Data Mining 4th year
Software Engineering

Duration: 3 hours
2025/2026

TP2 : Data Collection, Exploration, and Preprocessing

Dataset: insurance_claims.csv

Tools: Python, Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn

1. Data Collection and Exploration:

1. Import necessary libraries (pandas, numpy).

   [Pandas](http://pandas.pydata.org) is a Python library that provides extensive means for data analysis. Data scientists often work with data stored in table formats like `.csv`, `.tsv`, or `.xlsx`. Pandas makes it very convenient to load, process, and analyze such tabular data using SQL-like queries.

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


import seaborn as sns
sns.set()  #  Will import Seaborn functionalities


# we don't like warnings
# you can comment the following 2 lines if you'd like to
import warnings
warnings.filterwarnings('ignore')
```

2. Load the dataset into a DataFrame.

```
# # Disply all Columns

pd.options.display.max_columns = 40

autinsurance = pd.read_csv('insurance_claims.csv')
autinsurance
```

3. Display the first rows, number of rows/columns, and general information.

```
#autinsurance.info()
 Column names may be accessed (and changed) using the `.columns`
attribute as below
print("Old Column Names:\n", autinsurance.columns)


# print(autinsurance.columns)


! pip install pyjanitor
import janitor
autinsurance = autinsurance.clean_names()
print("New Janitor Column Names:\n", autinsurance.columns)
```

4. Identify data types and missing values.

```
#autinsurance.isna().sum()  ## Count Filled cells
                               ## Sum empty cells
```

5. dimensionality, features names, and feature types.

```
#dir(autinsurance)
#autinsurance.shape
```

Number of rows

```
#len(autinsurance)
```

Questions :

- How many rows and columns does the dataset contain?
- What types of data (categorical, numerical, binary) are present?
- Are there missing or inconsistent values?
- Which attributes could be the **target variable**?

2. Exploratory Data Analysis (EDA) & Visualization

The `**describe**` method shows basic statistical characteristics of each numerical feature (`int64` and `float64` types): number of non-missing values, mean, standard deviation, range, median, 0.25 and 0.75 quartiles.

1. Use .describe() to get mean, median, mode, min, max, std.

```
autinsurance.describe()
```

2. Distribution of target variable

```
autinsurance.describe(include=['object'])

autinsurance['fraud_reported'].value_counts()
autinsurance['fraud_reported'].value_counts(normalize=True)

(autinsurance['fraud_reported'].value_counts().plot(
        kind='bar',
        figsize=(4, 2),
        title='Distribution of Target Variable',
    )
);
plt.show()
```

3. Plot histograms for numeric variables.

```
features = ['age', 'months_as_customer', 'capital_gains',
'capital_loss']

autinsurance[features].hist(figsize=(6, 4));
```

#Second

```
autinsurance[features].plot(kind='density', subplots=True,
layout=(2, 2),
                  sharex=False, figsize=(6, 4));
```

4. Correlation matrix using Seaborn (sns.heatmap).

Let's look at the correlations among the numerical variables in our dataset.First, we will use the method [`corr()`]on a `DataFrame`

that calculates the correlation between each pair of features. Then, we pass the resulting *correlation matrix* to [`heatmap()`] from `seaborn`, which renders a color-coded matrix for the provided values:

autinsurance.head(5)

```
corr_matrix = autinsurance.corr(method = 'kendall')
corr_matrix
```

Highly correlated items = not good
Low correlated items = good

#How to calculate correlation between all columns and remove highly correlated ones using pandas?

```
# import numpy as np

# # Create correlation matrix
# corr_matrix = autinsurance.corr().abs()

# # Select upper triangle of correlation matrix
# upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape),
k=1).astype(np.bool))

# # Find features with correlation greater than 0.95
# to_drop = [column for column in upper.columns if any(upper[column] >
0.90)]

# # Drop features
# autinsurance.drop(to_drop, axis=1, inplace=True)

# autinsurance.shape
```

5. Identify possible relationships between features and the target.

```
# # Correlation with Target Variable

# autinsurance.corr()['age'].plot()
# #autinsurance.corr()[:].plot()
# #autinsurance.corr().plot()
```

6. Pair plots for key numerical variables.
7. Bar charts for categorical variables.

Questions:

- What are the main central tendency measures (mean, median, mode)?
- Which columns contain outliers?
- How would you justify removing or keeping these outliers?
- Which features are positively or negatively correlated?
- Which attributes seem to influence claim fraud most?

3. Data Cleaning

1. Detect missing values (`isnull().sum()`).
2. Apply different techniques:
    - Fill with mean/median/mode.
    - Replace categorical missing values with "Unknown".
    - Drop unnecessary or heavily missing columns.
3. Detect and remove duplicates.

Questions:

- Which cleaning method worked best for missing data?
- How did cleaning affect dataset shape and summary statistics?
- Why is it risky to fill all missing values automatically?
- Should outliers always be removed? Explain.

4. Data Transformation and Normalization

1. Apply:
    - Min–Max normalization (`MinMaxScaler`).
2. Encode categorical variables:
    - One-Hot Encoding (with `pd.get_dummies` or `OneHotEncoder`).
    - Label Encoding if needed.
3. Compare model inputs before and after transformation.
4. Generate a **Data Quality Report** (Write a short audit report describing data issues and preprocessing actions).

Questions:

- Which normalization technique do you recommend for this dataset?
- What is the effect of encoding categorical variables?
- Are there categorical variables with too many unique values?
- What main data quality problems did you identify?
- How did preprocessing improve your dataset’s consistency?