Name: Ashish Verma **Class:CS773** **HW8**

Detecting Plagiarism in Student Essays Using Data Mining

Objective: To build a system that determines whether an essay submitted by a student is original or plagiarized.

Data Required:

1. **Corpus of Student Essays:**
    - A large dataset of essays submitted by students, both from current and past submissions.
    - Labeled examples of known plagiarized and original essays.
2. **Reference Databases:**
    - Access to online repositories of academic papers, articles, books, and other sources commonly used by students.
    - Web crawlers to gather publicly available content from the internet.
3. **External Plagiarism Databases:**
    - Databases that contain examples of known plagiarized content and their sources.

Steps to Build the System:

1. Data Preprocessing:

- **Text Cleaning:**
    - Remove stop words, punctuation, and other non-informative elements from the essays.
    - Normalize text by converting to lowercase and handling synonyms.
- **Tokenization:**
    - Split essays into tokens (words or phrases) for analysis.
    - Generate n-grams (sequences of n words) to capture context.
- **Feature Extraction:**
    - Extract features such as word frequency, sentence structure, stylistic patterns, and syntactic complexity.
    - Compute TF-IDF (Term Frequency-Inverse Document Frequency) scores to identify significant terms.

2. Building the Plagiarism Detection Model:

- **Similarity Analysis:**
  - Use cosine similarity or Jaccard similarity to compare the submitted essay with the reference databases and corpus of student essays.
  - Implement sliding window techniques to compare smaller chunks of text for partial matches.
- **Machine Learning Classifiers:**
  - Train machine learning models (e.g., SVM, random forest, neural networks) on the labeled dataset of essays to distinguish between original and plagiarized content.
  - Features for training could include similarity scores, stylistic markers, and structural patterns.
- **Stylometry:**
  - Apply stylometric analysis to compare the writing style of the submitted essay with the student's previous work. Features include average sentence length, use of passive voice, vocabulary richness, and more.
  - Use clustering or classification techniques to detect discrepancies in writing style.

3. Post-Processing and Validation:

- **Plagiarism Score:**
  - Compute a plagiarism score based on the combination of similarity analysis and machine learning predictions.
  - Higher scores indicate a higher likelihood of plagiarism.
- **Human Review:**
  - Flag essays with high plagiarism scores for manual review by educators.
  - Provide detailed reports highlighting suspected plagiarized sections and their sources.

4. Continuous Improvement:

- **Feedback Loop:**
  - Incorporate feedback from manual reviews to refine and retrain the machine learning models.
  - Continuously update the reference databases with new sources and student submissions.

Detailed Workflow:

1. **Data Collection and Storage:**
    - Collect and store essays, reference materials, and known plagiarized examples in a structured database.
2. **Preprocessing:**
    - Clean and preprocess the collected text data.
3. **Feature Extraction:**
    - Extract meaningful features from the text data using NLP techniques.
4. **Similarity Analysis:**
    - Compute similarity scores between the submitted essay and reference materials.
5. **Machine Learning Model:**
    - Train and validate machine learning models to classify essays as original or plagiarized.
6. **Stylometric Analysis:**
    - Perform stylometric analysis to compare writing styles.
7. **Integration and Scoring:**
    - Integrate results from similarity analysis, machine learning models, and stylometric analysis to compute a final plagiarism score.
8. **Manual Review:**
    - Provide detailed reports for essays with high plagiarism scores for human review.
9. **Feedback and Refinement:**
    - Use feedback from manual reviews to refine models and update databases.

Summary

This scheme leverages data mining techniques, including NLP, similarity analysis, and machine learning, to build a robust plagiarism detection system. By combining automated analysis with human oversight, the system aims to accurately identify plagiarized content while continuously improving its accuracy and reliability.

Fraudulent Expense Claims Using Data Mining

Objective: To build a system that detects potentially fraudulent meal expense claims made by employees while entertaining clients.

