Reserves Forecasting Chain Ladder Analysis

– CRISP -DM Framework

DECEMBER 4

Universidad Nacional de Colombia
Authored by: Cristian Ávila

Introduction

Objective:
The purpose of this document is to analyze the Chain Ladder model for predicting reserves. In the insurance business, reserve development is a crucial process for addressing, measuring, and defining a company's solvency for upcoming quarters and years. Solvency, a regulatory concern, directly influences reputation, exposure to sanctions, and financial success. The fundamental concept of reserving involves allocating cash to cover potential claims, making the availability of cash critical for the insurance company's operations.

Scope:
This document will address the Chain Ladder problem within the CRISP-DM framework. The analysis will involve understanding existing data obtained from the CAS website, preparing the data for validation in different model tests (Deterministic, Linear Regression), and ultimately conducting cross-validation analysis to determine the model with the lowest prediction error, indicating better performance.

Significance of Solvency:
A robust reserve system is pivotal for an insurance company, ensuring the availability of funds to meet potential claim payments. The success in defining solvency not only complies with regulatory requirements but also safeguards the company's reputation and minimizes exposure to sanctions. The financial success of the company is intricately linked to the efficiency of its reserve management.

By undertaking a comprehensive analysis of the Chain Ladder model and employing a structured framework, this document aims to enhance the understanding of the reserve prediction process and contribute to the company's overall risk management and financial stability.

Business Understanding

1. Determining Business Objectives:

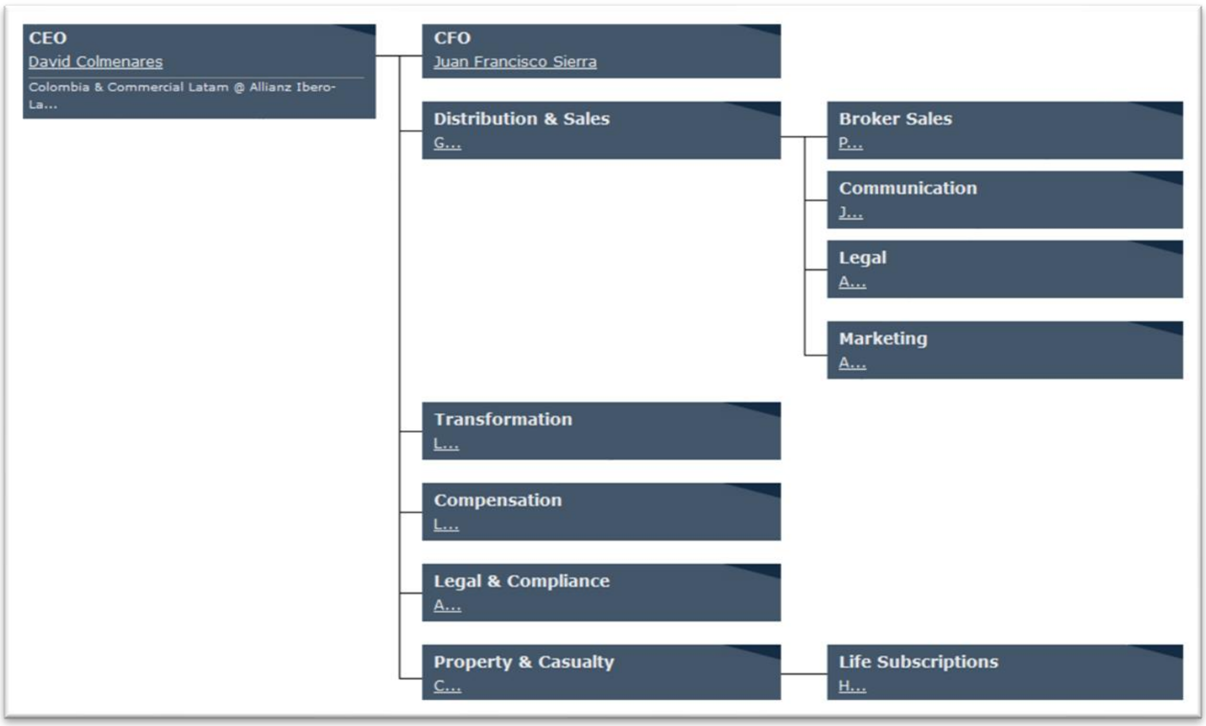
Is required to deploy a model to predict the optimal outcome of the reserve's estimation using Data Sciences techniques or Deep Learning approaches. The expected benefits for the business are:

- Meet compliance regulatory requirements regarding reserves estimation.
- Allow the insurance company to decide on the optimal allocation of premiums to be invested and to obtain returns.

2. Organization Chart:

An illustration of organizational charts is provided below, showcasing a real organization's structure with actual information.

Allianz → Taken From: https://www.theofficialboard.es/organigrama/allianz-colombia



3. Selected Line of Business to test the model:

- Medical Malpractice

4. Relevant Stakeholders to involve in the project:

- Actuary VP
- CFO
- Operations
- Medical Malpractice UW
- Compliance Mngr.
- Transformation VP
- IT VP

5. Determining Data Mining Goals:

- Define Model: Deep Triangle (Deterministic / Linear Regression)
- Type of problem: Time series prediction
- Time spans of each prediction: Year

6. Problem Statement

The implementation of the chain Ladder analysis commonly covers the following items to fully understand the impacts and scope of the estimation/ solution.

1. Provisions are indispensable in addressing the future liabilities of insurance companies, ensuring readiness for losses yet to materialize.
2. Provision data finds its conventional representation in a development triangle, a visual aid capturing the temporal and payment-delay aspects of losses.
3. The chain-ladder method emerges as a prominent predictive modeling technique for estimating reserves, leveraging the similarity in payment rates for losses of the same age.
4. Generalized linear models (GLMs) offer a fully stochastic approach, enabling a nuanced understanding of the distribution of loss payments and, consequently, refining the accuracy of provision estimates.

