PSZT - Uczenie Maszynowe

Stawczyk Przemysław 293153, Piotr Zmyślony 268833

Spis treści

1 Opis zagadnienia **2**
- 1.1 Treść zadania . . . 2
- 1.2 Narzędzia . . . 2

2 Opis preprocesingu i modelowania **2**
- 2.1 Opis danych wejściowych . . . 2
- 2.2 Analiza zbioru danych . . . 2
  - 2.2.1 Brakujące dane . . . 2
  - 2.2.2 Zbalansowanie danych . . . 4
- 2.3 Przepływ Danych . . . 4
  - 2.3.1 Wizualizacja Przepływu Danych . . . 4

3 Modele **4**
- 3.1 Parametry Modeli . . . 4

4 Wyniki Eksperymentu **5**
- 4.1 Porównanie klasyfikatorów . . . 5
- 4.2 Metoda interpolowania brakujących danych . . . 6
- 4.3 Parametry dobrane przez hipersiatkę . . . 8
- 4.4 Interpretacja . . . 9

1 Opis zagadnienia

1.1 Treść zadania

Przedstawić wyniki analizy zbioru *Bankruptcy*, opisać procedurę eksperymentalną uczenia maszynowego z wykorzystaniem algorytmów *random forest* i *k-najbliższych sąsiadów* oraz opisać wyniki strojenia parametrów powyższych algorytmów.

1.2 Narzędzia

Skrypty oraz algorytm zostały zaimplementowane w Pythonie 3. Wykorzystano biblioteki: *imblearn.over_sampling.SMOTE, sklearn, numpy, matplotlib, scipy.io, impyute*.

2 Opis preprocesingu i modelowania

2.1 Opis danych wejściowych

Jako dane wejściowe posiadamy 5 plików *.arff*, z których każdy zawiera ekonomiczne wskaźniki z systemu EMIS na temat polskich firm i ich klasyfikację względem tego, czy firmy zbankrutowały po n latach od roku, w którym zostały zebrane dane. Liczba n lat jest różna dla każdego z plików, od 1 do 5, a każda firma opisana jest przez 64 atrybuty, od *X1* do *X64*.

2.2 Analiza zbioru danych

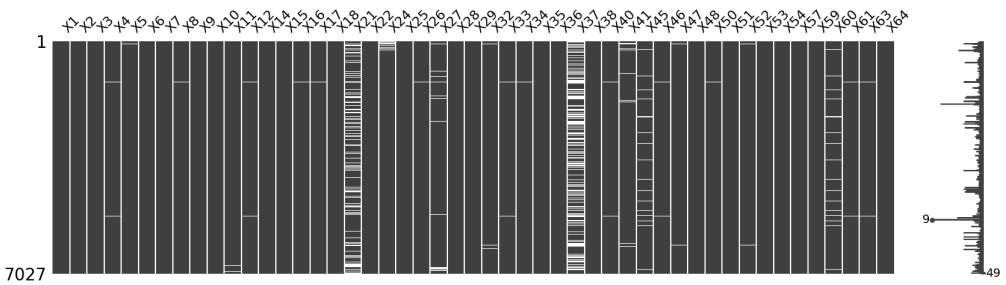
2.2.1 Brakujące dane

Zaczęliśmy od analizy brakujących danych w wierszach. Jak widać w poniższych wynikach w większości zbiorów około połowa wierszy ma brakujące pola.

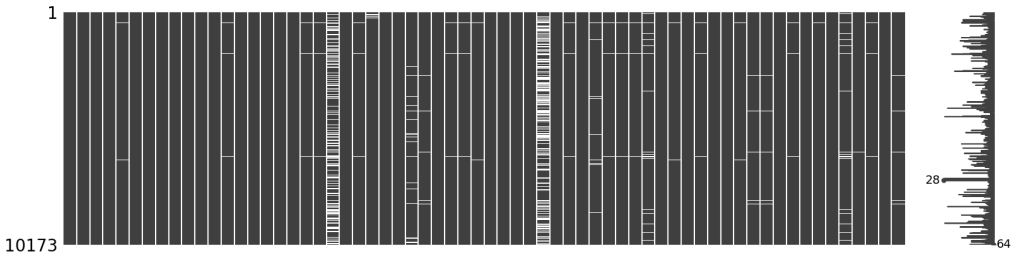
	1 rok	*2 lata*	*3 lata*	*4 lata*	*5 lat*
długość	7027	10173	10503	9792	5910
pełne wiersze	3194	4088	4885	4769	3031
wiersze wybrakowane	3833	6085	5618	5023	2879

