esempio Standard & Poor's, Moody's e Fitch) eseguono un continuo monitoraggio delle società emittenti proponendo differenti notazioni sulla qualità dei debitori. Il rating è un punteggio ponderato che gli istituti attribuiscono ai differenti emittenti. Maggiore è il rischio rappresentato dalle obbligazioni di una determinata società, minore sarà il voto e più alta la remunerazione spettante all'investitore. Il rating è un metodo utilizzato per valutare sia i titoli obbligazionari, sia le imprese in base al loro rischio finanziario.

Standard & Poor's, in particolare, è una società privata con base negli Stati Uniti che realizza ricerche finanziarie e analisi su titoli azionari ed obbligazionari. Il rating Standard & Poor's misura la capacità di restituire i crediti ricevuti in base a una scala standardizzata per la valutazione a lungo termine.

| Rating S&P | Descrizione | |
|------------|--|---|
| AAA | Eccellenti capacità di onorare le obbligazioni assunte | Grado d'Investimento <i>(Investment grade)</i> |
| AA+ | Alta qualità, rischio molto basso, ma la suscettibilità allo stesso nel lungo periodo appare superiore a quello delle AAA | |
| AA | Ottime capacità di onorare le obbligazioni assunte. Minime differenze rispetto al grado più alto | |
| AA- | | |
| A+ | | |
| A | Buone capacità di rispettare gli obblighi finanziari, ma in qualche modo sono maggiormente suscettibili ai cambiamenti delle circostanze e delle condizioni economiche, rispetto agli obbligati con un rating migliore | |
| A- | | |
| BBB+ | Adeguate capacità di rispettare gli obblighi finanziari. Tuttavia, condizioni economiche avverse o cambiamenti delle circostanze sono più facilmente associabili ad una minore capacità di adempiere agli obblighi finanziari assunti. | |
| BBB | | |
| BBB- | | |
| BB+ | Meno vulnerabili nel breve periodo, rispetto ad obbligati con rating inferiore. Tuttavia, il dover fronteggiare condizioni di incertezza economica, finanziaria, amministrativa potrebbe interferire con le capacità di soddisfacimento degli obblighi assunti | Grado di non investimento <i>(Speculative grade)</i> |
| BB | | |
| BB- | | |
| B+ | Più vulnerabili rispetto agli obbligati valutati 'BB', ma vi sono ancora le capacità per rispettare gli impegni finanziari. Condizioni economiche e/o finanziarie impreviste, ridurranno probabilmente le capacità e la volontà dell'obbligato, di adempiere | |
| B | | |
| B- | | |
| CCC+ | Attualmente vulnerabile, e la solvibilità delle obbligazioni assunte dipende prevalentemente da condizioni economiche e finanziarie favorevoli | |
| CCC | | |
| CCC- | | |
| CC | Attualmente molto vulnerabile, e la solvibilità delle obbligazioni assunte, dipende prevalentemente da condizioni economiche e finanziarie favorevoli | |
| C | Estremamente vulnerabile, forse in bancarotta o in ritardo nei pagamenti, che ancora vengono comunque effettuati | |
| CI | Pagamento degli interessi non pervenuto | |
| R | Amministrazione controllata/Liquidazione. Durante questo periodo, gli obbligati potrebbero decidere di pagare soltanto una parte dei creditori. | |
| SD | Default Selettivo. Alcune obbligazioni non sono state rimborsate | |
| D | Default/Insolvenza di tutte o la maggior parte delle obbligazioni emesse. | |
| NR | Non valutata | |

(fonte [https://it.wikipedia.org/wiki/Standard %26 Poor%27s](https://it.wikipedia.org/wiki/Standard_%26_Poor%27s))

Il rendimento di un'obbligazione è direttamente proporzionale alla rischioosità dell'emittente. Un'emittente con rating basso dovrà pagare cedole maggiori per trovare investitori disposti a finanziarla rispetto a un'emittente con rating più alto.

L'applicazione presenta alcune criticità dovute all'esigenza di semplificazione del problema di ottimizzazione del portafoglio finanziario. Questo ha determinato l'allontanamento dallo scenario reale del problema analizzato, ma è stato necessario per far fronte alla difficoltà di reperimento delle informazioni complete sui titoli obbligazionari e al peggioramento delle performance dell'algoritmo ricorsivo, con il conseguente aumento dei tempi di elaborazione, dovuto principalmente alle considerevoli dimensioni del database.

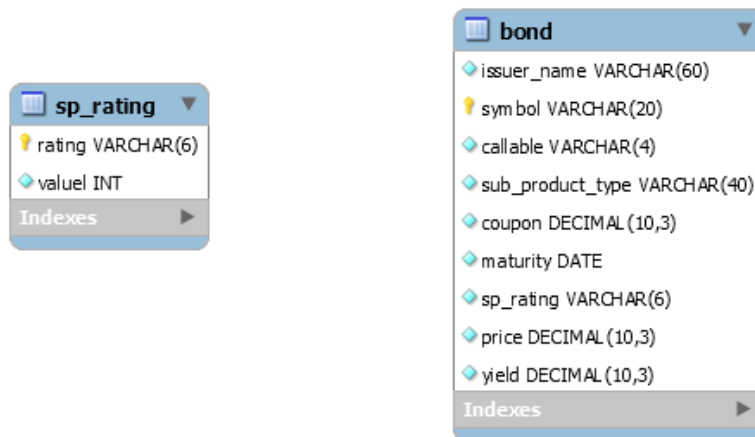
La difficoltà nel reperimento delle informazioni dei titoli obbligazionari ha comportato la semplificazione del metodo per il calcolo del rischio e del rendimento di portafoglio. Infatti il calcolo non è gestito in modo perfettamente coerente alla teoria di Markowitz, in quanto non si ha a disposizione una serie storica di dati su cui poter calcolare la media e la varianza dei titoli obbligazionari.

L'esigenza di ridurre i tempi di esecuzione dell'algoritmo ricorsivo ha richiesto invece l'introduzione di alcuni vincoli aggiuntivi al problema di ottimizzazione (cfr *"Descrizione ad alto livello delle strutture dati e degli algoritmi utilizzati"*).