Data Required:

1. **Expense Claim Records:**
    - Detailed records of meal expenses, including date, amount, employee ID, client details, and purpose of the meal.
2. **Employee Profiles:**
    - Information about employees, such as department, role, tenure, and historical expense claim patterns.
3. **Client Interaction Data:**
    - Information about client meetings, including schedules, client interactions, and associated expenses.

Steps to Build the System:

1. Data Preprocessing:

- **Data Cleaning:**
    - Remove duplicates and inconsistencies in the data.
    - Normalize and standardize data formats (e.g., date formats, currency).
- **Data Integration:**
    - Merge expense claim records with employee profiles and client interaction data.

2. Feature Extraction:

- **Descriptive Features:**
    - Extract basic features such as total amount, frequency of claims, and average claim amount.
    - Compute additional features such as meal duration, time of the day, and day of the week.
- **Behavioral Features:**
    - Analyze patterns in expense claims, such as periodicity and variability.

    - Identify deviations from typical behavior for each employee (e.g., sudden spikes in claim amounts).
- **Contextual Features:**
    - Include contextual information like location (geographical patterns), client importance (high-value vs. low-value clients), and purpose of the meeting.

3. Anomaly Detection:

- **Outlier Detection:**
    - Use statistical methods (e.g., z-scores, IQR) to identify claims that deviate significantly from the norm.
    - Apply clustering techniques (e.g., DBSCAN, k-means) to group similar claims and identify outliers.
- **Machine Learning Models:**
    - Train unsupervised models (e.g., Isolation Forest, Autoencoders) on historical claim data to detect anomalies.
    - Use supervised models (e.g., Random Forest, SVM) if labeled data (known fraudulent vs. legitimate claims) is available.
- **Pattern Recognition:**
    - Use association rule mining (e.g., Apriori algorithm) to identify common patterns in fraudulent claims.
    - Implement sequence analysis to detect unusual sequences of claims (e.g., high-frequency claims over a short period).

4. Fraud Scoring:

- **Feature Engineering:**
    - Create a fraud score based on the combination of extracted features and anomaly detection results.
    - Higher scores indicate a higher likelihood of fraud.
- **Threshold Setting:**
    - Determine thresholds for the fraud score to flag claims for further investigation.
    - Adjust thresholds based on the company's risk tolerance and historical fraud patterns.

5. Post-Processing and Review:

- **Alert Generation:**
  - Generate alerts for claims that exceed the fraud score threshold.
  - Prioritize alerts based on the severity of the score and potential impact.
- **Manual Review:**
  - Provide detailed reports for flagged claims, highlighting the reasons for suspicion.
  - Allow auditors or managers to review and validate the flagged claims.

6. Continuous Monitoring and Improvement:

- **Feedback Loop:**
  - Incorporate feedback from manual reviews to refine and improve the models.
  - Update models periodically with new data to adapt to changing patterns of behavior.
- **Periodic Audits:**
  - Conduct periodic audits of expense claims to ensure the system's effectiveness.
  - Continuously monitor and fine-tune the fraud detection system.

Detailed Workflow:

1. **Data Collection and Integration:**
   - Collect expense claim records, employee profiles, and client interaction data.
   - Integrate and preprocess the data for analysis.
2. **Feature Extraction and Engineering:**
   - Extract and engineer relevant features for detecting anomalies and patterns.
3. **Anomaly Detection:**
   - Apply statistical, clustering, and machine learning techniques to identify suspicious claims.
4. **Fraud Scoring:**
   - Compute a fraud score for each claim based on the extracted features and detection results.
5. **Alert Generation and Manual Review:**

    - Generate alerts for high-risk claims and provide detailed reports for manual review.
6. **Continuous Monitoring and Improvement:**
    - Implement a feedback loop and conduct periodic audits to refine the system.

Summary

By leveraging data mining techniques such as anomaly detection, pattern recognition, and machine learning, the company can effectively identify potentially fraudulent expense claims among a large number of employees. This approach allows for automated, scalable, and continuous monitoring of expense claims, significantly reducing the burden of manual checks while increasing detection accuracy.