In this document, we will address the reserves estimation problem by explaining and implementing the Chain Ladder Method. Additionally, different models, such as Deterministic and Linear Regression, will be covered for tackling the problem. First, let's provide a summary of what the Chain Ladder Method consists of:

Chain Ladder Method Overview:

1. Development of Losses Over Time:
   - The method is based on the idea that the development of insurance losses over time follows a consistent pattern.
   - Claims evolve from reported but not settled to fully settled over successive time periods (development periods).

2. Triangle of Loss Development:
   - Claims data is often arranged in a triangle format, known as a loss development triangle.
   - Rows represent accident years, and columns represent development periods. The entries in the triangle represent the number of incurred losses at a particular combination of accident year and development period.

3. Development Factors:
    - Development factors are calculated by comparing the losses at each development period to the losses at the previous period.
    - These factors represent the historical pattern of development and are used to project future developments.

4. Chain Ladder Estimation:
    - The Chain Ladder method involves "linking" or "chaining" these development factors across successive periods.
    - By applying these factors to the known losses in earlier periods, the method estimates the ultimate losses for each accident year.

The Chain Ladder Problem:

1. Data Quality and Completeness:
    - The accuracy of the estimates heavily relies on the quality and completeness of historical claims data.
    - Incomplete or inaccurate data can lead to unreliable estimates.

2. Assumption of Consistency:
    - The method assumes that the historical development pattern will continue into the future.
    - If there are significant changes in the claim's environment, the method may be less accurate.

3. Sensitivity to Outliers:
    - The Chain Ladder method can be sensitive to extreme values (outliers) in the data.
    - Outliers can disproportionately influence the development factors and, consequently, the reserve estimates.

4. Assumption of Homogeneity:
    - The method assumes homogeneity across all accident years, implying that the same development pattern applies to each year.
    - This assumption may not hold if there are distinct characteristics or changes in the claim's portfolio over time.

Despite these challenges, the Chain Ladder method remains widely used because of its simplicity and historical effectiveness. Actuaries often apply adjustments and additional statistical techniques to address some of the limitations and improve the accuracy of reserve estimates.

7. Timeline

Project Plan

Task	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Project Definition														
Data Analysis														
Model Analysis														
Model Validation														
Implementation Stage														
Release and Monitoring														

The summary of each one of the stages are shown below:

Project Definition:

- Objective: Clearly define the project's goals and objectives. Understand the problem statement and the value the project aims to deliver.
- Activities: Identify stakeholders, gather requirements, and establish key performance indicators (KPIs). Develop a comprehensive project plan, including timelines and resource allocation.
- Business Case definition (CBA – Cost benefits analysis).
- Risk analysis.

Data Analysis:

- Objective: Explore and understand the available data, identifying patterns, trends, and potential insights.
- Activities: Data collection, data cleaning, and exploratory data analysis (EDA). Utilize statistical methods and visualization tools to gain insights into the dataset. Formulate hypotheses and refine the project scope based on initial findings.

Model Analysis:

- Objective: Select and develop models that address the project's objectives, leveraging the insights gained from data analysis.

- Activities: Choose appropriate modeling techniques based on the nature of the problem (e.g., regression, classification). Train and fine-tune models using relevant algorithms. Evaluate model performance and select the most suitable one for further validation.

Model Validation:

- Objective: Assess the reliability and generalizability of the chosen model to ensure it performs well on new, unseen data.
- Activities: Split the dataset into training and testing sets. Validate the model using cross-validation or other validation techniques. Evaluate metrics such as accuracy, precision, recall, or F1 score to gauge model performance. Refine the model if necessary.

Implementation Stage:

- Objective: Integrate the validated model into the broader business or system environment.
- Activities: Develop and implement the necessary infrastructure and software to deploy the model. Collaborate with IT teams for seamless integration. Ensure data pipelines and model deployment align with the organization's technological architecture.

Release and Monitoring:

- Objective: Deploy the model into production and monitor its performance over time.
- Activities: Roll out the model to the production environment. Implement monitoring systems to track the model's behavior, performance, and any deviations from expected outcomes. Set up alerts for potential issues. Continuously optimize and update the model as needed.

Data Understanding & Preparation

Data Understanding & Preparation

This document contains the results obtained after performing data understanding & preparation analysis to the file medmal_pos.csv, which contains the information of claims for Medical Malpractice LoB.

The analysis was performed in google collab using python programming Language in the following link

https://colab.research.google.com/drive/1EUs1_qiNmV4mz8u4vnGQA3gfqeyNhsbD#scrollTo=K8fqrVgkQsL8

In the next sections, the results and analysis will be shown according to guidance and tips provided during the class.