Gli input dell'applicazione sono rappresentati dai parametri definiti dall'utente a livello di interfaccia grafica: il capitale iniziale (budget), il rating minimo, il rendimento richiesto e la durata. L'output è rappresentato dalla definizione dell'allocazione ottima del capitale nei vari titoli obbligazionari lungo l'orizzonte temporale considerato e dal calcolo del rendimento complessivo ottenuto a fine periodo. In particolare l'applicazione permette di visualizzare la composizione del portafoglio a differenti date dell'orizzonte temporale.

Descrizione del data-set utilizzato per l'analisi

Il data-set utilizzato per l'analisi del problema e per testare il funzionamento del software sviluppato contiene le principali informazioni relative ai titoli a tasso fisso del mercato obbligazionario e il metodo di classificazioni di rating definito dalla società Standard & Poor's. Esso è strutturato secondo il seguente diagramma ER.



Il database *investments* contiene due tabelle: *bond*, in cui sono presenti le informazioni relative alle singole obbligazioni, e *sp_rating*, in cui è presente la scala di valutazione dell'indice rating. La tabella *sp_rating* contiene il rating definito dall'agenzia Standard & Poor's, cioè l'indice utilizzato come metodo di classificazione per valutare i titoli obbligazionari e le imprese in base al relativo rischio finanziario. La tabella è costituita dalla colonna *rating*, che riporta la scala dell'agenzia Standard & Poor's con i relativi valori di rating, e dalla colonna *value*, che associa ad ogni rating un valore numerico. Alla scala di rating è stata associata una scala di valori numerici (scala numerica da 0 a 24 crescente al crescere del rating, dove il rating massimo AAA corrisponde al valore numerico 24) per permettere un utilizzo più semplice dell'attributo all'interno dell'applicazione.

La tabella *bond* è costituita dalle colonne *issuer_name* (nome del titolo obbligazionario), *symbol* (codice univoco del titolo obbligazionario), *callable* (indicazione relativa all'esigibilità anticipata o meno del titolo), *sub_product_type* (tipologia di titolo obbligazionario), *coupon* (tasso di rendimento attuale), *maturity* (scadenza), *sp_rating* (rating Standard & Poor's), *price* (prezzo di acquisto) e *yield* (tasso di rendimento stimato alla scadenza del titolo obbligazionario).

L'attributo *yield* (yield to maturity YTM) rappresenta il tasso implicito di rendimento del titolo obbligazionario nel caso in cui si mantenga il titolo sino alla scadenza e vengano pagate le relative cedole rispettando tempi e modalità di pagamento. Invece l'attributo *coupon* rappresenta la cedola del titolo obbligazionario, cioè l'interesse pagato dall'obbligazione secondo le modalità prefissate. Il data-set comprende un sottoinsieme di titoli a tasso fisso del mercato obbligazionario mondiale: obbligazioni Corporate e obbligazioni Government (classificazione definita dall'attributo *sub_product_type*). Esso contiene esclusivamente titoli obbligazionari senza opzione callable, cioè senza clausola di rimborso anticipato da parte dell'emittente del titolo. In questo modo tutte le obbligazioni vengono sempre rimborsate alla scadenza e ciò permette di semplificare il calcolo del portafoglio finanziario ottimo (la trattazione di obbligazioni callable avrebbe introdotto maggiore complessità a livello algoritmico). Inoltre sono stati esclusi dal data-set i titoli obbligazionari senza cedola (zero coupon bond ZCB).

Descrizione ad alto livello delle strutture dati e degli algoritmi utilizzati

Le principali strutture dati e l'algoritmo ricorsivo si trovano nella classe *Model* del package *it.polito.tdp.financialportfolio.model* del progetto. Il package *it.polito.tdp.financialportfolio.db* è utilizzato per la connessione al database (classe *DBConnect*) e per il caricamento tramite query SQL dei dati presenti nel database (classe *FinancialPortfolioDAO*). Infine il package *it.polito.tdp.financialportfolio* contiene la classe *Main* per l'avvio dell'applicazione, la definizione dell'interfaccia grafica (file *FinancialPortfolio.fxml*) e i metodi di interazione dell'utente con l'interfaccia grafica (classe *FinancialPortfolioController*). Di seguito viene riportata la descrizione ad alto livello delle principali strutture dati e degli algoritmi utilizzati.

Tutte le informazioni relative alle obbligazioni a tasso fisso vengono importate all'interno dell'applicazione tramite metodi DAO. All'interno della classe *FinancialPortfolioDAO* sono presenti il metodo *listRatings()* per ottenere una lista dei valori di rating definiti da Standard & Poor's, il metodo *getRatingOfValue()* per ottenere il rating Standard & Poor's corrispondente ad un determinato valore numerico e il metodo *getValueRating()* per ottenere il valore numerico corrispondente ad un determinato rating Standard & Poor's. Infine il metodo *listBonds()* restituisce una lista di titoli obbligazionari con rating maggiore o uguale del primo parametro e con scadenza compresa tra le due date passate come parametri.

```
public List<Bond> listBonds(int sp_rating_value, LocalDate maturity, LocalDate today){
    String sql="SELECT DISTINCT bond.issuer_name, bond.symbol, bond.callable, bond.sub_product_type, bond.coupon, bond.maturity, sp_rating.value, bond.price, bond.yield "+
        "FROM bond, sp_rating " +
        "WHERE bond.sp_rating=sp_rating.rating AND sp_rating.value>=? AND bond.maturity<=? AND bond.maturity>=? "+
        "ORDER BY bond.yield DESC "+
        "LIMIT 40 ";
    List<Bond> result=new ArrayList<>();
    Connection conn=DBConnect.getConnection();
```

Al fine di ottimizzare l'algoritmo ricorsivo riducendone i tempi di esecuzione, la query SQL seleziona soltanto un numero limitato di titoli obbligazionari (clausola LIMIT) tra quelli migliori, cioè quelli con rendimento massimo (clausola di ordinamento decrescente in base all'attributo *yield*). È possibile variare il parametro della clausola LIMIT a piacere, ovviamente all'aumentare del numero di titoli selezionati aumenterà il tempo di esecuzione dell'algoritmo ricorsivo con conseguente peggioramento delle performance. L'introduzione di questa semplificazione è stata necessaria a causa delle considerevoli dimensioni del database utilizzato.

