Текстэн мэдээллийн сэтгэл хөдлөлийг статистик аргаар ангилах нь: Гэнэн Байес болон Ложистик регрессийн харьцуулсан шинжилгээ

Л.Анужин, Г.Батням, Э.Мэндбаяр, Б.Хувьзаяа, А.Эрхэс

2025 оны 11-р сарын 28

Энэхүү төслийн хүрээнд нийгмийн сүлжээний “Sentiment140” өгөгдөл дээр хяналттай машин сургалтын (Supervised Learning) аргуудыг ашиглан хэрэглэгчийн сэтгэл хөдлөлийг эерэг болон сөрөг гэж ангилав. Бид өгөгдлийн түүвэр дээр Гэнэн Байесын (Naive Bayes) аргаар постериор магадлалыг тооцоолох болон Ложистик регрессийн (Logistic Regression) ложит загварыг харьцуулан туршив. Судалгааны үр дүнд Ложистик регресс загвар нь [X]%-ийн нарийвчлалтайгаар текстийн утгыг зөв таньж, харьцуулсан загвараас илүү оновчтой үр дүн үзүүлсэн байна.

Агуулга

Шаардлагатай багцууд

Шаардлагатай багцуудыг дараах байдлаар урьдчилан суулгана.

```
pip install streamlit pandas numpy scikit-learn matplotlib seaborn plotly
```

1 Удиртгал

Нийгмийн сүлжээ, тэр дундаа Twitter (одоогийн X) платформ нь олон нийтийн санаа бодол, хандлагыг илэрхийлэх томоохон эх сурвалж юм. Хэрэглэгчид өөрсдийн үзэл бодлоо богино хэмжээний текст буюу “жиргээ” (tweet) хэлбэрээр илэрхийлдэг. Эдгээр их хэмжээний өгөгдөлд дүн шинжилгээ хийх нь бизнесийн байгууллага болон судлаачдад хэрэглэгчийн сэтгэл ханамжийг үнэлэхэд чухал ач холбогдолтой боловч сая сая жиргээг хүний оролцоотойгоор уншиж ангилах боломжгүй юм.

Иймд энэхүү төслийн ажлаар бид Стэнфордын их сургуулийн судлаачдын боловсруулсан **Sentiment140** өгөгдлийн санд тулгуурлан, сэтгэл хөдлөлийг автоматаар таньж, эерэг болон сөрөг гэж ангилах статистик загварыг хөгжүүлэв. Судалгаанд **Гэнэн Байесын алгоритм** (Naive Bayes) болон **Ложистик регресс** (Logistic Regression) гэсэн хоёр өөр статистик аргыг ашиглаж, тэдгээрийн үр дүнг харьцуулан шинжиллээ.

2 Өгөгдөл ба Боловсруулалт

2.1 Өгөгдлийн эх сурвалж ба бүтэц

Энэхүү судалгаанд бид Alec Go, Richa Bhayani, Lei Huang нарын (2009) цуглуулсан **Sentiment140** [1] өгөгдлийн санг ашиглав. Уг өгөгдлийн сан нь Twitter API ашиглан цуглуулсан 1.6 сая жиргээнээс бүрдэнэ. Өгөгдлийн хаягжуулалт (Labeling) нь “Зайнаас хяналттай сургалт” (Distant Supervision) аргаар хийгдсэн. Тодруулбал:

- Сэтгэл хөдлөл илэрхийлсэн тэмдэгт `:)` агуулсан жиргээг **“Эерэг” (4)**,
- `:(` тэмдэгт агуулсан жиргээг **“Сөрөг” (0)** гэж автоматаар ангилсан.

