Зээл эргэн төлөх эрсдэлийн үнэлгээ

Э.Эрдэнэтуяа /22B1NUM5619/ С.Баярбилэг /24B1NUM2607/

2025 оны 12-р сарын 5

Энэхүү ажилаар зээлийн хүсэлт гаргасан хүн зээлээ буцааж төлөх эсэхийг урьдчилан таамаглах зорилгоор Kaggle-ээс авсан Credit Risk Dataset өгөгдөл дээр Gaussian Naive Bayes загварыг сургаж, түүний гүйцэтгэлийг тооцно.

Агуулга

1 Оршил 2

2 Өгөгдөл 2
- 2.1 Өгөгдлийн эх сурвалж . . . 2
- 2.2 Өгөгдлийн бүтэц . . . 2
- 2.3 Дутуу мэдээлэл . . . 3

3 Өгөгдлийн шинжилгээ 4
- 3.1 Зээл төлөх байдлын харьцаа . . . 4
- 3.2 Зээлийн зориулалт болон зээл төлөх байдал . . . 4
- 3.3 Тоон мэдээллийн шинжилгээ . . . 5

4 Өгөгдлийн боловсруулалт 6
- 4.1 1-р алхам: Дутуу мэдээлэл бөглөх . . . 7
- 4.2 2-р алхам: Ангиллын мэдээллийг тоо болгох . . . 7
- 4.3 3-р алхам: Сургалт болон тестийн багцад хуваах . . . 7
- 4.4 4-р алхам: Өгөгдлийг мастабчлах . . . 7

5 Naive Bayes загвар 8
- 5.1 Товч тайлбар . . . 8
- 5.2 Загварыг сургах . . . 8

6 Үр дүн 9
- 6.1 Загварын гүйцэтгэл . . . 9
- 6.2 Алдааны матриц . . . 10
- 6.3 ROC Curve . . . 12
- 6.4 Дэлгэрэнгүй тайлан . . . 13

7 Дүгнэлт 14
- 7.1 1. Загварын ерөнхий гүйцэтгэл . . . 14
- 7.2 2. Давуу тал . . . 14
- 7.3 3. Хязгаарлалт . . . 14

7.4 4. Банкны хувьд хэрхэн ашиглах вэ? 14
7.5 5. Цаашид сайжруулах арга 14
7.6 6. Эцсийн санал 15

Програмын код 15

Багийн гишүүдийн хувь нэмэр 15

Ашигласан материал 15

1 Оршил

Банкууд зээл олгохдоо хамгийн их санаа зовдог асуудал бол “Энэ хүн зээлээ буцааж төлөх үү?” гэсэн асуулт юм. Хэрэв зээлдэгч зээлээ төлөхгүй бол банкинд алдагдал учирна. Иймээс зээл олгохоосоо өмнө хүний зээл төлөх чадварыг урьдчилан мэдэх нь маш чухал [1]. Өгөгдөл боловсруулалтын аргуудыг ашиглан энэ асуултыг шийдэхээр **Naive Bayes** [2] алгоритмийн аргыг ашиглана.

Бидний судалгааны зорилго:

1. Зээлийн өгөгдлийг нарийвчлан судлах
2. Naive Bayes аргыг ашиглан зээл төлөх эсэхийг таамаглах
3. Бидний загварын үр дүнг үнэлэх
4. Энэ аргыг практикт хэрхэн ашиглах талаар санал гаргах

2 Өгөгдөл

2.1 Өгөгдлийн эх сурвалж

Kaggle вэб сайтаас “Credit Risk Dataset” [3] нэртэй өгөгдлийн багцыг авсан. Энэ өгөгдөлд 32,581 хүний зээлийн мэдээлэл багтсан байна. Өгөгдөл дотор нийт 12 төрлийн хувьсагч байна.

```
df = pd.read_csv('data/credit_risk_dataset.csv')
print(f"Өгөгдлийн хэмжээ: {df.shape[0]} хүн, {df.shape[1]} мэдээлэл")
```

```
Өгөгдлийн хэмжээ: 32581 хүн, 12 мэдээлэл
```

2.2 Өгөгдлийн бүтэц

Өгөгдөлд доорх мэдээллүүд багтсан байна:

- **person_age**: Зээлдэгчийн нас
- **person_income**: Жилд олох орлого (ам.доллараар)
- **person_home_ownership**: Байр сууцтай эсэх (өөрийнх, түрээс гэх мэт)
- **person_emp_length**: Хэдэн жил ажилласан
- **loan_intent**: Зээлийг юунд зориулж байгаа (гэр авах, суралцах гэх мэт)
- **loan_grade**: Зээлийн чанар (A-ээс G хүртэл, A нь хамгийн сайн)
- **loan_amnt**: Зээлийн хэмжээ (ам.доллараар)
- **loan_int_rate**: Зээлийн хүү (хувиар)