Data Understanding

For understanding the data, we perform the following steps:

1. Displaying the first few rows of the dataset, as a quick wat to get an overview of what the data looks like:

```
[62] # Display the first few rows of the dataset
     print(df.head())

   GRCODE              GRNAME  AccidentYear  DevelopmentYear  DevelopmentLag  \
0     669  Scpie Indemnity Co          1988             1988               1
1     669  Scpie Indemnity Co          1988             1989               2
2     669  Scpie Indemnity Co          1988             1990               3
3     669  Scpie Indemnity Co          1988             1991               4
4     669  Scpie Indemnity Co          1988             1992               5

   IncurLoss_F2  CumPaidLoss_F2  BulkLoss_F2  EarnedPremDIR_F2  \
0        121905            2716        97966            129104
1        112211           24576        64117            129104
2        103226           43990        39008            129104
3         99599           59722        20736            129104
4         96006           71019        13599            129104

   EarnedPremCeded_F2  EarnedPremNet_F2  Single  PostedReserve97_F2
0               -6214            135318       0              344558
1               -6214            135318       0              344558
2               -6214            135318       0              344558
3               -6214            135318       0              344558
4               -6214            135318       0              344558
```

Can be identified that the file has this structure:

- GRCODE NAIC company code (including insurer groups and single insurers)
- GRNAME NAIC company name (including insurer groups and single insurers)
- AccidentYear Accident year (1988 to 1997)
- DevelopmentYear Development year (1988 to 1997)
- DevelopmentLag Development year (AY-1987 + DY-1987 - 1)
- IncurLoss_ Incurred losses and allocated expenses reported at year end.

- CumPaidLoss_ Cumulative paid losses and allocated expenses at year end
- BulkLoss_ Bulk and IBNR reserves on net losses and defense and cost containment expenses reported at year end
  PostedReserve97_ Posted reserves in year 1997 taken from the Underwriting and Investment Exhibit – Part 2A, including net losses unpaid and unpaid loss adjustment expenses.
- EarnedPremDIR_ Premiums earned at incurred year - direct and assumed.
- EarnedPremCeded_ Premiums earned at incurred year – ceded.
- EarnedPremNet_ Premiums earned at incurred year - net
  Single 1 indicates a single entity, 0 indicates a group insurer.

2. Check for missing Values

```
# Check for missing values
print(df.isnull().sum())
```

```
GRCODE              0
GRNAME              0
AccidentYear        0
DevelopmentYear     0
DevelopmentLag      0
IncurLoss_F2        0
CumPaidLoss_F2      0
BulkLoss_F2         0
EarnedPremDIR_F2    0
EarnedPremCeded_F2  0
EarnedPremNet_F2    0
Single              0
PostedReserve97_F2  0
dtype: int64
```

data.isnull() returns a Data Frame of the same shape as data but with Boolean values indicating whether each element is NaN (null) or not. .sum() is used to count the number of True values (which are equivalent to missing values) in each column., Our data set does not contain NaN elements.

3. Summary statistics:

	GRCODE	AccidentYear	DevelopmentYear	DevelopmentLag
count	3400.000000	3400.000000	3400.000000	3400.000000
mean	22809.764706	1992.500000	1997.000000	5.500000
std	14708.377001	2.872704	4.062617	2.872704
min	669.000000	1988.000000	1988.000000	1.000000
25%	10341.000000	1990.000000	1994.000000	3.000000
50%	19764.000000	1992.500000	1997.000000	5.500000
75%	36234.000000	1995.000000	2000.000000	8.000000
max	44504.000000	1997.000000	2006.000000	10.000000

	IncurLoss_F2	CumPaidLoss_F2	BulkLoss_F2	EarnedPremDIR_F2
count	3.400000e+03	3.400000e+03	3400.000000	3400.000000
mean	5.851528e-17	-5.433562e-17	1095.803235	14111.605882
std	1.000147e+00	1.000147e+00	7612.672277	26399.284476
min	-4.338370e-01	-4.612378e-01	-32101.000000	-781.000000
25%	-4.332026e-01	-3.917256e-01	0.000000	0.000000
50%	-4.091345e-01	-3.808023e-01	0.000000	1500.000000
75%	-9.548327e-02	-1.355527e-01	107.250000	18094.500000
max	6.262040e+00	6.220053e+00	104402.000000	131948.000000

	EarnedPremCeded_F2	EarnedPremNet_F2	Single	PostedReserve97_F2
count	3400.000000	3400.000000	3400.000000	3400.000000
mean	1803.497059	12308.108824	0.852941	57065.529412
std	3893.424584	24824.225795	0.354217	134355.533990
min	-6214.000000	-728.000000	0.000000	0.000000
25%	0.000000	0.000000	1.000000	629.000000
50%	106.500000	1302.000000	1.000000	5875.000000
75%	1473.500000	13490.000000	1.000000	46762.000000
max	25553.000000	135318.000000	1.000000	702246.000000

data.describe() generates summary statistics for numerical columns in the dataset. This includes count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum.

From the dataset we can state, that contains information that makes sense in the regard of claims historical information were incurred losses have been registered.

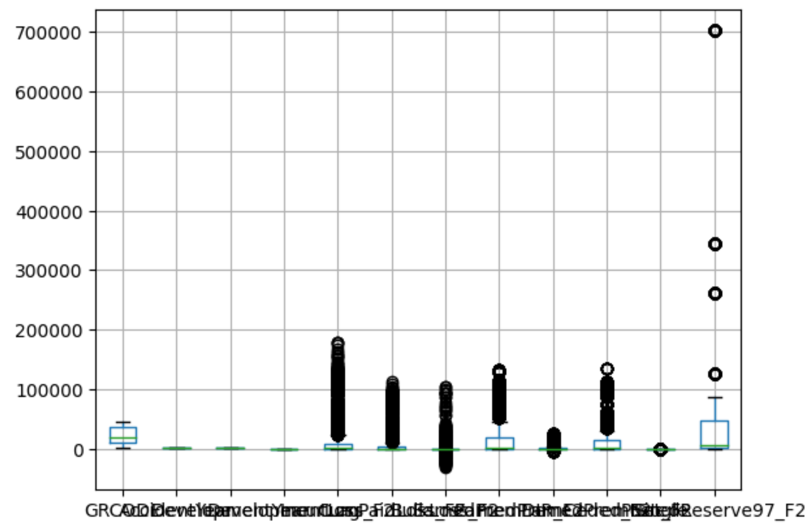
4. Data Visualization

sns.pairplot() generates a grid of scatter plots for the specified columns. In this case, it creates scatter plots for AccidentYear, DevelopmentYear, IncurLoss_F2, andCumPaidLoss_F2.
plt.show() displays the generated plot.

