**Thema: Theoretische und praktische Aspekte eines K1 geschützten Content-Filter Systems in Verbindung mit der Entwicklung und praktischen Erprobung im Unternehmensumfeld**

Mit dem Fortschreiten der neuen Technologien im Bereich der Informations- und Kommunikationstechnologie (IKT) haben sich die im Internet enthaltenen Daten drastisch erhöht, was dazu führte, dass einige Internetsuchen ungenau waren. Mit dem Aufkommen von Suchmaschinen wie Google wurde die Suche jedoch einfacher und genauer.

Das reicht jedoch nicht aus, da wir jeden Tag mit Hackern, gewalttätigen oder pornografischen Websites, Fake-Websites und vielen anderen Bedrohungen konfrontiert sind.

Die Frage, die wir uns stellen, ist, ob es nicht eine Möglichkeit gibt, all diese Bedrohungen zu blockieren oder die Möglichkeit den Zugang zu Webseiten anhand ihrer Kategorisierung zu unterbinden. Dabei müssen die Kategorien nicht zwangsläufig schädlicher Natur oder primär für die Strafverfolgung relevant sein.

Die naheliegendste Lösung wäre, ein System einzurichten, das den Inhalt einer Webseite analysiert und sagt, zu welcher Kategorie sie gehört, um den normalen Benutzer vor potenziellen Bedrohungen zu schützen.

**Aufgabenstellung**

1. **Aktueller Stand in der Firma**

Die Firma Telco Tech GmbH verfügt über eine Datenbank die bereits Datensätze zur Kategorisierung von Webseiten enthält.

1. **Ziel**

Ziel ist es, eine Anwendung zu entwickeln, die eine URL oder Webseite (die vom Benutzer eingegeben wird) und auch, bis zu einer gewissen Tiefe, ihre weiterführenden Unterseiten oder Webseiten (und deren weiterführenden Unterseiten). klassifiziert oder kategorisiert, indem sie den Inhalt der Seite analysiert und in eine Liste von vordefinierten Kategorien einordnet.

1. **Methoden**

Die Arbeit wird in vier Schritten durchgeführt:

* Die Daten des Unternehmens analysieren:

Wir haben die Daten des Unternehmens erhalten, es sind URLs, die bereits kategorisiert wurden und als Kategorien haben wir zum Beispiel: Pornografie, Shopping, Religionen, Nachrichten, Computerspiele usw. Wir müssen nun eine Methode finden, um den Inhalt der Webseiten zu analysieren.

* Dataset Erstellen:

In diesem Schritt müssen wir die Webseiten analysieren und die relevantesten Daten behalten, dann die Features auswählen und sie extrahieren, um sie im Dataset zu speichern.

* Modellauswahl

Nachdem das Dataset erstellt wurde, können wir nun ein Modell für die Klassifizierung auswählen. Wir haben mehrere Möglichkeiten: Wir können entweder einen Naïve-Bayes-Algorithmus, einen Decision Tree oder eine Support Vector Machine (SVM) verwenden. Noch besser ist es, die verschiedenen Klassifikationsalgorithmen zu vergleichen und denjenigen mit der höchsten Punktzahl auszuwählen.

* Implementierung des Modells

Hier müssen wir nur unser Modell mit einer neuen URL testen, um zu sehen, ob es wirklich funktioniert.

**Checkliste für Machine-Learning dieses Projekts**

1. **Aufgabenstellung und gesucht nach der Gesamtsituation**

Das Geschäftsziel dieses Projekts ist es, ein System zu entwickeln, mit dem wir herausfinden können, zu welcher Kategorie eine Website gehört, und zwar anhand der Informationen, die auf dieser Website verfügbar sind, sowie anhand einer Liste von Kategorien, die wir zuvor zusammengestellt haben.