- **loan_status**: **Үндсэн асуулт** - зээлээ төлсөн эсэх (0=төлсөн, 1=төлөөгүй)
- **loan_percent_income**: Зээлийн хэмжээ нь орлогын хэдэн хувтай
- **cb_person_default_on_file**: Өмнө нь зээл төлөөгүй түүх
- **cb_person_cred_hist_length**: Зээлийн түүх жилийн хугацаа

Хүснэгт 1: Өгөгдлийн жишээ (эхний 8 хүний мэдээлэл)

	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb_person_default_on_file	cb_person_cred_hi
0	35000	16.02	1	0.59	Y	3
1	1000	11.14	0	0.10	N	2
2	5500	12.87	1	0.57	N	3
3	35000	15.23	1	0.53	N	2
4	35000	14.27	1	0.55	Y	4
5	2500	7.14	1	0.25	N	2
6	35000	12.42	1	0.45	N	3
7	35000	11.11	1	0.44	N	4

2.3 Дутуу мэдээлэл

```
missing_data = df.isnull().sum()
missing_pct = (missing_data / len(df)) * 100

missing_df = pd.DataFrame({
    'Хувьсагч': missing_data.index,
    'Дутуу тоо': missing_data.values,
    'Хувь (%)': np.round(missing_pct.values, 2)
})
missing_df = missing_df[missing_df['Дутуу тоо'] > 0].sort_values('Дутуу тоо', ascending=Fals

if len(missing_df) > 0:
    print("Дутуу мэдээлэлтэй хэсгүүд:")
    print(missing_df.to_string(index=False))
else:
    print("Бүх мэдээлэл бүрэн байна")
```

```
Дутуу мэдээлэлтэй хэсгүүд:
         Хувьсагч  Дутуу тоо  Хувь (%)
    loan_int_rate       3116      9.56
person_emp_length        895      2.75
```

Дутуу мэдээллүүдийг дараах байдлаар бөглөсөн:

- Тоон мэдээлэл (нас, орлого гэх мэт): дундаж утгаар бөглөх
- Ангиллын мэдээлэл (байрны төрөл гэх мэт): хамгийн их гардаг утгаар бөглөх

3 Өгөгдлийн шинжилгээ

3.1 Зээл төлөх байдлын харьцаа

```
fig, ax = plt.subplots(1, 2, figsize=(12, 4))

# Баганан график
counts = df['loan_status'].value_counts()
colors = ['#2ecc71', '#e74c3c']
ax[0].bar(['Төлсөн', 'Төлөөгүй'], counts.values, color=colors, alpha=0.8, edgecolor='black')
ax[0].set_title('Зээл төлөх байдлын тоо', fontsize=12, fontweight='bold')
ax[0].set_ylabel('Хүний тоо', fontsize=11)
ax[0].grid(axis='y', alpha=0.3)
for i, v in enumerate(counts.values):
    ax[0].text(i, v + 500, str(v), ha='center', fontweight='bold')

# Дугуй график
default_pct = df['loan_status'].value_counts(normalize=True) * 100
ax[1].pie(default_pct, labels=['Төлсөн', 'Төлөөгүй'], autopct='%1.1f%%',
          colors=colors, startangle=90, textprops={'fontsize': 11, 'fontweight': 'bold'})
ax[1].set_title('Зээл төлөх байдлын хувь', fontsize=12, fontweight='bold')

plt.tight_layout()
fig.savefig(os.path.join(output_dir, "target_distribution.pdf"), bbox_inches='tight')
plt.show()
```

Зураг 1: Зээлээ төлсөн болон төлөөгүй хүмүүсийн харьцаа

Өгөгдлөөс харахад нийт хүмүүсийн **21.8%** нь зээлээ төлж чадаагүй байна. Энэ нь тэнцвэргүй өгөгдөл (өдийг олон, нөгөөг цөөн) гэж авч үзнэ.