Сургалтын явцад загвар зөвхөн тэмдэгтийг цээжлэхээс сэргийлж, жиргээний текстээс эдгээр тэмдэгтүүдийг (emoticons) устгасан байдаг. Бидний ашигласан өгөгдлийн бүтэц дараах байдалтай байна:

Хүснэгт 1: Өгөгдлийн бүтцийн жишээ (sample_set.csv)

Ангилал (Target)	Жиргээ (Text)
4	Listening to Black Eyed Peas. …
4	Sorry haven’t posted in a while now.. Been busy busy busy.. WORKING …
4	**?** thanks! I’ll tell my mom. I refuse to go to choir after …
0	No good movies …
0	**?** Eh. Nothing much. And I bet it will. (: I’m about to be o…

2.2 Өгөгдлийн түүвэрлэлт (Sampling)

Sentiment140 өгөгдлийн сан нь нийт 1.6 сая мөр бүхий их хэмжээний өгөгдлийг агуулдаг. Энэхүү төслийн хүрээнд тооцооллын нөөц болон хугацааг хэмнэх зорилгоор бид “Санамсаргүй түүвэрлэлт” (Simple Random Sampling)-ийн аргыг ашиглав.

Бид Python хэлний `sample()` функцийг ашиглан нийт өгөгдлөөс **50,000** жиргээг санамсаргүй байдлаар сонгон авч сургалтад ашигласан. Туршилтын үр дүн дахин давтагдах (reproducible) боломжтой байхын тулд `random_state=42` тохиргоог ашигласан болно.

```
# Бүтэн өгөгдлийг унших (1.6 сая мөр)
df_full = pd.read_csv("training.1600000.processed.noemoticon.csv",
                      encoding="latin-1", header=None)

# 50,000 мөрийг санамсаргүйгээр түүвэрлэх (Seed = 42)
df = df_full.sample(n=50000, random_state=42)
```

2.3 Өгөгдлийг цэвэрлэх ба Бэлтгэх (Preprocessing)

Түүхий өгөгдөл (Raw Data) нь дүн шинжилгээ хийхэд саад болох олон төрлийн “шуугиан” (noise) агуулдаг. Таны багийн бичсэн кодын дагуу бид Python хэлний `re` (Regular Expression) санг ашиглан текстэн өгөгдлийг дараах байдлаар цэвэрлэлээ.

1. Жижиг үсэгт шилжүүлэх: `lower()` функц ашиглан бүх текстийг жижиг үсэг болгов.
2. URL болон Хэрэглэгчийн нэр: `http` болон `@username` хэлбэрийн холбоосуудыг устгав.
3. HTML тэмдэгт: `&` гэх мэт кодыг `and` гэх мэт энгийн үгээр солив.
4. RT (Retweet): Жиргээг дамжуулсан тэмдэглэгээг хасав.
5. Тусгай тэмдэгт: Үсэг болон тооноос бусад тэмдэгтүүдийг устгав.

```
import re

def clean_tweet(text):
    text = str(text).lower()                              # Жижиг үсэг болгох
    text = re.sub(r"http\S+|www\.\S+", "", text)          # URL устгах
    text = re.sub(r"@\w+", "", text)                      # @username устгах
    text = re.sub(r"&amp;", "and", text)                  # &amp → and болгох
    text = re.sub(r"rt[\s]+", "", text)                   # RT (retweet) устгах
    text = re.sub(r"[^a-z0-9\s]", " ", text)              # Тусгай тэмдэгт устгах
    text = re.sub(r"\s+", " ", text).strip()              # Илүүдэл зай цэвэрлэх
    return text

# Өгөгдлийг цэвэрлэх (Жишээ өгөгдөл дээр)
df_sample["clean_text"] = df_sample["text"].astype(str).apply(clean_tweet)

# Target хувьсагчийг 0 (Сөрөг) ба 1 (Эерэг) болгож хөрвүүлэх
# (Sentiment140 өгөгдөлд 4 нь Эерэг байдаг)
if set(df_sample["target"].unique()) == {0, 4}:
    df_sample["target"] = df_sample["target"].map({0: 0, 4: 1})
```

Цэвэрлэгээ хийсний дараах үр дүнг доорх хүснэгтэд харуулав.