I dati che vengono estratti dal database sono memorizzati tramite la classe *Bond*, che identifica il singolo titolo obbligazionario. La classe *Investment* rappresenta l'investimento di una somma di denaro in un titolo obbligazionario ad una determinata data. In particolare l'attributo *date* della classe *Investment* rappresenta la data in cui avviene l'investimento e permette di gestire, a livello di algoritmo, il calcolo degli interessi maturati periodicamente e il reinvestimento di questa nuova liquidità nel corso del tempo fino al termine della durata del portafoglio finanziario. La classe *Portfolio* identifica un portafoglio finanziario e contiene una lista di investimenti (classe *Investment*), che rappresenta la struttura dati utilizzata nell'algoritmo per la memorizzazione del portafoglio ottimo. La gestione degli investimenti all'interno del portafoglio avviene tramite l'utilizzo della struttura dati *ArrayList*, che permette di avere una bassa complessità computazionale e di ricercare facilmente, iterando su di essa, uno specifico elemento. All'interno della classe *Portfolio* sono presenti il metodo *getTotAmountBond()* che restituisce il capitale investito in un titolo ad una determinata data, il metodo *getTotAmountInvested()* che fornisce il capitale complessivo investito ad una determinata data, il metodo *getTotEarning()* che calcola il rendimento complessivo del

portafoglio e il metodo *getTotRisk()* che calcola il rischio complessivo del portafoglio in base al valore numerico di rating delle obbligazioni. Di seguito è riportato il codice di questi due ultimi metodi che utilizzano formule semplificate, in quanto non si hanno a disposizione dati storici con cui poter calcolare la media e la varianza dei titoli obbligazionari.

```
public float getTotEarning(int durata) {
    float earning=0;
    for(Investment i : investments) {
        earning+=i.getAmount()*(i.getBond().getCoupon()/100)*(i.getDate().until(i.getBond().getMaturity(),ChronoUnit.DAYS)/365)+(i.getAmount()*(1-(i.getBond().getPrice()/100)));
    }
    return earning;
}
```

```
public int getTotRisk() {
    int risk=0;
    for(Investment i : investments) {
        risk+=(i.getAmount()/1000)*(24-i.getBond().getSp_rating());
    }
    return risk;
}
```

Nella classe *Model* è presente l'algoritmo ricorsivo utilizzato per la risoluzione del problema di ottimizzazione del portafoglio finanziario. La complessità dell'algoritmo è stata ridotta con alcune ipotesi semplificative ed introducendo alcuni vincoli aggiuntivi al problema. Innanzitutto l'algoritmo considera soltanto titoli obbligazionari a tasso fisso e senza opzione callable. Inoltre nell'algoritmo vengono definite una percentuale minima *MIN_SINGLE_INVESTMENT* e una percentuale massima *MAX_SINGLE_INVESTMENT* di possesso per ogni titolo in portafoglio, che permettono di ridurre il numero di iterazioni della ricorsione. Queste percentuali sono definite nella classe *Model* come costanti e il loro valore può essere facilmente cambiato a piacere.

L'algoritmo ricorsivo è fatto partire tramite il metodo *searchPortfolioPlus()*, che richiama il metodo *searchPortfolio()* per allocare la liquidità disponibile ad una certa data, passata come parametro di input al metodo, in base ai vincoli del problema. Infatti il metodo *searchPortfolio()* ritorna il valore della liquidità non investita, cioè la liquidità disponibile alla data specificata in input al metodo.

```
public Portfolio searchPortfolio(double budget, int minRating, float rendimento, int durata, String obbiettivo, LocalDate data) {
    if(MIN_SINGLE_INVESTMENT*budget>1000) {
        MIN_INVESTMENT_AMOUNT=((int)(MIN_SINGLE_INVESTMENT*budget/1000))*1000;
    }
    else {
        MIN_INVESTMENT_AMOUNT=5000;
    }
    optimalSolution=new Portfolio();
    liquidity=0;
    if(budget<MIN_INVESTMENT_AMOUNT) {
        liquidity=budget-optimalSolution.getTotAmountInvested(data);
        return optimalSolution;
    }
    Portfolio partialSolution=new Portfolio();
    for(Bond b : this.getBonds(minRating, LocalDate.now().plusYears(durata), data)) {
        if(data.until(b.getMaturity(),ChronoUnit.YEARS)>0 && data.until(b.getMaturity(),ChronoUnit.YEARS)<=durata && b.getSp_rating()>=minRating &&
            (b.getPrice()/100)*MIN_INVESTMENT_AMOUNT<=budget) {
            Investment itemp=new Investment(codiceInvestment++,b,MIN_INVESTMENT_AMOUNT, data);
            partialSolution.addInvestment(itemp);
            recursive(1, partialSolution, budget, minRating, rendimento, durata, obbiettivo, data);
            partialSolution.removeInvestment(itemp);
        }
    }
    if((optimalSolution.getTotEarning(durata)/budget)*100<rendimento) {
        optimalSolution.clearPortfolio();
    }
    liquidity=budget-optimalSolution.getTotAmountInvested(data);
    return optimalSolution;
}
```