Unsere Lösung bzw. das endgültige System wird wie folgt verwendet: Da der Benutzer die Kategorie einer Website bereits über ihre URL kennt, kann er den Zugang zu einer bestimmten Website einfach akzeptieren oder blockieren. Das System analysiert den Inhalt der Website und interne Links, die zu anderen Webseiten führen, um die richtige Kategorie zu finden. In den Firewalls besteht die Möglichkeit über einen Reverse-Proxy den Zugang zu Webseiten anhand ihrer Kategorien zu unterbinden. Die zu unterbindenden Kategorien werden dabei vom Administrator konfiguriert. Wird eine Webseite aufgerufen, für die noch keine Kategorie im lokalen Cache gespeichert ist, wird die Kategorie von unserem Filter-Server erfragt. Für den eigentlichen Benutzer ist dieser Vorgang transparent, also nicht sichtbar.

Es besteht ein begrenzter Datensatz aus bekannten Kategorien für eine vergleichsweise überschaubare Anzahl von Webseiten, die zur Abfrage vom Filter-Server zur Verfügung stehen. Der Datensatz kann jederzeit beliebig erweitert werden. Problematisch ist jedoch, wie die Kategorien festgelegt werden. Da keine automatische Komponente vorhanden ist bleibt nahezu nur das manuelle Sichten und einordnen, was unter dem Gesichtspunkt der Wirtschaftlichkeit (und weiterer Gesichtspunkte) problematisch ist.

Um dieses Problem zu lösen, werden wir uns des supervised learning bedienen, denn um das Modell zu trainieren, werden wir ihm Daten zur Verfügung stellen, die eine Liste von Domainnamen mit jeweils einer zugeordneten Kategorie sind. Genauer gesagt werden wir die Klassifizierung verwenden, die eine Aufgabe in einem Problem des überwachten Lernens in Machine Learning ist.

Die Ergebnisse werden durch die Genauigkeit gemessen, die ein Maß für die Leistung eines Klassifikationsmodells im maschinellen Lernen ist. Mit anderen Worten, es ist der Prozentsatz der richtigen Antworten, die das Modell gemacht hat. Zunächst soll sie 75% betragen, im Laufe der Entwicklung aber auf 95% oder mehr steigen.

Anhand der prozentualen Genauigkeit der, durch das trainierte Modell getroffenen, Kategorisierung der untersuchten Webseiten.

Das Qualitätsmaß stellt sprichwörtlich das Ziel dar. Eine hohe Zuverlässigkeit bei der Kategorisierung des analysierten Inhalts ist für den Dienst unabdingbar.

Die minimale zum Erreichen des Geschäftsziels nötige Leistung ist Zunächst > 70% wobei die finale Lösung über 90% oder besser liegen sollte.

Die Sammlung der Daten erfordert ein wenig Wissen über das Scrapping und die Funktionsweise einer Webseite. Das habe ich bereits in einem der Projekte getan und werde seine Fähigkeiten auch in diesem Projekt nutzen. Für den Rest der Arbeit ist es ein bisschen neu für mich, also werde ich ein bisschen recherchieren.

Es ist klar, dass die manuelle Lösung dieses Problems sehr mühsam und zeitaufwendig wäre, denn für jede Website, die eröffnet wird, wird nach den häufigsten Wörtern gesucht, die auf dieser Website vorkommen, und dann wird eine große Datenbank erstellt, in der die Wörter, die eine Kategorie definieren, gespeichert werden. Um also eine Website zu kategorisieren, wird diese Datenbank nach dem Vorkommen der Wörter durchsucht.

1. **Datenbeschaffung**

Für dieses Projekt wurden die Daten bereitgestellt, es handelt sich eigentlich um eine Textdatei, die eine Liste von 1 185 685 Zeilen enthalt und 206.2 MB als Kapazität hat. Sie besteht aus 257 verschiedenen Kategorien. Aber wir wollen URL haben, um auf den Inhalt von Webseiten zugreifen zu können.