Хүснэгт 2: Цэвэрлэсэн өгөгдлийн жишээ

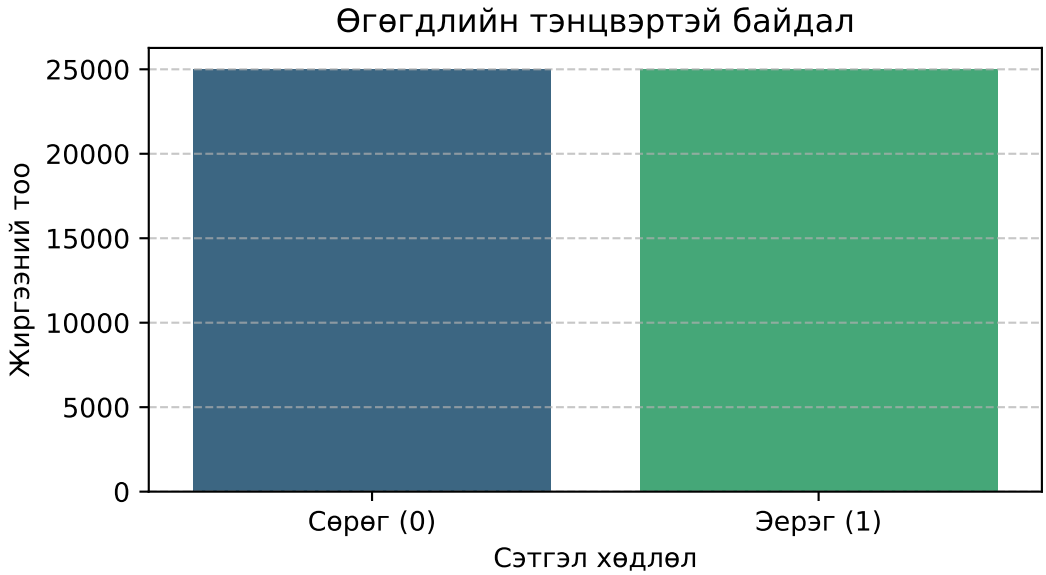
Ангилал	Эх текст	Цэвэр текст
1	Listening to Black Eyed Peas. …	listening to black eyed peas…
1	Sorry haven’t posted in a while now.. Be…	sorry haven t posted in a while now been…
1	**?** thanks! I’ll tell my mom. …	thanks i ll tell my mom i refuse to go t…
0	No good movies …	no good movies…
0	**?** Eh. Nothing much. And I bet…	eh nothing much and i bet it will i m ab…

2.4 Өгөгдлийн шинжилгээ (EDA)