5. Checking outliers

```
# Check for outliers
print(df.boxplot())
```

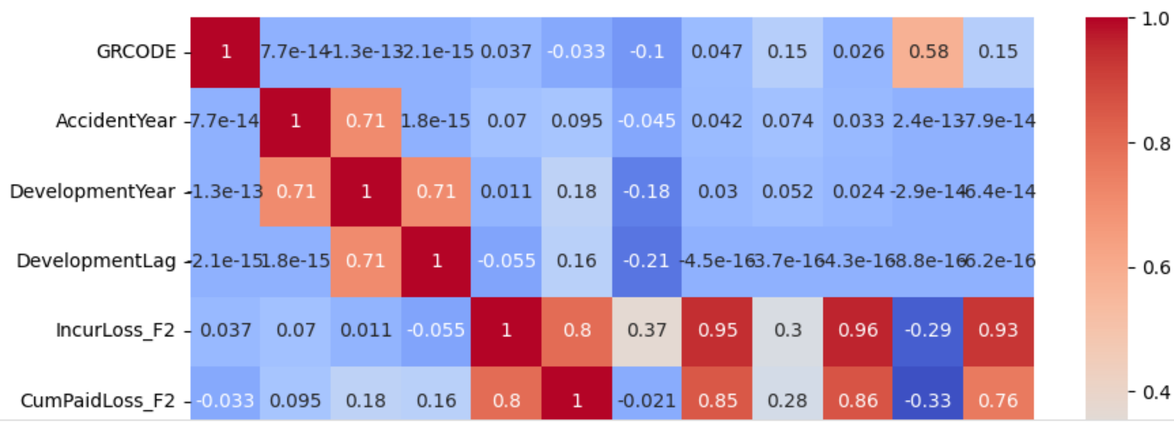
```
Axes(0.125,0.11;0.775x0.77)
```



6. Correlation Heatmap

```
[64] # Correlation heatmap
     corr = df.corr()
     plt.figure(figsize=(10, 8))
     sns.heatmap(corr, annot=True, cmap='coolwarm')
     plt.show()
```

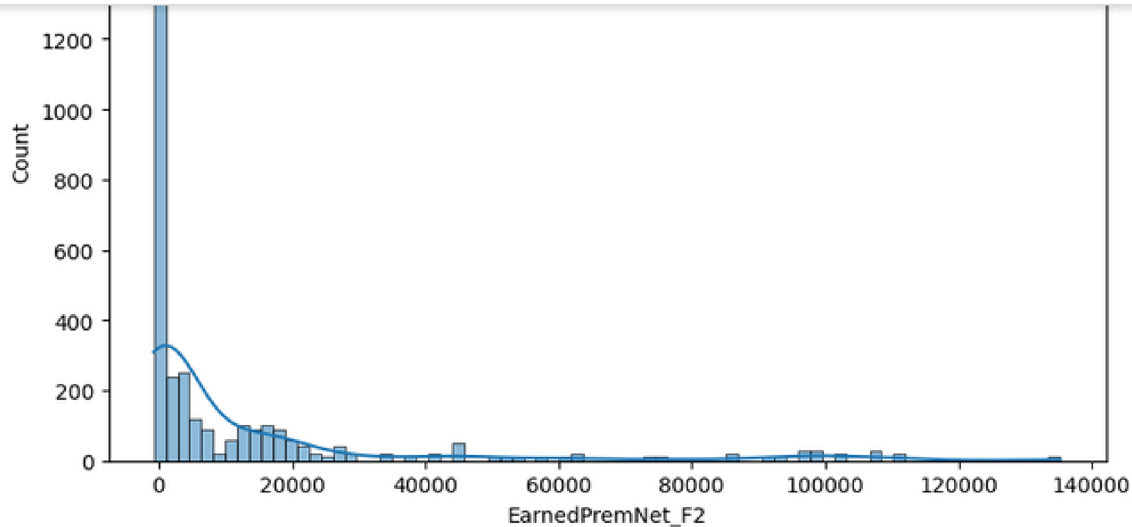
```
<ipython-input-64-6bfaeecc9ed7>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated.
  corr = df.corr()
```



data.corr() computes the pairwise correlation of columns in the dataset.
sns.heatmap() creates a heatmap to visualize the correlations.
annot=True adds the correlation values to the heatmap.
cmap='coolwarm' sets the color scheme.

7. Additional Data understanding steps:

In this section we try to match each column of the data set to a distribution, then we calculate the variance on each column.




```
Variance of each column:
GRCODE                 2.163364e+08
AccidentYear           8.252427e+00
DevelopmentYear        1.650485e+01
DevelopmentLag         8.252427e+00
IncurLoss_F2           1.000294e+00
CumPaidLoss_F2         1.000294e+00
BulkLoss_F2            5.795278e+07
EarnedPremDIR_F2       6.969222e+08
```

8. General results obtained during data understanding analysis:
- 34 Insurance companies exits in the data base.
- There are 1288 registries with empty/null information regarding incurred losses and cumpaid losses. The total number of rows in the file is 3400.
- After performing boxplot analysis, can be seen that de variable *PostedReserve97_F2,* presents outliers with the highest variance from the mean.
- Correlation:
    - ✓ Values Range:

- The correlation coefficient ranges from -1 to 1.
- A value of 1 indicates a perfect positive correlation (as one variable increases, the other also increases proportionally).
- A value of -1 indicates a perfect negative correlation (as one variable increases, the other decreases proportionally).
- A value of 0 indicates no linear correlation.