Nel metodo *searchPortfolioPlus()* si effettua un ciclo che ogni semestre (per semplificazione si utilizza come base temporale un semestre) cerca di allocare la nuova liquidità disponibile al momento attuale in titoli, che vengono così sommati al portafoglio ottimo. Questa nuova liquidità disponibile (rappresentata dalla variabile temporanea *amount* nel codice) è costituita dalla liquidità ritornata precedentemente dal metodo *searchPortfolio()*, dagli interessi maturati nel semestre dei titoli presenti in portafoglio e dal rimborso degli eventuali titoli scaduti nel corso del semestre.

```
public Portfolio searchPortfolioPlus(double budget, int minRating, float rendimento, int durata, String obbiettivo, LocalDate data) {
    if(bonds!=null) {
        bonds.clear();
    }
    dateBonds=LocalDate.now().minusYears(1);
    budgetiniziale=budget;
    this.optimalSolutionPlus= new Portfolio();
    this.searchPortfolio(budget, minRating, rendimento, durata, obbiettivo, data);
    for(Investment i : optimalSolution.getInvestments()) {
        int k=0;
        for(k=0; k<optimalSolutionPlus.getInvestments().size(); k++) {
            if(optimalSolutionPlus.getInvestments().get(k).getBond().equals(i.getBond())) {
                break;
            }
        }
        if(k==optimalSolutionPlus.getInvestments().size()) {
            optimalSolutionPlus.addInvestment(i);
        }
        else {
            optimalSolutionPlus.getInvestments().get(k).setAmount(optimalSolutionPlus.getInvestments().get(k).getAmount()+i.getAmount());
        }
    }
    LocalDate l;
    for(l=data.plusMonths(6); l.isBefore(data.plusYears(durata)); l=l.plusMonths(6)) {
        double amount=liquidity;
        for(Investment itemp : optimalSolutionPlus.getInvestments()) {
            if(itemp.getBond().getMaturity().isAfter(l.minusMonths(6)) && itemp.getBond().getMaturity().isBefore(l)) {
                amount+=itemp.getAmount()+itemp.getAmount()*(itemp.getBond().getCoupon()/100)*((l.minusMonths(6)).until(itemp.getBond().getMaturity(),ChronoUnit.DAYS))/365;
            }
            else {
                if(itemp.getBond().getMaturity().isAfter(l)) {
                    amount+=itemp.getAmount()*(itemp.getBond().getCoupon()/100)*((l.minusMonths(6)).until(l,ChronoUnit.DAYS))/365;
                }
            }
        }
        if(amount>=1000) {
            int duration=(int) (durata-(data.until(l, ChronoUnit.DAYS))/365);
            this.searchPortfolio(amount, minRating, rendimento, duration, obbiettivo, l);
            for(Investment i : optimalSolution.getInvestments()) {
                optimalSolutionPlus.addInvestment(i);
            }
        }
        else {
            liquidity=amount;
        }
    }
    return optimalSolutionPlus;
}
```

Il metodo *recursive()* contiene l'algoritmo ricorsivo per identificare l'allocazione ottima del capitale nel rispetto dei vincoli del problema.

```
public void recursive(int step, Portfolio partialSolution, double budget, int minRating, float rendimento, int durata, String obiettivo, LocalDate data) {
    if(budget-partialSolution.getTotAmountInvested(data)<MIN_INVESTMENT_AMOUNT) {
        if(obiettivo.equals("massimizzare il ricavo")) {
            if(partialSolution.getTotEarning(durata)>optimalSolution.getTotEarning(durata)) {
                optimalSolution.clearPortfolio();
                for(Investment i : partialSolution.getInvestments()) {
                    optimalSolution.addInvestment(i);
                }
            }
        }
        else {
            if(optimalSolution.getInvestments().isEmpty()) {
                optimalSolution.clearPortfolio();
                for(Investment i : partialSolution.getInvestments()) {
                    optimalSolution.addInvestment(i);
                }
            }
            if(partialSolution.getTotRisk(<optimalSolution.getTotRisk()) {
                optimalSolution.clearPortfolio();
                for(Investment i : partialSolution.getInvestments()) {
                    optimalSolution.addInvestment(i);
                }
            }
        }
    }
    return;
}

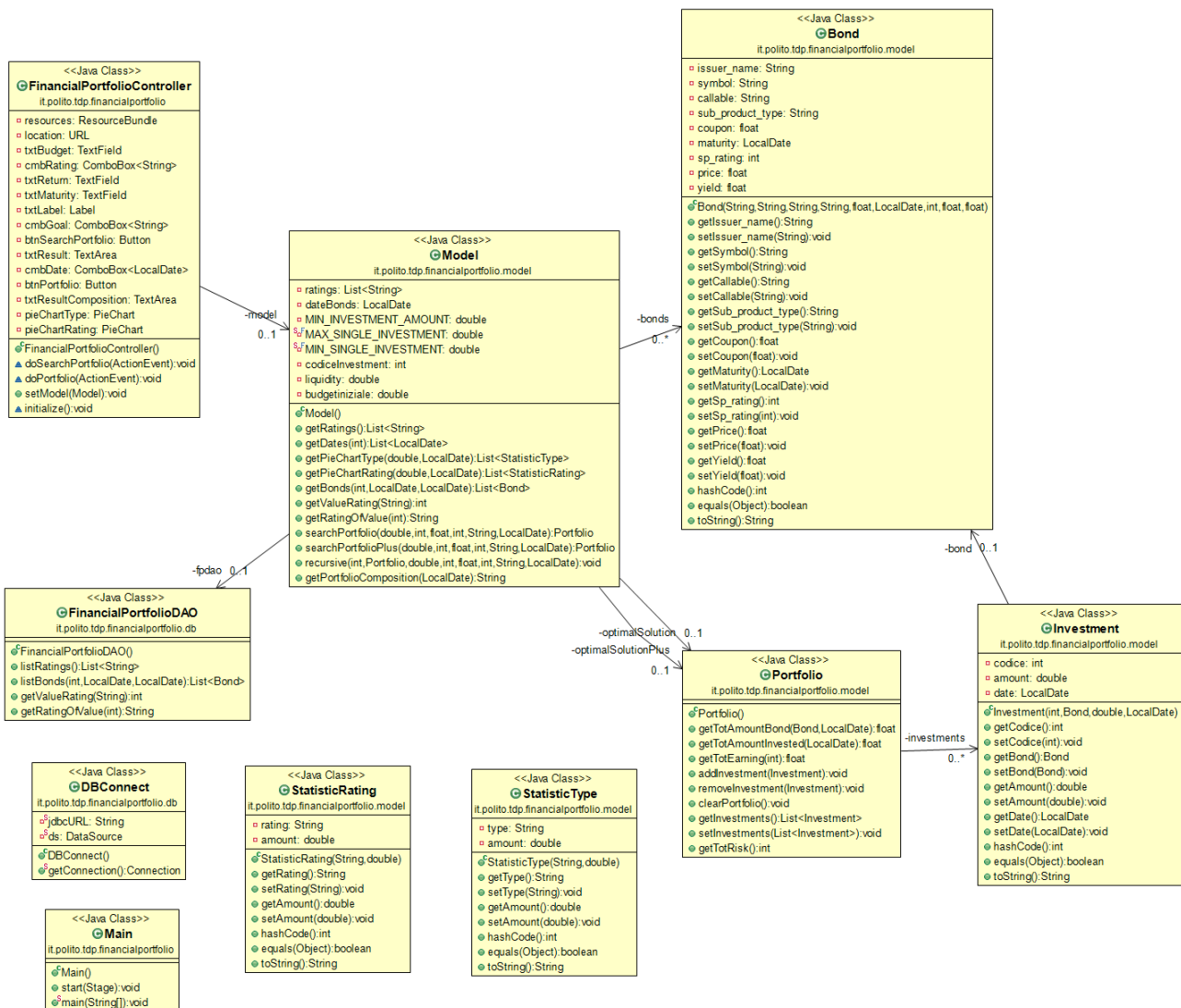
for(Bond b : this.getBonds(minRating, LocalDate.now().plusYears(durata), data)) {
    if(data.until(b.getMaturity(),ChronoUnit.YEARS)>0 && data.until(b.getMaturity(),ChronoUnit.YEARS)<=durata && b.getSp_rating(>=minRating
        && (b.getPrice()/100)*MIN_INVESTMENT_AMOUNT<=budget) {
        Investment itemp;
        if(((partialSolution.getTotAmountBond(b, data)+MIN_INVESTMENT_AMOUNT)*(b.getPrice()/100)+optimalSolutionPlus.getTotAmountBond(b, data))<=MAX_SINGLE_INVESTMENT*budgetiniziale
            && budget-partialSolution.getTotAmountInvested(data)>=(b.getPrice()/100)*MIN_INVESTMENT_AMOUNT) {
            itemp=new Investment(codiceInvestment++,b,MIN_INVESTMENT_AMOUNT, data);
            partialSolution.addInvestment(itemp);
            recursive(step+1, partialSolution, budget, minRating, rendimento, durata, obiettivo, data);
            partialSolution.removeInvestment(itemp);
        }
    }
}
```