Następnie przeprowadziliśmy analizę rozkładu brakujących danych w kolumnach i wierszach korzystając z biblioteki pythona *missingno* [Rys. 1-5]

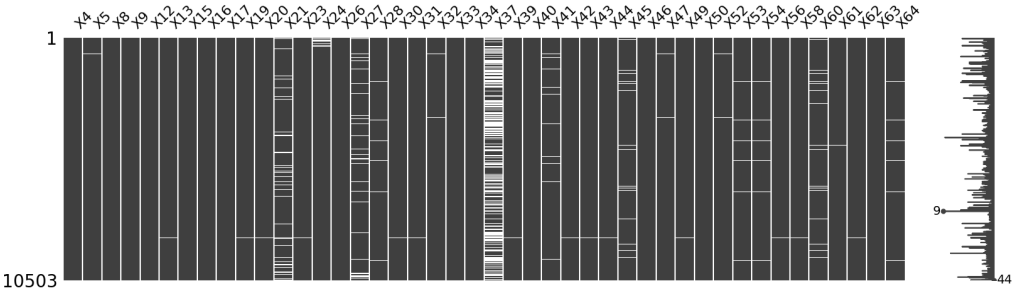
Rysunek 1: 1 rok



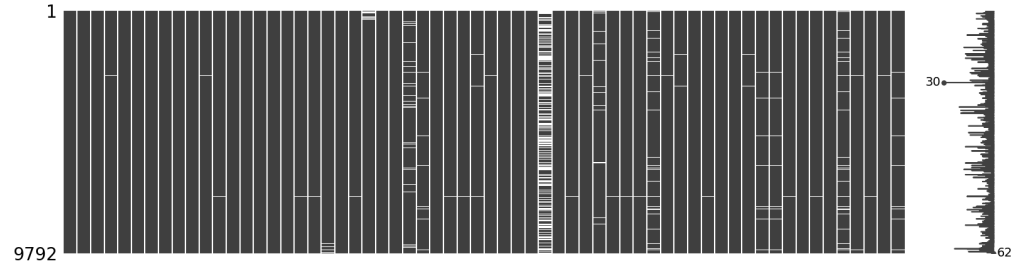
Rysunek 2: 2 lata



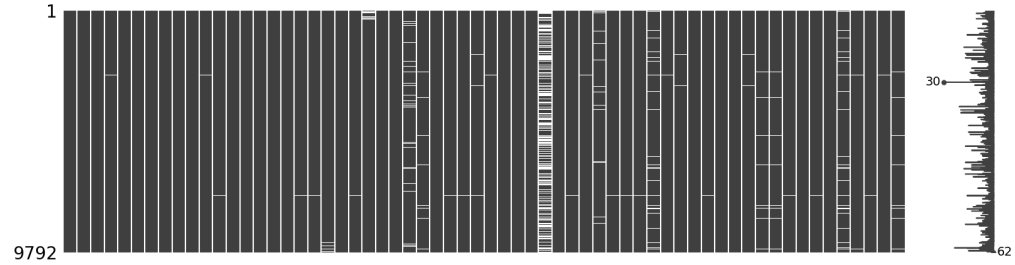
Rysunek 3: 3 lata



Rysunek 4: 4 lata



Rysunek 5: 5 lat



Jak widać większość brakujących danych jest w kolumnie *X37*. Kolumna *X21* ma brakujące w niektórych ale nie wszystkich latach.
Trudno nam było ocenić jaki charakter mają braki w tych danych, czy są skorelowane w wartościami w innych kolumnach czy zupełnie losowe. Wierszy z brakującymi danymi jest około połowy lub więcej. Aby nie odrzucać tak dużej liczby krotek zdecydowaliśmy się interpolować brakujące dane.
W tym celu wybraliśmy 4 metody:

1. Wstawianie średniej w danej kolumnie *(Jako punkt odniesienia)*
2. K najbliższych krotek
3. Spodziewanej Maksymalizacji *(Expected Maximalisation)*
4. Algorytm MICE

2.2.2 Zbalansowanie danych

Dokonaliśmy analizy ile z poszczególnych rekordów należy do klas klasyfikacyjnych