✓ Diagonal Elements:
  - The diagonal elements of the matrix (i.e., where i equals j) will always be 1 because a variable has a perfect correlation with itself.

✓ Interpreting Other Entries:
  - Positive values suggest a positive correlation (both variables tend to increase or decrease together).
  - Negative values suggest a negative correlation (as one variable increases, the other tends to decrease).

✓ Strength of Correlation:
  - The closer the correlation coefficient is to 1 or -1, the stronger the correlation.
  - Values around 0 indicate a weak or no correlation.

Data Preparation

This section of the code is dedicated to data preprocessing, with a specific emphasis on deciding which rows of data to include or exclude in the context of the Chain Ladder data analysis problem. Rows containing zero values for incurred losses and cumulative paid losses are slated for removal. This decision is pivotal for several reasons.

Firstly, eliminating rows with zero values contributes to enhancing data quality by excluding instances of missing or incomplete data. This ensures that the dataset remains robust and reliable for subsequent analysis. Additionally, the removal of zero values aligns with the need to improve model performance. Certain machine learning models and statistical analyses can be sensitive to zero values, potentially introducing noise or bias that adversely affects the accuracy of the model.

Furthermore, the process aims to enhance interpretability. Zero values might not align with the specific context of the analysis, and their removal serves to make the dataset more interpretable. This alignment with the assumptions and requirements of the analysis is crucial for deriving meaningful insights.

Lastly, the focus on relevant data is paramount in the context of addressing a reserve problem. By concentrating on information where claims and losses are paid, the analysis becomes more tailored to the core objective of reserve estimation. In summary, the decision to remove rows with zero values plays a critical role in improving data quality, model performance, interpretability, and ensuring the relevance of the data for addressing the specific challenges posed by the Chain Ladder data analysis problem.

Modeling

Deterministic & Linear regression approach

As mentioned before, Chain Ladder problem will be tacked with deterministic and linear regression approach, then performance of two models will be compared. Before staring with the development of each model a brief description of them in terms of capabilities, pros and cons will be shown:

Deterministic Approach:

Description:

1. Nature: The deterministic approach is a simpler method that relies on predetermined factors and fixed assumptions.
2. Assumption: Assumes a constant development pattern across all accident years.
3. Methodology:
   - Average Factors: Historical development factors are averaged to estimate future developments.
   - Constant Pattern: Assumes that the same pattern will apply to all years.

Pros:

1. Simplicity: Easy to understand and implement.
2. Quick Calculation: Requires minimal computation.

Cons:

1. Lack of Flexibility: Assumes a uniform development pattern, which may not capture variations.
2. Ignores Individual Patterns: Does not account for unique characteristics of each accident year.

Linear Regression Approach:

Description:

1. Nature: The linear regression approach is a statistical method that models the relationship between variables.
2. Assumption: Assumes a linear relationship between historical and future developments.
3. Methodology:
    - Regression Analysis: Uses historical data to identify a linear relationship between development factors and accident years.
    - Predictive Model: Develops a model to predict future developments based on this identified relationship.

Pros:

1. Flexibility: Can capture variations and changing patterns over time.
2. Statistical Rigor: Utilizes regression analysis, providing a more data-driven approach.

Cons:

1. Complexity: Requires statistical expertise for proper implementation.
2. Sensitivity to Outliers: Vulnerable to the influence of extreme values in the dataset.

In Conclusion:

Deterministic Approach: Simple and straightforward, but may oversimplify patterns and lack flexibility, potentially leading to less accurate estimates.

Linear Regression Approach: More sophisticated, allowing for flexibility and capturing nuanced patterns, but requires statistical expertise and is sensitive to outliers.

The choice between these approaches often depends on the complexity of the data, the availability of historical information, and the level of precision required in the reserve's estimation process. Analysts may choose one approach over the other based on the specific characteristics of the insurance portfolio being analyzed.

Modeling Building

Deterministic Approach

First is we create a class named `ChainLadder` with methods for performing a Chain Ladder analysis on insurance claims data. Let's break down the code step by step:

This method performs the Chain Ladder analysis on the provided dataset. It renames columns, organizes data into triangles, and calculates development factors to estimate reserves. The results are stored in a dictionary called `diccionario_todos_triangulos`, where each key corresponds to a unique identifier from the "GRCODE" column in the dataset.

Chain Ladder Deterministic Analysis Steps:

1. Data Preparation:
   - The provided dataset is prepared by renaming columns for consistency.

2. Triangle Formation:
   - The method iterates over unique values in the "GRCODE" column and creates triangles of incurred losses over accident years and development lags.
   - Triangles include full, half, and cumulative versions of the original data.

3. Factor Calculation:
   - Development factors are calculated based on the cumulative triangles.

4. Estimation of Ultimate Losses:
   - Using the calculated factors, the method estimates ultimate losses for each cell in the triangle.

5. Reserve Calculation:
   - The total reserve is computed as the sum of the differences between reversed and flipped diagonals of the estimated triangle.

6. Results Storage:
   - The results for each unique "GRCODE" identifier are stored in a dictionary (`diccionario_triangulo`) and added to the overall results dictionary (`diccionario_todos_triangulos`).

7. Returning Results:

The method returns the dictionary `diccionario_todos_triangulos`, which contains detailed results of the Chain Ladder analysis for each unique "GRCODE" in the dataset.

Overall, this code defines a class that encapsulates the Chain Ladder analysis logic, making it modular and reusable for different insurance claims datasets. The primary goal is to estimate reserves and understand the development patterns of incurred losses over time.