Сургалтад ашиглаж буй өгөгдлийн тэнцвэртэй байдлыг шалгах нь чухал юм. Бидний ашиглаж буй түүвэр өгөгдөлд Сөрөг (0) болон Эерэг (1) ангилал хэрхэн тархсаныг доорх диаграммаар харуулав.

```
import matplotlib.pyplot as plt
import seaborn as sns

# Графикийн хэмжээг тохируулах
plt.figure(figsize=(6, 3))

# Countplot зурах
sns.countplot(x=df_sample["target"], palette="viridis")

# Тэнхлэгийн нэрс
plt.xticks([0, 1], ["Сөрөг (0)", "Эерэг (1)"])
plt.xlabel("Сэтгэл хөдлөл")
plt.ylabel("Жиргээний тоо")
plt.title("Өгөгдлийн тэнцвэртэй байдал")
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```

Зураг 1: Сэтгэл хөдлөлийн ангиллын тархалт



Диаграммаас харахад өгөгдөл тэнцвэртэй байгаа нь загвар аль нэг тал руу хэт хазайх (bias) эрсдэлгүйг харуулж байна.

3 Судалгааны арга зүй

Энэхүү судалгаанд бид текстэн мэдээллийг ангилахдаа **хяналттай машин сургалтын** (Supervised Machine Learning) түгээмэл аргууд болох **Гэнэн Байес** болон **Ложистик регресс** загваруудыг ашиглав.

3.1 Векторжуулах арга (Feature Extraction)

Машин сургалтын загварууд нь текстийг шууд ойлгох боломжгүй тул бид тэдгээрийг тоон хэлбэрт шилжүүлэх шаардлагатай. Үүнд дараах хоёр аргыг ашиглав:

1. **CountVectorizer (Bag of Words):** Үгсийн давтамжийг тоолох энгийн арга.
2. **TF-IDF (Term Frequency-Inverse Document Frequency):** Үгсийн давтамжийг нийт баримт бичигт эзлэх хувиар жинлэн үнэлэх арга.

3.2 Гэнэн Байесын алгоритм (Naive Bayes)

Энэ нь Байесын зарчимд суурилсан ангиллын алгоритм юм. Лекц XVI (хуудас 147)-д дурдсанаар юмс үзэгдлийг хамгийн их **постериор магадлалтай** ангид хуваарилдаг:

$$\text{argmax}_k \, P(C_k) \prod_{i=1}^{n} P(X_i|C_k)$$

Энд:

- $P(C_k)$: Приор магадлал (Тухайн ангиллын ерөнхий магадлал).
- $P(X_i|C_k)$: Үнэний хувь (Likelihood) буюу тухайн ангилалд X_i шинж чанар (үг) илрэх магадлал.
- $\prod$: Үржвэр (Бүх үгсийн магадлалыг хооронд нь үржүүлж байна).

Бид энэхүү төсөлд `MultinomialNB` хувилбарыг ашигласан.

3.3 Ложистик регресс (Logistic Regression)

Ложистик регресс нь үр дүнг 0-ээс 1-ийн хооронд магадлалаар илэрхийлдэг. Лекц XVI (хуудас 144)-д дурдсанаар энэ нь шугаман регрессийн утгыг **ложистик функц** ашиглан хувиргадаг:

$$p = \frac{e^{a+bX}}{1 + e^{a+bX}}$$

Энд:

- p: Жиргээ эерэг байх магадлал.
- $a + bX$: Текстэн өгөгдлийн шинж чанаруудын шугаман хослол.
- e: Натурын логарифмын суурь.

Энэ арга нь хувьсагчдын нарийн хамаарлыг илрүүлэхдээ сайн ажилладаг бөгөөд үр дүнг магадлалаар илэрхийлдэг давуу талтай.

4 Туршилт ба Үр дүн

Бид боловсруулсан 50,000 мөр өгөгдлийг сургалтын (Training set) болон тестийн (Test set) олонлогт **70:30** харьцаатайгаар санамсаргүйгээр хуваасан. Ингэхдээ ангиллын тэнцвэртэй байдлыг хадгалахын тулд `stratify` параметрийг ашиглав.

```
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Өгөгдлийг бэлтгэх (өмнөх хэсэгт цэвэрлэсэн df_sample-г ашиглана)
# clean_text баганад NaN байхгүй эсэхийг шалгаад string болгох
X = df_sample["clean_text"].fillna("").astype(str)
y = df_sample["target"]

# 70% сургалт, 30% тест
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42
)

# 1. Гэнэн Байес (Naive Bayes) Pipeline
nb_model = Pipeline([
    ("vect", CountVectorizer(max_features=15000, ngram_range=(1,2))),
    ("clf", MultinomialNB())
])

# 2. Ложистик Регресс (Logistic Regression) Pipeline
lr_model = Pipeline([
    ("vect", TfidfVectorizer(max_features=20000, ngram_range=(1,2))),
    ("clf", LogisticRegression(max_iter=1000, solver="liblinear"))
])