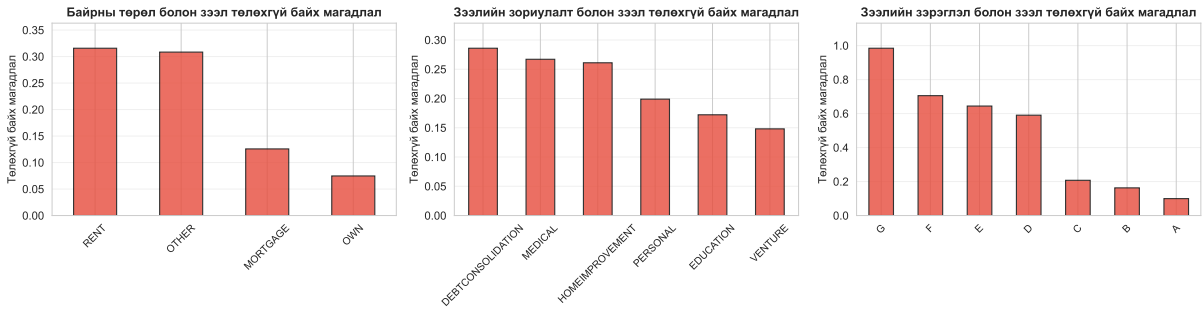
3.2 Зээлийн зориулалт болон зээл төлөх байдал

```
cat_cols_viz = ['person_home_ownership', 'loan_intent', 'loan_grade']
titles = ['Байрны төрөл', 'Зээлийн зориулалт', 'Зээлийн зэрэглэл']

fig, axes = plt.subplots(1, 3, figsize=(15, 4))
```

```
for idx, (col, title) in enumerate(zip(cat_cols_viz, titles)):
    default_rate = df.groupby(col)['loan_status'].mean().sort_values(ascending=False)
    bars = default_rate.plot(kind='bar', ax=axes[idx], color='#e74c3c', alpha=0.8, edgecolor
    axes[idx].set_title(f'{title} болон зээл төлөхгүй байх магадлал', fontsize=11, fontweigh
    axes[idx].set_ylabel('Төлөхгүй байх магадлал', fontsize=10)
    axes[idx].set_xlabel('')
    axes[idx].tick_params(axis='x', rotation=45, labelsize=9)
    axes[idx].grid(axis='y', alpha=0.3)
    axes[idx].set_ylim(0, max(default_rate) * 1.15)

plt.tight_layout()
fig.savefig(os.path.join(output_dir, "categorical_analysis.pdf"), bbox_inches='tight')
plt.show()
```



Зураг 2: Зээлийн зориулалтанд үндэслэн зээл төлөлтөнд үзүүлэх нөлөө

Дээрх графикуудаас:

- **Зээлийн зэрэглэл**: Муу зэрэглэлтэй (F, G) хүмүүс зээлээ төлөхгүй байх магадлал өндөр байна
- **Зээлийн зориулалт**: Зарим зориулалт (жишээ нь: бизнес эсвэл орон сууц) нь эрсдэлтэй байна
- **Байрны эзэмшил**: Өөрийн байртай хүмүүс зээлээ илүү сайн төлдөг