Czy zbankrutowano:	*rok 1*	*rok 2*	*rok 3*	*rok 4*	*rok 5*
Nie	6756	9773	10008	9277	5500
Tak	271	400	495	515	410
procent większości	3.857 %	3.932 %	4.713 %	5.259 %	6.937 %

Dane w zbiorach są mocno niezbalansowane dlatego zdecydowaliśmy się na interpolację korzystając z metody *SMOTE (Synthetic Minority Over Sampling Technique)*

2.3 Przepływ Danych

Po powyższej analizie zdecydowaliśmy o następującym przepływie oryginalnych danych do konstrukcji modeli.
Walidacji modeli planujemy dokonać korzystając K-krotnej walidacji krzyżowej.

2.3.1 Wizualizacja Przepływu Danych

3 Modele

Zgodnie z poleceniem wykorzystaliśmy algorytmy tworzenia modeli :

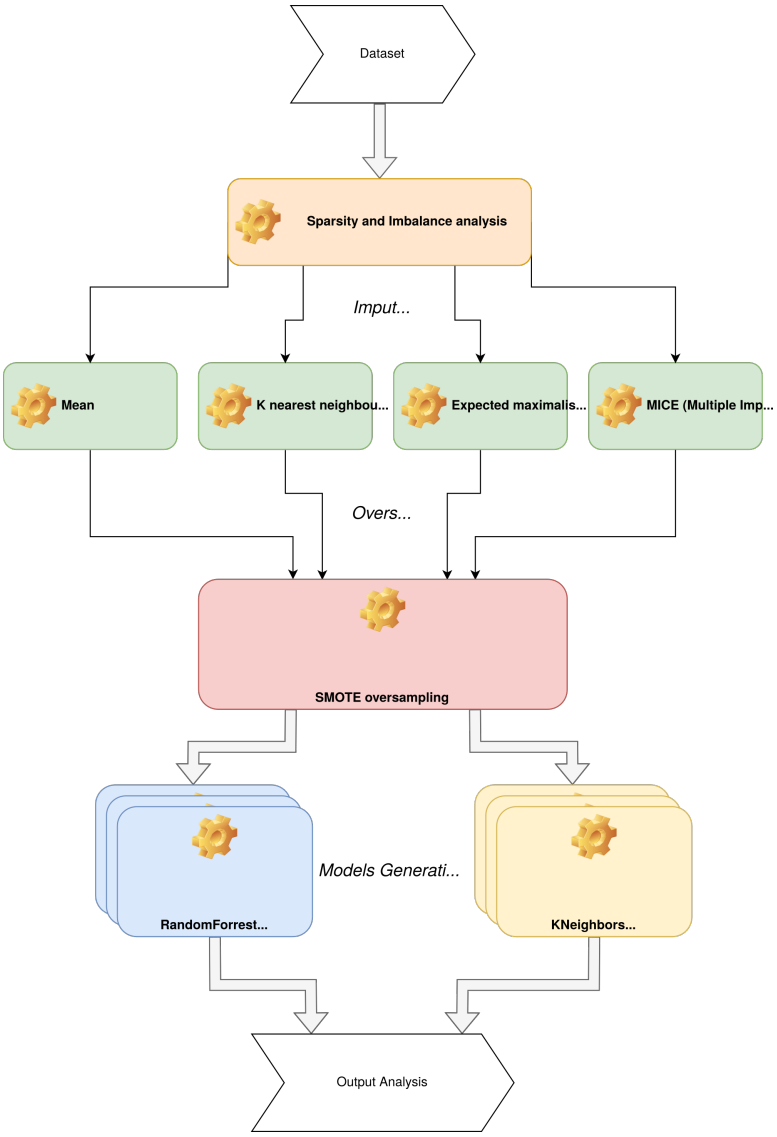
- Las Losowy *[RF - Random Forrest]*
- K Najbliższych sąsiadów *[KNN - K Nearest Neighbors]*

Implementacje wymienionych algorytmów pochodzą z biblioteki *sklearn.*

3.1 Parametry Modeli

W początkowej fazie porównywania modeli chcieliśmy zbadać, który klasyfikator (KNN czy Random Forest) daje ogólnie lepsze wyniki. Z tego początku parametry dobierane były ręcznie, tak aby utworzyć zestawienie pokazujące różnice między tymi dwoma klasyfikatorami (Rysunek

Rysunek 6: Przepływ Danych



??) oraz porównać wpływ zmiany niektórych parametrów na rezultat końcowy dopasowywania modeli.