Da wir URLs brauchen, müssen wir ein Programm schreiben, das jeden Domainnamen liest, ihn in eine Url umwandelt und dann eine Anfrage vom Client an den Server stellt, um herauszufinden, ob der Server die Anfragen des Clients bearbeiten kann, indem er ihm das Web(Content)-Seite liefert. Nur diejenigen, deren HTTP-Response-Code 200 beträgt, werden behalten und in das Dataset aufgenommen. Es gibt auch weitere HTTP-Response-Codes, die eine Erreichbarkeit der aufgerufenen Webseite nicht ausschließen. letztendlich erhalten wir eine kleinere Liste von eindeutigen URLs, die einen HTTP-Response-Code von 200 haben. Nachdem das Modell trainiert wurde und ausgehend von einer URL oder einem URI, der von einem Menschen oder einer Maschine als Ausgangspunkt übermittelt wurde, muss der Inhalt der URL abgefragt und analysiert werden. Wenn im Inhalt weitere URLs identifiziert werden, muss derselbe Prozess rekursiv auf diese angewendet werden. Tatsächlich könnte das so in Gang gesetzte System innerhalb seiner Betriebsdauer ein offenes Ende haben, da weitere eindeutige URLs ohne vorherige Analyse identifiziert werden können.

Um dieses Vorprogrammierungs-, Transformations- und sogar Modelltrainingprogramm durchzuführen, benötigt man selbstverständlich eine Arbeitsumgebung mit ausreichender Leistung und ausreichend Speicherplatz. In unserem Fall haben wir einen Desktop-Computer mit Ubuntu 22.04.1 LTS, einem i8-Prozessor mit einer Frequenz von 3,7 GHz und einem Speicherplatz von 1,5 TB verwendet.

Anschließend werden die frisch generierten Daten in einem Dataframe (eine Klasse der Bibliothek Pandas) gespeichert, was uns die einfache Manipulation der Daten ermöglicht. Die Daten mit leeren Spalten werden aus dem Dataset entfernt und wir werden die Größe und den Typ der Daten überprüfen müssen. Schließlich werden wir einen Testdatensatz für unsere zukünftigen Tests erstellen.

1. **Bearbeitung**

Zuerst musste ich eine Methode schreiben, mit der ich die Datei lesen konnte, die die verschiedenen Domainnamen und die dazugehörige Kategorie enthielt. Da ich eine URL und keinen Domainnamen wollte, musste ich die Domainnamen in URLs umwandeln und dann überprüfen, ob ich eine Antwort erhalte, indem ich eine Anfrage an den Server richte, daher der Http-Code 200. All dies wurde mithilfe von Threads parallelisiert, um die Programmausführung zu beschleunigen. Das Ganze wurde in einer json-Datei gespeichert.

Als nächstes ging es darum, den Inhalt der Webseiten zu erhalten. Dies war mit Hilfe der Beautifulsoup-Bibliothek möglich, und zwar für statische Seiten, die keine menschliche Präsenz (z. B. einen Klick auf die Schaltfläche) benötigen, um ihren Inhalt zu erhalten. Bei dynamischen Websites, bei denen die Anwesenheit eines Menschen erforderlich ist, musste ich die Selenium-Bibliothek verwenden, um diese Interaktion zu ermöglichen (ein automatischer Klick auf die Schaltfläche).

Das Ergebnis dieser Extraktion sollte in einem Datenframe gespeichert werden, in dem der Name, die Kategorie, die URL, die Wörter, die auf der Website enthalten sind, und die Sprache, die als Spalte des Datenframes verwendet wird, enthalten sind. Der Datendatensatz soll in eine csv-Datei umgewandelt werden.

Im nächsten Kapitel (Kapitel 2) werden wir die Grundlagen einiger Konzepte zur Klassifizierung von Webseiten, Data Mining und Natural Language Processing erläutern. Anschließend werden wir in Kapitel 3 die Datensammlung untersuchen, d.h. die Techniken, die es uns ermöglichen, einen Datenbestand für die zukünftige Analyse zu erhalten. Diese Daten werden natürlich verarbeitet und analysiert, bevor sie in den Modellen des maschinellen Lernens verwendet werden, was in Kapitel 4 geschieht. In diesem Kapitel werden wir die verschiedenen Modelle des maschinellen Lernens zur Klassifizierung untersuchen. Kapitel 5 befasst sich mit dem Experimentieren, der Auswahl des besten Modells und der Präsentation der Ergebnisse, und das letzte Kapitel enthält eine Zusammenfassung und einen Ausblick auf die Zukunft.