Nella classe *Model* sono anche presenti il metodo *getRatings()* utilizzato per popolare la Combo Box del rating minimo, il metodo *getDates()* utilizzato per popolare la Combo Box delle date di visualizzazione del portafoglio, il metodo *getPieChartType()* che crea il grafico relativo alla diversificazione del portafoglio per tipologia di titolo, il metodo *getPieChartRating()* che crea il grafico relativo alla diversificazione del portafoglio per rating ed alcuni metodi utilizzati per il caricamento dei dati del database. Infine il metodo *getPortfolioComposition()* restituisce la composizione del portafoglio, cioè l'elenco degli investimenti presenti in portafoglio nella data selezionata.

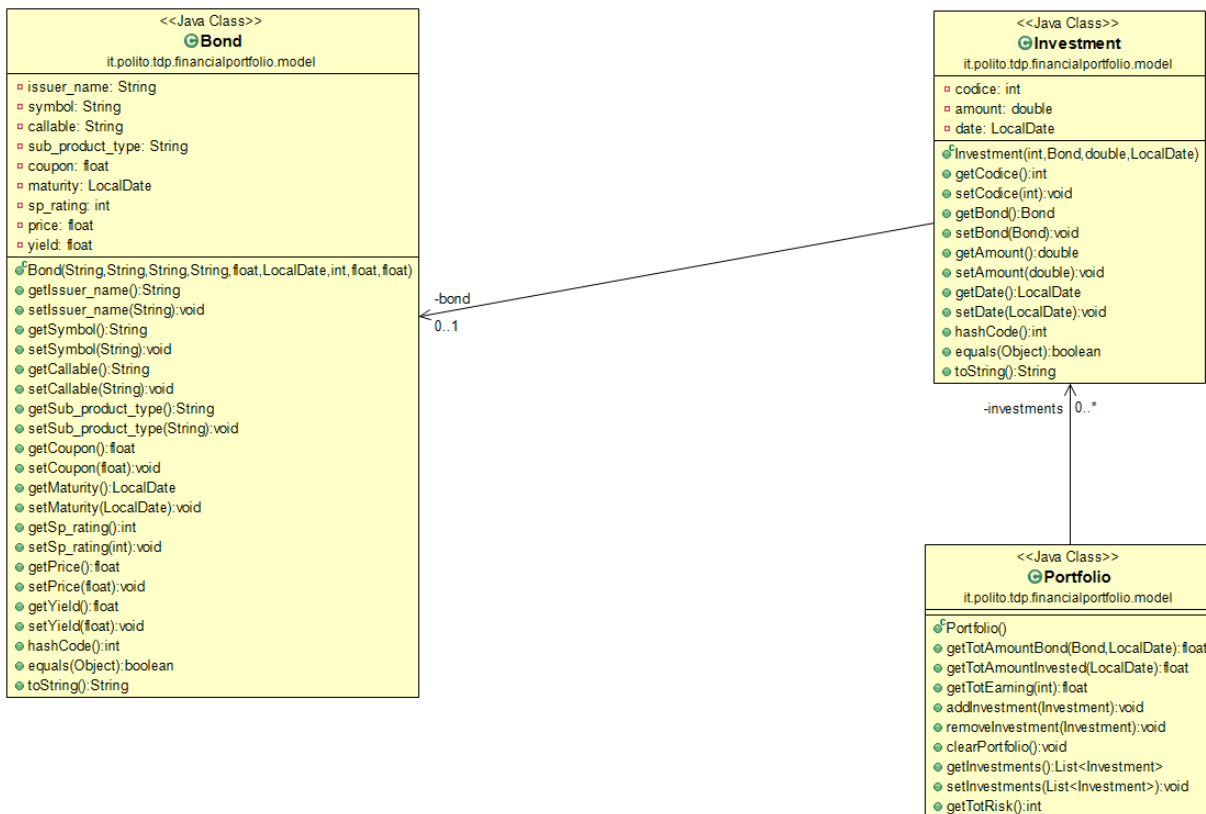
Le classi *StatisticRating* e *StatisticType* vengono utilizzate per la realizzazione dei grafici Pie Chart relativi alla rappresentazione grafica della diversificazione del portafoglio finanziario. La prima classe permette di raggruppare gli investimenti per rating (utilizzata nel metodo *getPieChartRating()*), mentre la seconda classe permette di raggruppare gli investimenti per tipologia (utilizzata nel metodo *getPieChartType()*).