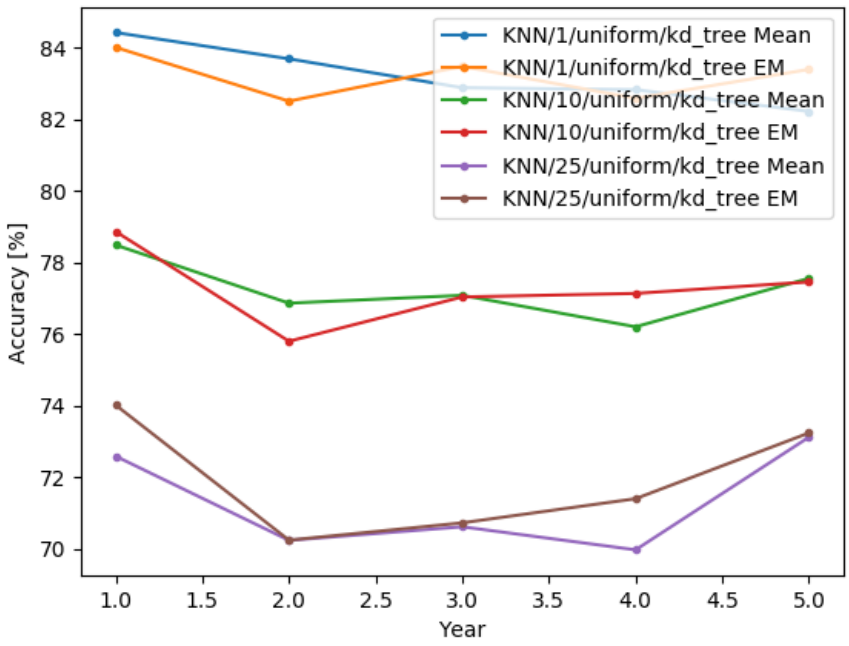
Później, po ustaleniu, że Random Forest jest znacznie klasyfikatorem dla tego problemu, użyliśmy wyszukiwania na hipersiatce, tak aby program sam znalazł hiperparametry dla tego modelu.

4 Wyniki Eksperymentu

4.1 Porównanie klasyfikatorów

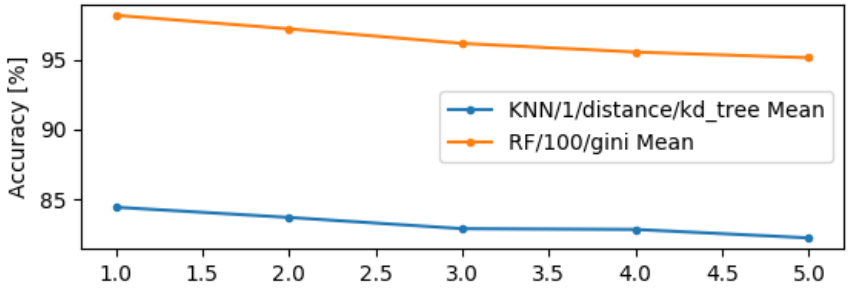
Początkowo, sprawdziliśmy jaka ilość sąsiadów w metodzie KNN daje najlepszy model, co zademonstrowane jest na Rys. 7. Zaobserwowaliśmy, że nasz model najlepiej dopasowywał się jeśli za najbliższego sąsiada uznawany był tylko jeden punkt. Wynik dopasowania każdego klasyfikatora uzyskiwaliśmy dzięki 5-krotnej walidacji krzyżowej.

Rysunek 7: Porównanie liczby K najbliższych sąsiadów



Okazuje się, że klasyfikator Random Forest sprawdza się znacznie lepiej niż najlepszy model klasyfikatora KNN. Porównanie dla tych modeli znajduje się na Rys. 8.