# Загваруудыг сургах
nb_model.fit(X_train, y_train)
lr_model.fit(X_train, y_train)

# Таамаглал дэвшүүлэх
y_pred_nb = nb_model.predict(X_test)
y_pred_lr = lr_model.predict(X_test)

# Нарийвчлал тооцох
acc_nb = accuracy_score(y_test, y_pred_nb)
acc_lr = accuracy_score(y_test, y_pred_lr)
```

4.1 Загваруудын харьцуулалт

Туршилтын үр дүнд хоёр загварын нарийвчлал (Accuracy) дараах байдалтай гарав. Бидний таамаглаж байснаар Ложистик регресс загвар нь илүү өндөр үр дүн үзүүллээ.

Хүснэгт 3: Загваруудын нарийвчлалын харьцуулалт

Загвар	Нарийвчлал (Accuracy)	Тайлбар
Гэнэн Байес	76.91%	Хурдан ажилладаг боловч үгсийн хамаарлыг тооцдоггүй.

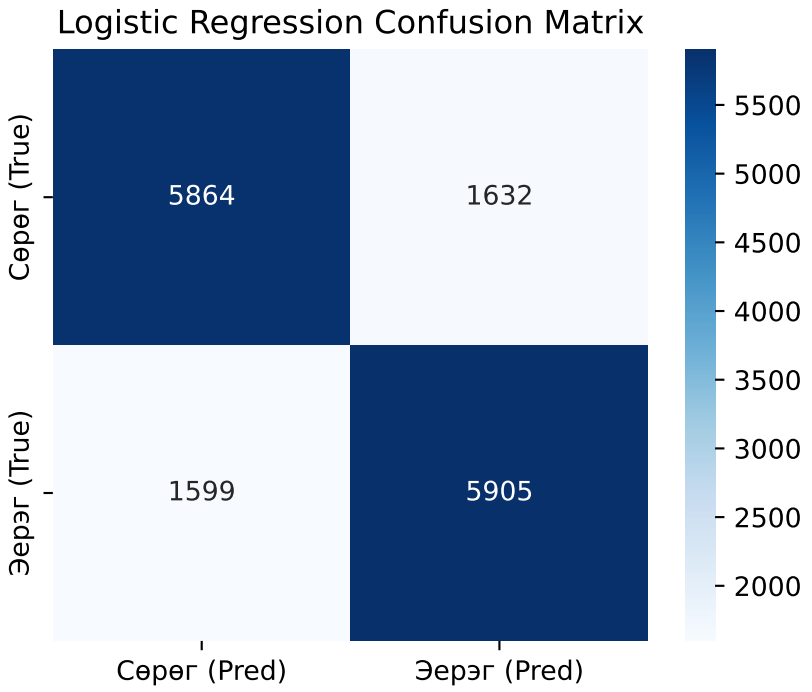
Загвар	Нарийвчлал (Accuracy)	Тайлбар
Ложистик Регресс	78.46%	TF-IDF жинлэлт ашигласан тул илүү нарийвчлалтай.

4.2 Төөрөгдлийн матриц (Confusion Matrix)

Загвар хэрхэн ажилласныг илүү нарийвчлан харахын тулд өндөр үр дүн үзүүлсэн Ложистик Регресс загварын Төөрөгдлийн матрицыг байгуулъя. Энэ нь загвар “Сөрөг” болон “Эерэг” жиргээг хэр зөв ялгаж байгааг харуулна.

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# Матриц тооцоолох
cm = confusion_matrix(y_test, y_pred_lr)

# График зурах
plt.figure(figsize=(5, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=["Сөрөг (Pred)", "Эерэг (Pred)"],
            yticklabels=["Сөрөг (True)", "Эерэг (True)"])
plt.title("Logistic Regression Confusion Matrix")
plt.show()
```

Зураг 2: Ложистик Регресс загварын Төөрөгдлийн матриц



Дүгнэлт

Багийн гишүүдийн оролцоо

Ашигласан материал

Хавсралт: Програмын код

[1] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” *CS224N project report, Stanford*, vol. 1, no. 12, p. 2009, 2009.