Diagramma delle classi delle parti principali dell'applicazione

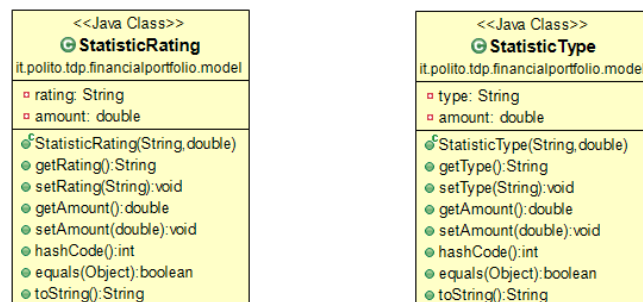
L'applicazione è strutturata in base al pattern architetturale MVC (Model-View-Controller). La classe *Model* gestisce il comportamento dell'applicazione, la classe *FinancialPortfolioController* si occupa dell'interazione con l'utente (scambio di dati in input e in output) e la classe *FinancialPortfolioDAO* controlla l'interazione con il database. In particolare la classe *Model* è in grado di interfacciarsi con entrambe le altre due classi.



All'interno della classe *Bond* sono presenti le informazioni relative a *issuer_name*, *symbol*, *callable*, *sub_product_type*, *coupon*, *maturity*, *sp_rating* (viene memorizzato il valore numerico associato al rating), *price* e *yield*. La classe *Investment* contiene le informazioni relative a *codice* (codice univoco dell'investimento), *bond* (obbligazione), *amount* (valore monetario dell'investimento) e *date* (data dell'investimento). La classe *Portfolio* identifica un portafoglio finanziario e rappresenta la struttura dati utilizzata dall'algoritmo per la memorizzazione del portafoglio ottimo. All'interno della classe *Portfolio* è presente una lista di investimenti (classe *Investment*).

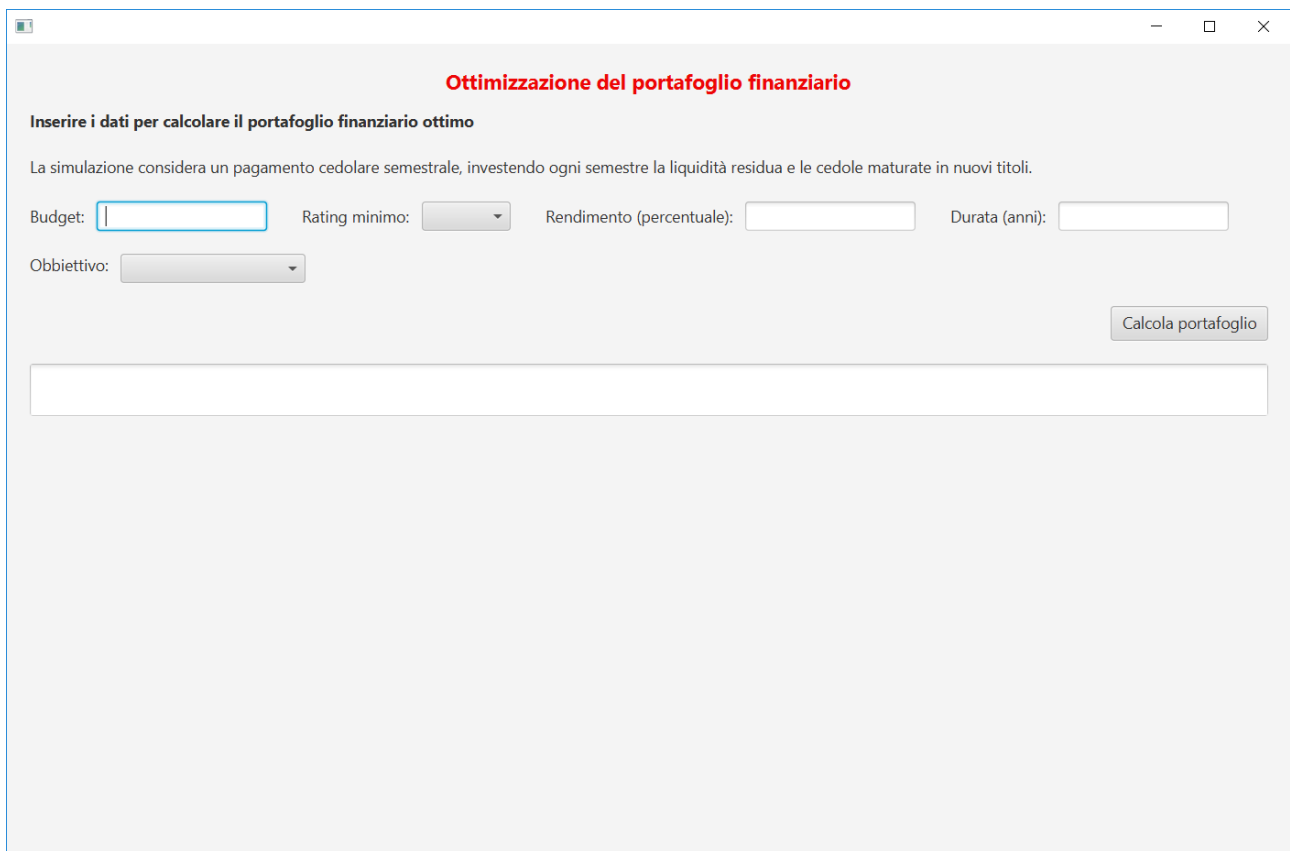


Le classi *StatisticRating* e *StatisticType* sono utilizzate per la creazione dei grafici Pie Chart per la visualizzazione della composizione del portafoglio finanziario. Il primo Pie Chart mostra la composizione del portafoglio ottimo in base alla tipologia di obbligazioni (identificata dall'attributo *sub_product_type*), mentre il secondo Pie Chart mostra la composizione del portafoglio ottimo in base alla diversificazione del rischio (identificato dall'attributo *sp_rating*).