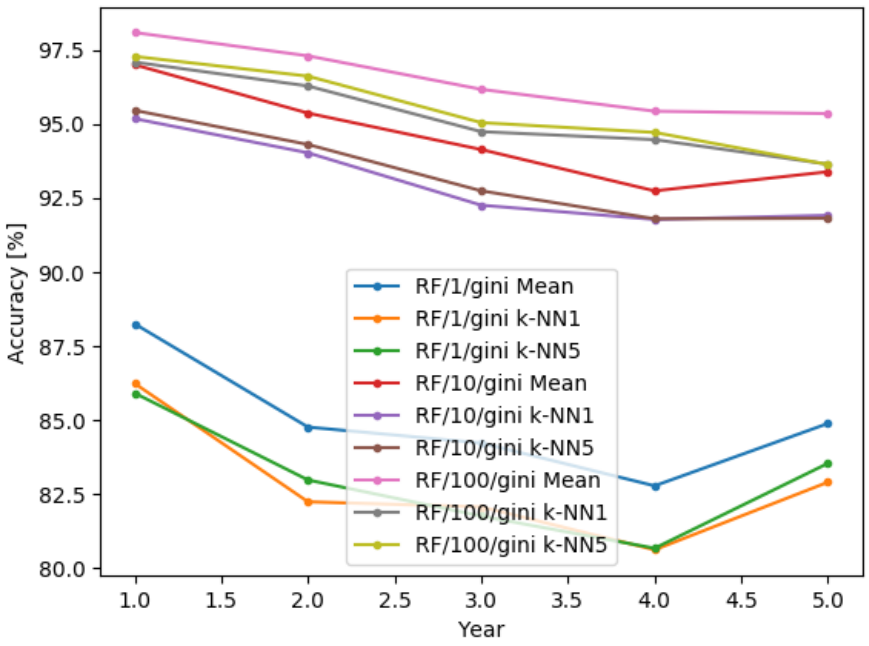
Rysunek 8: Porównanie KSS i RF



4.2 Metoda interpolowania brakujących danych

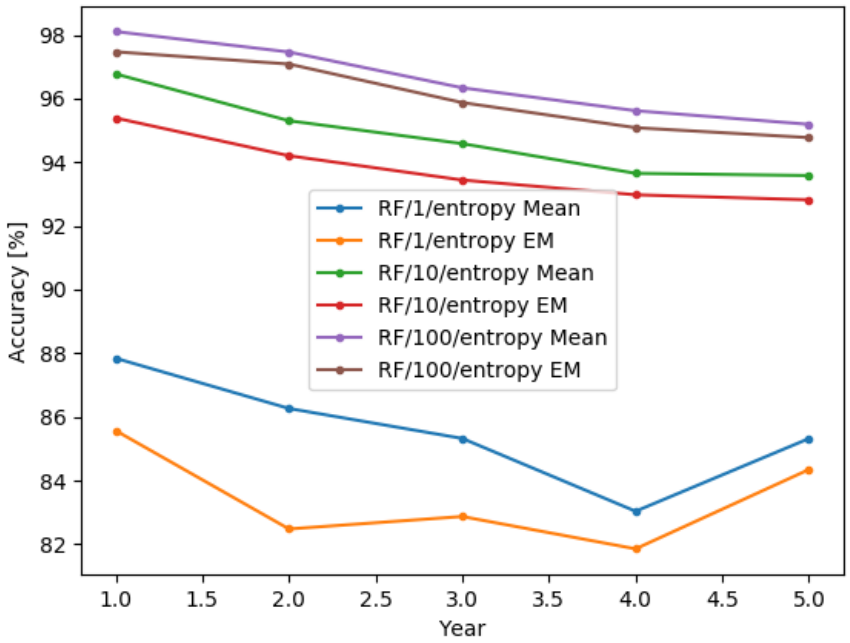
Na Rys. 9 możemy zaobserwować różnicę w precyzji modeli, zależnie od typu wstawiania danych - zwykłe wstawianie wartości uśrednianej osiąga wynik lepszy niż metoda uzupełniania danych poprzez wstawianie średniej k najbliższych sąsiadów.

Rysunek 9: Porównianie metod wstawiania danych



Okazuje się, że metoda Mean jest również lepsza niż EM, co obrazuje Rys. 10. Można stąd wnioskować, że wpływ najbardziej wybrakowanych kolumn *X37* i *X21* na potencjalny, przyszły status bankructwa jest znikomy. Stąd, do liczenia najlepszych hiperparametrów zastosowaliśmy tylko metodę wstawiania wartości średnich.