Once the class is defined, we test the performance of the code determining de results of one insurance company, in this case we test with GRCODE 669, and the results are shown below:

Insurer information as is:

DevelopmentLag	1	2	3	4	5	6	7	8	9	10
AccidentYear										
1988	121905	112211	103226	99599	96006	90487	82640	80406	78920	78511
1989	122679	113165	110037	101142	90817	81919	77491	73577	72716	72317
1990	118157	117497	116377	99895	89252	81916	79134	76333	75612	75350
1991	117981	122443	121056	113795	102830	98071	94870	91062	90493	90345
1992	131059	130155	124195	113974	106817	99182	92588	91000	89256	89251
1993	134700	130757	125253	114717	111294	98014	96872	95714	96017	96047
1994	136749	128192	121355	111877	96152	91502	90498	91870	91848	91938
1995	140962	132405	118332	100050	88809	82360	81986	81887	81796	81782
1996	134473	128980	113645	104273	99276	97782	97282	97738	97601	97251
1997	137944	127727	114057	107001	102143	99665	99942	99968	99590	99378

Insurer information as is – ignoring lower triangle “to be estimated.”

DevelopmentLag	1	2	3	4	5	6	7	8	9	10
AccidentYear										
1988	121905	112211.0	103226.0	99599.0	96006.0	90487.0	82640.0	80406.0	78920.0	78511.0
1989	122679	113165.0	110037.0	101142.0	90817.0	81919.0	77491.0	73577.0	72716.0	NaN
1990	118157	117497.0	116377.0	99895.0	89252.0	81916.0	79134.0	76333.0	NaN	NaN
1991	117981	122443.0	121056.0	113795.0	102830.0	98071.0	94870.0	NaN	NaN	NaN
1992	131059	130155.0	124195.0	113974.0	106817.0	99182.0	NaN	NaN	NaN	NaN
1993	134700	130757.0	125253.0	114717.0	111294.0	NaN	NaN	NaN	NaN	NaN
1994	136749	128192.0	121355.0	111877.0	NaN	NaN	NaN	NaN	NaN	NaN
1995	140962	132405.0	118332.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1996	134473	128980.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1997	137944	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Insurer prediction reserves estimation of lower triangle.

DevelopmentLag	1	2	3	4	5	6	7	8	9	10
AccidentYear										
1988	121905	234116	337342	436941	532947	623434	706074	786480	865400	943911
1989	122679	235844	345881	447023	537840	619759	697250	770827	843543	915860
1990	118157	235654	352031	451926	541178	623094	702228	778561	854173	929523
1991	117981	240424	361480	475275	578105	676176	771046	862108	952601	1042946
1992	131059	261214	385409	499383	606200	705382	797970	888970	978226	1067477
1993	134700	265457	390710	505427	616721	714735	811607	907321	1003338	1099385
1994	136749	264941	386296	498173	594325	685827	776325	868195	960043	1051981
1995	140962	273367	391699	491749	580558	662918	744904	826791	908587	990369
1996	134473	263453	377098	481371	580647	678429	775711	873449	971050	1068301
1997	137944	265671	379728	486729	588872	688537	788479	888447	988037	1087415

As can be seen lower triangle estimation is similar to the existing information -as is-, later in this document we will perform analysis to understand how precise and exact are the obtained prediction of reserves.

Modeling Building

Linear regression approach

Using a linear regression approach in the context of Chain Ladder reserves estimation involves applying regression techniques to model the relationship between various factors (such as development lags) and incurred losses. The idea is to use historical data to identify a linear relationship and then use that relationship to predict future developments. Here's a step-by-step explanation:

Steps in Applying Linear Regression to Chain Ladder Reserves Estimation:

1. Data Preparation:
   - Organize historical data into a suitable format, often in the form of a development triangle where rows represent accident years, columns represent development lags, and entries represent incurred losses.

2. Feature Selection:
   - Identify relevant features (independent variables) and the target variable (incurred losses). In the context of Chain Ladder, development lags are common independent variables.

3. Linear Regression Model:
   - Fit a linear regression model to the historical data. The model will have the form:
   - **$Y=\beta0+\beta1X1+\beta2X2+\ldots+\beta nXn+\epsilon Y=\beta0+\beta1X1+\beta2X2+\ldots+\beta nXn+\epsilon$**
   - Y is the target variable (incurred losses), $\beta0\beta0$ is the intercept, $\beta1,\beta2,\ldots,\beta n\beta1,\beta2,\ldots,\beta n$ are the coefficients for the independent variables $X1,X2,\ldots,XnX1,X2,\ldots,Xn$, and $\epsilon\epsilon$ represents the error term.

4. Model Evaluation:
   - Assess the goodness of fit of the model using relevant metrics, such as R-squared, mean squared error, or others. This step helps evaluate how well the model captures the historical development patterns.

5. Prediction:
   - Use the fitted model to predict future developments. This involves applying the identified linear relationship to estimate incurred losses for future time periods.

6. Reserve Calculation:
    - The predicted values from the linear regression model can be used to estimate ultimate losses for each accident year.

7. Adjustments and Refinement:
    - Evaluate the model's performance and make adjustments as needed. This may involve refining the feature selection, considering interactions or non-linear relationships, or incorporating additional factors for a more accurate model.

Considerations and Challenges:

1. Assumptions:
    - Linear regression assumes a linear relationship between the independent and dependent variables. If the relationship is non-linear, additional techniques or adjustments may be necessary.

2. Data Quality:
    - The accuracy of predictions heavily depends on the quality and completeness of the historical data. Outliers and missing data can affect the model's performance.

3. Changing Patterns:
    - Linear regression assumes that the historical pattern will continue into the future. If there are significant changes in the claim’s environment, the model may be less accurate.