Alcune videate dell'applicazione

L'applicazione presenta la seguente interfaccia grafica iniziale.



The screenshot shows a web application window titled "Ottimizzazione del portafoglio finanziario". The interface is designed for inputting financial data to calculate an optimal portfolio. At the top, there is a red heading. Below it, a sub-heading instructs the user to enter data. A descriptive paragraph explains the simulation's assumptions: semi-annual coupon payments, reinvestment of residual liquidity and matured coupons into new titles. The input section includes a "Budget:" text field, a "Rating minimo:" dropdown menu, a "Rendimento (percentuale):" text field, and a "Durata (anni):" text field. Below these, an "Obiettivo:" dropdown menu is present. A "Calcola portafoglio" button is located on the right side of the input area. At the bottom of the form, there is a large, empty text area for displaying the results.

Nella zona superiore dell'applicazione è possibile inserire o selezionare i dati per il calcolo del portafoglio finanziario ottimo. Una serie di Combo Box e Text Field permette all'utente di definire i vincoli del problema di ottimizzazione del portafoglio: budget, rating minimo, rendimento, durata e obiettivo. Inoltre viene specificato il meccanismo di calcolo del portafoglio, cioè le ipotesi utilizzate nella simulazione: pagamento cedolare semestrale, investimento semestrale della liquidità residua e delle cedole maturate.

Nel corpo centrale sono presenti una Text Area ed un Button *"Calcola portafoglio"*, che permettono di eseguire il calcolo del portafoglio ottimo e mostrare il relativo risultato all'utente.

— □ ×

Ottimizzazione del portafoglio finanziario

Inserire i dati per calcolare il portafoglio finanziario ottimo

La simulazione considera un pagamento cedolare semestrale, investendo ogni semestre la liquidità residua e le cedole maturate in nuovi titoli obbligazionari.

Budget: Rating minimo: Rendimento (percentuale): Durata (anni):

Obiettivo:

Calcola portafoglio

Il portafoglio ottimo, calcolato in relazione ai dati inseriti dall'utente, ha un valore pari a 53998€ alla fine del periodo.

Selezionare una data per visualizzare la relativa composizione del portafoglio:

Visualizza portafoglio

Nella Text Area, dopo aver definito tutti i parametri di input e aver cliccato sul pulsante “*Calcola portafoglio*”, compare l’indicazione relativa al valore della liquidità ottenuta al termine del periodo in base agli investimenti presenti in portafoglio.

Nella zona inferiore dell’applicazione, dopo aver calcolato il portafoglio ottimo, è possibile visualizzare la composizione del portafoglio di titoli obbligazionari. Nella Combo Box è possibile selezionare la data a cui si vuole visualizzare la composizione del portafoglio. Di seguito sono presenti una Text Area ed un Button “*Visualizza portafoglio*”, che permettono all’utente di visualizzare i titoli presenti in portafoglio alla data selezionata.

Ottimizzazione del portafoglio finanziario

Inserire i dati per calcolare il portafoglio finanziario ottimo

La simulazione considera un pagamento cedolare semestrale, investendo ogni semestre la liquidità residua e le cedole maturate in nuovi titoli obbligazionari.

Budget:
 Rating minimo:
 Rendimento (percentuale):
 Durata (anni):

Obiettivo:

Il portafoglio ottimo, calcolato in relazione ai dati inseriti dall'utente, ha un valore pari a 53998€ alla fine del periodo.

Selezionare una data per visualizzare la relativa composizione del portafoglio:

La composizione del portafoglio ottimo in data 2020-08-06 è la seguente:

2018-02-06 OB.RUSSIAN FEDERATION 7.05% RUB 19GEN28 RU000A0JTK38 10000€

2018-02-06 OB.BEI 8% MXN 11GEN27 XS1547492410 5000€

2018-02-06 OB.PETROBRAS GL FIN 6% USD 27GEN28 MWC USN6945AAK36 5000€

Nella Text Area, dopo aver selezionato la data e aver cliccato sul pulsante “*Visualizza portafoglio*”, compare l’elenco dei titoli presenti in portafoglio con le relative quote di capitale investite. Questo risultato ottenuto rappresenta la composizione del portafoglio finanziario nella data selezionata dall’utente. Nella zona inferiore dell’applicazione compaiono inoltre due Pie Chart, che mostrano la diversificazione del portafoglio alla data selezionata dall’utente in base al rating minimo e alla tipologia di titolo obbligazionario.

Di seguito è riportato il link al video dimostrativo del software:

https://youtu.be/3_0QfjALV9A

Tabelle con risultati sperimentali ottenuti

Nelle tabelle seguenti sono riportati i tempi di esecuzione dell'algoritmo ricorsivo e il valore della liquidità ottenuta dal portafoglio finanziario al variare dei parametri di input forniti all'applicazione.