Rysunek 10: Porównianie metod wstawiania danych



4.3 Parametry dobrane przez hipersiatkę

Hiperparametry dobierane były generując dla każdego z pięciu zestawów danych po 72 klasyfikatory (stosując 5-krotną walidację krzyżową było ich w sumie 360) i później wybierając ten, który dawał najlepszy odsetek prawidłowych klasyfikacji na zbiorach testowych.

Zmieniane hiperparametry modelu klasyfikatora to: *n_estimators* (ilość drzew decyzyjnych), *criterion* (sposób na który drzewa podejmują decyzje), *max_features* (maksymalna liczbę funkcji do rozważenia podczas szukania podziału), *max_depth* (maksymalna głębokość drzewa). Poniżej, w tabeli, jest lista wszystkich testowanych parametrów:

n_estimators	*criterion*	*max_features*	*max_depth*
1, 10, 100	"gini", "entropy"	"auto", "log2", 16	10, 100, 1000, None

Otrzymaliśmy następujące rezultaty dla kolejnych ilości lat:

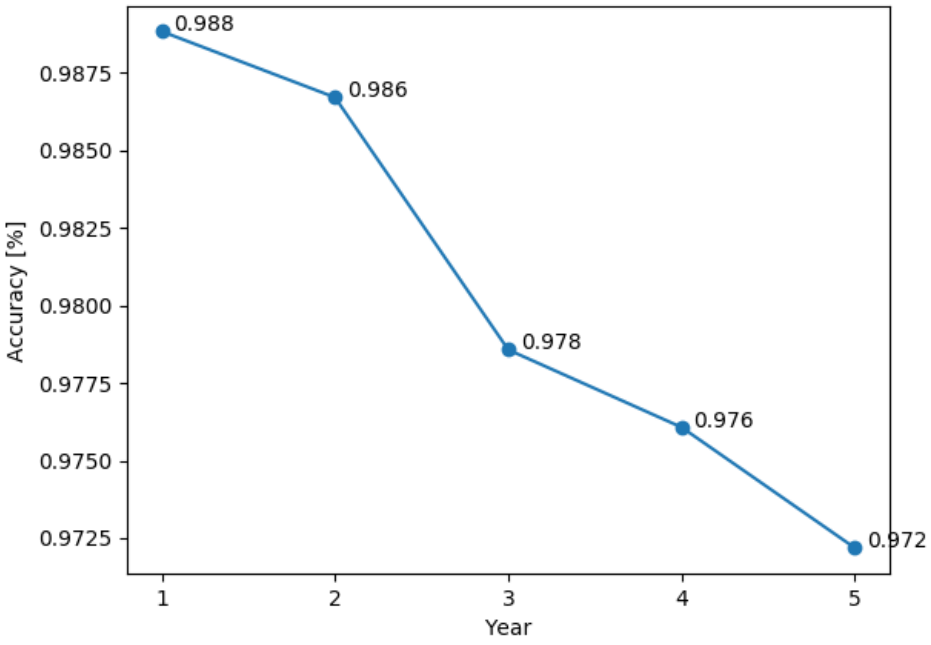
	1 rok	**2 lata**	**3 lata**	**4 lata**	**5 lat**
n_estimators	100	100	100	100	100
criterion	entropy	entropy	gini	entropy	entropy
max_features	log2	auto	log2	16	16
max_depth	100	None	None	100	100

Widzimy, że tak twierdzi teoria - za każdym razem została wybrana największa ilość estymatorów, a reszta parametrów wahała się zależnie od analizowanego zbioru danych.

Efekt działania przeszukiwania hipersiatki - modele o najlepiej dobranych hiperparametrach z podanych zbiorów, są widoczne na Rys. 11.

Prawdopodobnie uzyskanie lepszego wyniku byłoby możliwe analizując więcej hiperparametrów o większym zakresie dostępnych ustawień, jednak jesteśmy ograniczeni przez moc obliczeniową naszych urządzeń.

Rysunek 11: Precyzja modeli dopasowanych hipersiatką



4.4 Interpretacja

Możemy zauważyć spadek poprawności klasyfikacji firm wraz z wydłużaniem się okresu prognozowania, co jest dosyć oczywistym faktem, ponieważ przewidywanie przyszłości jest coraz trudniejsze dla coraz bardziej oddalonych w czasie sytuacji. Stąd też większe prawdopodobieństwo, że firma której się powodzi zbankrutuje w okresie 5 lat, niż nagle, po upływie roku.