4. Interpretability:
    - Linear regression models provide interpretable coefficients that can offer insights into the impact of each independent variable on the target variable.

5. Model Complexity:
    - The simplicity of linear regression can be an advantage, but more complex relationships may require advanced modeling techniques.

By leveraging linear regression, analysts can create a data-driven model to estimate reserves based on historical development patterns. However, it's essential to validate the model's assumptions and continuously refine it for accurate predictions in dynamic insurance environments.

The implementation in python was developed creating the class `Reserva_Regresion_lineal`, the features of the class are described below:

The Reserva_Regresion_lineal class has four attributes:

`tabla`: The data table to be used for training and prediction.
`origin`: The column in the table that indicates the accident year.
`development`: The column in the table that indicates the loss lag.
`columns`: The columns in the table to be used for training and prediction.
`matriz_de_ceros`: A matrix of zeros of the same size as the data table to be used for training and prediction.

The `Regresion_lineal` method trains three linear regression models:

1. A normal linear model.
2. A linear regression model with Ridge regularization.
3. A linear regression model with Lasso regularization.

The predict method predicts the unobserved losses in a new data table. First, it trains the three linear regression models. Then, for each unobserved observation, it predicts the loss with the model that performed best on the training dataset.

The `predict_test` method predicts the unobserved losses in a new data table. First, it trains the three linear regression models. Then, it predicts the unobserved losses with the model that performed best on the training dataset.

Example of use:
Python

```
import pandas as pd
from Reserva_Regresion_lineal import Reserva_Regresion_lineal

# Load the data table
datos = pd.read_csv("data/data.csv")

# Create class object
reserva = Reserva_Regresion_lineal(datos, "AccidentYear", "DevelopmentLag", "IncurLoss_C",
"GRCODE", np.zeros((datos.shape[0], 10)))
```

```
# Train the models
resultados = reserva.Regresion_lineal()

# Predict unobserved losses in a new data set
datos_test = pd.read_csv("data/data_test.csv")
resultados_test = reserva.predict_test(datos_test)
```

This code loads the data table data.csv and creates an object of the class Reserva_Regresion_lineal. Then, the method Regresion_lineal trains the three linear regression models. The method predict_test predicts the unobserved losses in the data set data_test.csv.

Once we perform the class, the results for one insurer company can be validated, we can be compared

Insurer information as is:

DevelopmentLag	1	2	3	4	5	6	7	8	9	10
AccidentYear										
1988	121905	112211	103226	99599	96006	90487	82640	80406	78920	78511
1989	122679	113165	110037	101142	90817	81919	77491	73577	72716	72317
1990	118157	117497	116377	99895	89252	81916	79134	76333	75612	75350
1991	117981	122443	121056	113795	102830	98071	94870	91062	90493	90345
1992	131059	130155	124195	113974	106817	99182	92588	91000	89256	89251
1993	134700	130757	125253	114717	111294	98014	96872	95714	96017	96047
1994	136749	128192	121355	111877	96152	91502	90498	91870	91848	91938
1995	140962	132405	118332	100050	88809	82360	81986	81887	81796	81782
1996	134473	128980	113645	104273	99276	97782	97282	97738	97601	97251
1997	137944	127727	114057	107001	102143	99665	99942	99968	99590	99378

Insurer information after model performance: -Lower triangle prediction-

DevelopmentLag	1	2	3	4	5	6	7	8	9	10
AccidentYear										
1988	121905	112211	103226	99599	96006	90487	82640	80406	78920	78511
1989	122679	113165	110037	101142	90817	81919	77491	73577	72716	72317
1990	118157	117497	116377	99895	89252	81916	79134	76333	75612	75350
1991	117981	122443	121056	113795	102830	98071	94870	91062	90493	90345
1992	131059	130155	124195	113974	106817	99182	92588	91000	89256	89251
1993	134700	130757	125253	114717	111294	98014	96872	95714	96017	96047
1994	136749	128192	121355	111877	96152	91502	90498	91870	91848	91938
1995	140962	132405	118332	100050	88809	82360	81986	81887	81796	81782
1996	134473	128980	113645	104273	99276	97782	97282	97738	97601	97251
1997	137944	127727	114057	107001	102143	99665	99942	99968	99590	99378

Evaluation

MAPE analysis

Once the results are obtained, we performed MAPE analysis to understand if how precise the model estimation fits to knew data. We will provide a short explanation of what M.A.P.E is.

MAPE stands for Mean Absolute Percentage Error. It is a statistical measure of how accurate a forecasting model is. It is calculated by taking the average of the absolute percentage errors of each prediction made by the model. The lower the MAPE, the more accurate the model.

MAPE is a popular metric for forecasting models because it is relatively easy to understand and interpret. It is also a good measure of how well a model is performing relative to the size of the values being predicted. However, MAPE can be sensitive to outliers, and it can be biased towards smaller values.

Here is the formula for MAPE:

MAPE = (1/n) * Σ | (Actual - Forecast) / Actual| * 100

Where:

- n is the number of predictions made by the model
- Actual is the actual value of the variable being predicted.
- Forecast is the predicted value of the variable being predicted.

MAPE is often used to compare the performance of different forecasting models. It can also be used to track the performance of a forecasting model over time.

Here are some examples of when MAPE might be used:

A company might use MAPE to forecast its sales for the next year.
A weather forecast might use MAPE to forecast the temperature for the next day.
A financial analyst might use MAPE to forecast the stock price of a company.

And in the context of this document, we will use MAPE to forecast the reserve modelling for the reserve estimation.