Variazione del budget mantenendo costanti gli altri parametri.

| Budget [€] | Rating minimo | Rendimento [%] | Durata [anni] | Obbiettivo | Liquidità portafoglio [€] | Tempo di elaborazione dell'algoritmo [secondi] |
|------------|---------------|----------------|---------------|------------------------|---------------------------|--|
| 10000 | A | 2 | 10 | massimizzare il ricavo | 16134 | 4,620725483 |
| 20000 | A | 2 | 10 | massimizzare il ricavo | 33276 | 0,399278064 |
| 30000 | A | 2 | 10 | massimizzare il ricavo | 52157 | 0,597103962 |
| 40000 | A | 2 | 10 | massimizzare il ricavo | 68961 | 1,318851824 |
| 50000 | A | 2 | 10 | massimizzare il ricavo | 88689 | 2,647759546 |
| 60000 | A | 2 | 10 | massimizzare il ricavo | 103809 | 8,718260259 |
| 70000 | A | 2 | 10 | massimizzare il ricavo | 124281 | 389,1921793 |

Variazione del rating minimo mantenendo costanti gli altri parametri.

| Budget [€] | Rating minimo | Rendimento [%] | Durata [anni] | Obbiettivo | Liquidità portafoglio [€] | Tempo di elaborazione dell'algoritmo [secondi] |
|------------|---------------|----------------|---------------|------------------------|---------------------------|--|
| 10000 | AAA | 2 | 10 | massimizzare il ricavo | 16105 | 3,100268997 |
| 10000 | AA | 2 | 10 | massimizzare il ricavo | 16105 | 2,52289125 |
| 10000 | A | 2 | 10 | massimizzare il ricavo | 16134 | 4,209154288 |
| 10000 | BBB | 2 | 10 | massimizzare il ricavo | 16134 | 1,181790923 |

Variazione della durata mantenendo costanti gli altri parametri.

| Budget [€] | Rating minimo | Rendimento [%] | Durata [anni] | Obbiettivo | Liquidità portafoglio [€] | Tempo di elaborazione dell'algoritmo [secondi] |
|------------|---------------|----------------|---------------|------------------------|---------------------------|--|
| 10000 | A | 2 | 2 | massimizzare il ricavo | 11303 | 0,704169949 |
| 10000 | A | 2 | 3 | massimizzare il ricavo | 11995 | 1,467894569 |
| 10000 | A | 2 | 4 | massimizzare il ricavo | 12611 | 1,041759539 |
| 10000 | A | 2 | 5 | massimizzare il ricavo | 14211 | 1,813232098 |
| 10000 | A | 2 | 6 | massimizzare il ricavo | 14371 | 2,2952742 |
| 10000 | A | 2 | 7 | massimizzare il ricavo | 14541 | 2,673065208 |
| 10000 | A | 2 | 8 | massimizzare il ricavo | 14593 | 2,799069552 |
| 10000 | A | 2 | 9 | massimizzare il ricavo | 16134 | 3,616795865 |
| 10000 | A | 2 | 10 | massimizzare il ricavo | 16134 | 3,630171449 |

Variazione dell'obbiettivo mantenendo costanti gli altri parametri.

| Budget [€] | Rating minimo | Rendimento [%] | Durata [anni] | Obbiettivo | Liquidità portafoglio [€] | Tempo di elaborazione dell'algoritmo [secondi] |
|------------|---------------|----------------|---------------|------------------------|---------------------------|--|
| 10000 | A | 2 | 10 | massimizzare il ricavo | 16134 | 3,630171449 |
| 10000 | A | 2 | 10 | minimizzare il rischio | 13496 | 44,82339465 |

In particolare si osserva che l'applicazione è in grado di calcolare il portafoglio finanziario ottimo in tempi ragionevoli, con un tempo di esecuzione dell'algoritmo ricorsivo inferiore al minuto, solo per alcuni casi caratterizzati da un basso valore di budget, un alto valore di rating minimo e un valore non troppo elevato di durata dell'investimento. Negli altri casi la dimensione dell'insieme di titoli obbligazionari estratto dal database è particolarmente elevata e di conseguenza l'algoritmo ricorsivo evidenzia un peggioramento delle performance con conseguente aumento del tempo di esecuzione. La ricerca di un'eventuale soluzione per un'ulteriore ottimizzazione dell'algoritmo potrebbe portare ad una diminuzione del tempo di esecuzione migliorandone le performance, ma solo entro certi limiti. Infatti la ricorsione, pur permettendo di risolvere in poche righe di codice problemi complessi, comporta sempre un enorme svantaggio in termini di prestazioni.

Valutazione risultati ottenuti e conclusioni

L'obiettivo dell'applicazione è fornire uno strumento utile per ottenere dati e risultati su cui basare le decisioni di investimento di un capitale. La soluzione realizzata per il problema di ottimizzazione del portafoglio finanziario si basa sull'utilizzo di un algoritmo ricorsivo, che si è rilevato essere non particolarmente appropriato per un database di grandi dimensioni. Infatti la ricorsione, pur permettendo di risolvere facilmente il problema complesso di ottimizzazione del portafoglio, comporta un considerevole svantaggio in termini di prestazioni. Un algoritmo ricorsivo è caratterizzato sempre dall'impiego di un maggior numero di risorse: esso genera una quantità enorme di overhead, occupando lo stack per un numero di istanze uguale al numero di chiamate della funzione che è necessario effettuare per risolvere il problema. Per far fronte a questo problema di performance sono state introdotte alcune semplificazioni concettuali, che hanno permesso di ridurre il tempo di esecuzione dell'algoritmo introducendo ulteriori vincoli nel problema. Il software realizzato rappresenta un valido strumento per gestire ed investire la liquidità posseduta. L'applicazione realizzata rappresenta una soluzione iniziale del problema di ottimizzazione del portafoglio finanziario, che può essere migliorata tramite un'ulteriore e più approfondita analisi di studio della complessità del mercato obbligazionario.

L'applicazione presenta come punti di forza la semplicità e la versatilità d'utilizzo, quest'ultima caratterizzata dalla possibilità di considerare differenti scenari variando semplicemente i parametri di input dell'applicazione. Inoltre la rappresentazione grafica dei risultati ottenuti dall'algoritmo permette un'immediata comprensione e una maggiore leggibilità da parte di utenti meno esperti finanziariamente, permettendo quindi l'utilizzo dell'applicazione anche a persone non operanti nel settore degli investimenti finanziari.



Quest'opera è distribuita con licenza Creative Commons Attribuzione - Non commerciale – Condividi allo stesso modo 4.0 Internazionale.

Copia della licenza consultabile al sito web <https://creativecommons.org/licenses/by-nc-sa/4.0/>