The implementation of MAPE calculation error in the code we obtained the following results:

```
reultados_validacion[0]['Metricas MAPE'] #Acá se evaluan los tres modelos en el conjunto de validación o aseguradora de validación

{'MAPE': 17.862565448352345,
 'MAPE_ridge_1': 17.862732438617133,
 'MAPE_lasso': 17.86437860509435}
```

Based on the provided MAPE results, the linear regression model appears to be the best model among the three models tested. This is because it has the lowest MAPE value, which indicates that it is the most accurate in predicting the observed losses. The MAPE values for the ridge and lasso models are slightly higher, indicating that they are not as accurate as the linear regression model.

Here is a table summarizing the MAPE results:

Model	MAPE
Linear regression	17.862565448352345
Ridge regression	17.862732438617133
Lasso regression	17.86437860509435

As can be seen, the MAPE values for the ridge and lasso models are very close to the MAPE value for the linear regression model. This suggests that all three models are performing relatively well, and that the differences in their MAPE values are not statistically significant. However, the fact that the linear regression model has the lowest MAPE value suggests that it is the most likely model to generalize well to new data. This means that it is the most likely model to accurately predict losses for insurance reserves that were not included in the training data.

Therefore, based on the available evidence, the linear regression model is the best model for predicting unobserved losses for insurance reserves.

Cross Validation Modelling:

We implemented a code with the following characteristics:

The code is a stratified cross-validation algorithm for selecting linear regression models for insurance reserve estimation. The algorithm works as follows:

1. First, the algorithm creates a list of all the insurers in the cross-validation data set.

2. Then, for each insurer in the list, the algorithm performs the following steps:
- Randomly selects an insurer from the list to use as the validation set.
- Randomly selects the rest of the insurers from the list to use as the training set.
- Trains a linear regression model on the training set.
- Predicts the unobserved losses in the validation set.

3. After performing these steps for all insurers in the list, the algorithm selects the model that has the lowest prediction error on the validation set.

The code implements this algorithm as follows:

- The variable lista_aseguradoras contains a list of all the insurers in the cross-validation data set.
- The variable mejore_modelos_test_full is a dictionary that contains the linear regression models for each insurer.

- The for loop for i in range(len(lista_aseguradoras)): iterates over every insurer in the list.
  - The variable conj_test contains the code of the insurer that will be used as the validation set.
  - The variable datos_test contains the data for the insurer that will be used as the validation set.
  - The variable conj_entre_valid contains a list of the codes of the insurers that will be used as the training and validation set.

- The for loop for j in range(len(conj_entre_valid)): iterates over every insurer in the training and validation set.
  - The variable conj_vali contains the code of the insurer that will be used as the validation set.
  - The variable conj_entre contains a list of the codes of the insurers that will be used as the training set.
  - The variable datos_train contains the data for the insurers that will be used as the training set.
  - The variable datos_validacion contains the data for the insurer that will be used as the validation set.
  - The variable model1 is an object of the Reserva_Regresion_lineal class.
  - The variable model1_regresion contains the results of training the linear regression model.
  - The variable model1_prediccion contains the predictions of the linear regression model for the validation set.
  - The variable modelo_test contains the results of predicting the linear regression model for the test set.

- The variable modelo_final is a dictionary that contains the prediction errors of each linear regression model.

- The for loop for i in mejore_modelos_test_full.keys() iterates over each linear regression model in the mejore_modelos_test_full dictionary.
  - The variable i contains the code of the linear regression model.
  - The variable modelo_final[i] contains the prediction error of the linear regression model.

- The variable nombre_modelo_final contains the code of the linear regression model with the lowest prediction error.

- The variable Mejor_modelo_reserva contains the results of the linear regression model with the lowest prediction error.

In summary, the stratified cross-validation algorithm implemented in the code you provided works as follows:

1. The cross-validation data set is divided into two sets: a training set and a validation set.
2. For each insurer in the cross-validation data set, an insurer from the training set is randomly selected to use as the validation set.

3. A linear regression model is trained on the training set.
4. The unobserved losses in the validation set are predicted.
5. The prediction error of the linear regression model is calculated.
6. The linear regression model with the lowest prediction error is selected.

This algorithm is an effective way to select linear regression models for insurance reserve estimation.

And the result obtained are shown as follows:

```
[55] Mejor_modelo_reserva["nombre mejor modelo"]

     'Regresión de Lasso'
```

```
[58] Mejor_modelo_reserva["mejor modelo"]

     ▾ Lasso
     Lasso(alpha=0.001)
```

For selecting the best model:

The code calculates the MAPE (Mean Absolute Percentage Error) for the Chain-Ladder method of insurance reserve estimation. MAPE is a measure of the accuracy of forecasting models, and it is calculated by averaging the absolute percentage errors for each prediction made by the model.

The code first iterates over each insurer in the data set. For each insurer, it calculates the absolute difference between the estimated losses and the observed losses for all triangles and all development lags. It then divides the sum of these differences by the total number of triangles and multiplies the result by 100 to express MAPE as a percentage. The MAPE value for each insurer is appended to a list.

Finally, the code calculates the average MAPE across all insurers and prints the result. This average MAPE value represents the overall performance of the Chain-Ladder method for the given data set.

And the result is obtained as follows:

```
[61] print("Métrica MAPE con el método Chain-Ladder:",Mape_chain_ladder)
     print("Métrica MAPE con el modelo final:",Mejor_modelo_reserva["Metricas MAPE"]['MAPE modelo final'])

Métrica MAPE con el método Chain-Ladder: 7.036457837171299
Métrica MAPE con el modelo final: 0.9320003843622218
```

Conclusion:

After obtaining results, it can be concluded that the best fitting model is Lasso Linear Regression, with MAPE metric of 0,93%. Lasso Linear Regression fits better the model to new data.