3.3 Тоон мэдээллийн шинжилгээ

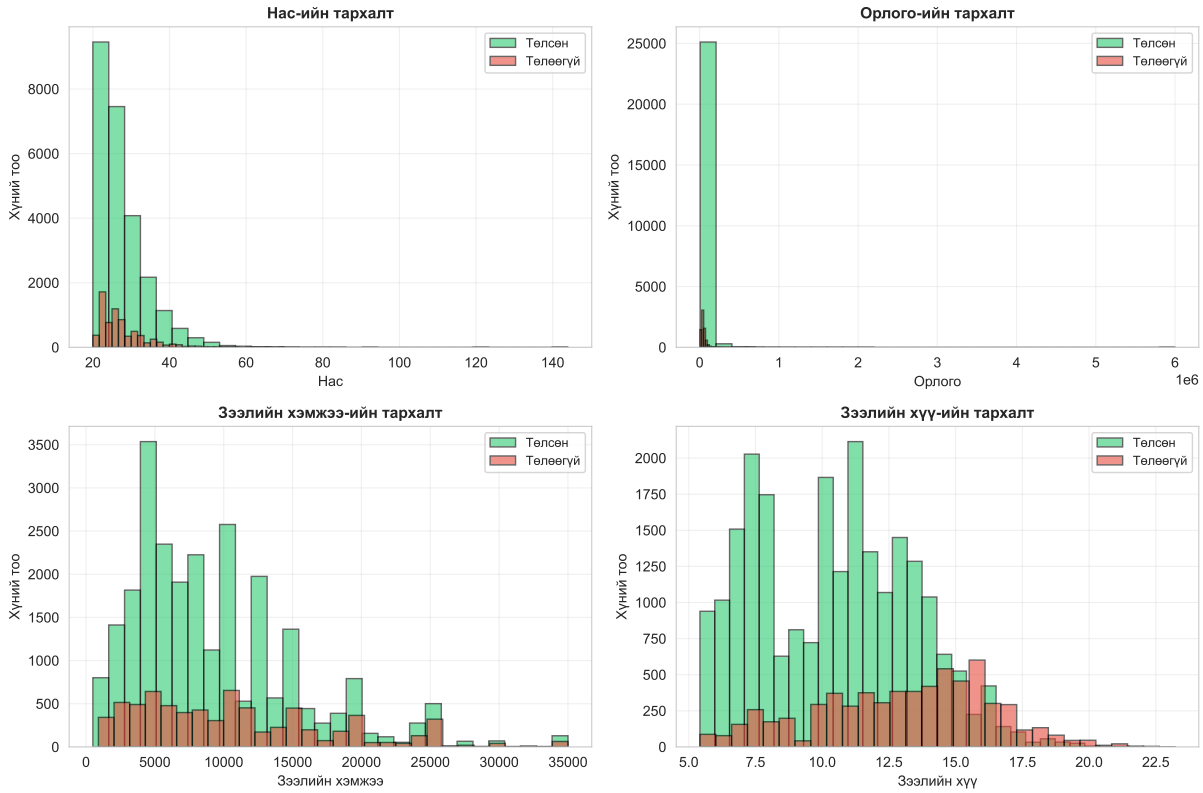
```
num_cols_viz = ['person_age', 'person_income', 'loan_amnt', 'loan_int_rate']
titles = ['Нас', 'Орлого', 'Зээлийн хэмжээ', 'Зээлийн хүү']

fig, axes = plt.subplots(2, 2, figsize=(12, 8))
axes = axes.ravel()

for idx, (col, title) in enumerate(zip(num_cols_viz, titles)):
    # Хуваарилалтын график
    df[df['loan_status'] == 0][col].hist(ax=axes[idx], bins=30, alpha=0.6,
                                         label='Төлсөн', color='#2ecc71', edgecolor='black
    df[df['loan_status'] == 1][col].hist(ax=axes[idx], bins=30, alpha=0.6,
                                         label='Төлөөгүй', color='#e74c3c', edgecolor='bla
    axes[idx].set_title(f'{title}-ийн тархалт', fontsize=11, fontweight='bold')
    axes[idx].set_xlabel(title, fontsize=10)
    axes[idx].set_ylabel('Хүний тоо', fontsize=10)
    axes[idx].legend(fontsize=9)
    axes[idx].grid(alpha=0.3)

plt.tight_layout()
```

```
fig.savefig(os.path.join(output_dir, "numerical_analysis.pdf"), bbox_inches='tight')
plt.show()
```



Зураг 3: Тоон хувьсагчдын тархалт (Төлсөн vs Төлөөгүй)

4 Өгөгдлийн боловсруулалт

Машин сургалтын загварт оруулахаас өмнө өгөгдлийг заавал цэвэрлэж, зохих хэлбэрт оруулах шаардлагатай тул бид дараах үндсэн 4 алхмаар боловсруулна.

Түүнээс өмнө Target ба шинж чанарыг салгаж, хувьсагчдын төрлийг тогтоосон:

```
# Өгөгдөл хуулах
df_processed = df.copy()

# Хариу болон асуултуудыг салгах
y = df_processed['loan_status']  # Хариу: Төлсөн үү?
X_raw = df_processed.drop(columns=['loan_status'])  # Асуултууд: бусад бүх мэдээлэл

# Ямар төрлийн мэдээлэл байгааг тодорхойлох
num_cols = X_raw.select_dtypes(include=['int64', 'float64']).columns
cat_cols = X_raw.select_dtypes(include=['object']).columns

print(f"Тоон мэдээлэл: {len(num_cols)} ширхэг")
print(f"Ангиллын мэдээлэл: {len(cat_cols)} ширхэг")
```

```
Тоон мэдээлэл: 7 ширхэг
Ангиллын мэдээлэл: 4 ширхэг
```

4.1 1-р алхам: Дутуу мэдээлэл бөглөх

```
# Дутуу мэдээллийг бөглөх багаж бэлтгэх
imputer_num = SimpleImputer(strategy='median')  # Тоон мэдээлэлд дундаж ашиглах
imputer_cat = SimpleImputer(strategy='most_frequent')  # Ангилалд хамгийн их гардагыг ашигла

# Бөглөх
X_raw[num_cols] = imputer_num.fit_transform(X_raw[num_cols])
X_raw[cat_cols] = imputer_cat.fit_transform(X_raw[cat_cols])

print("□ Дутуу мэдээллүүд бөглөгдлөө")
```

```
□ Дутуу мэдээллүүд бөглөгдлөө
```

4.2 2-р алхам: Ангиллын мэдээллийг тоо болгох

```
# Үг болон ангиллыг тоо болгох (One-hot encoding)
X = pd.get_dummies(X_raw, drop_first=True)

print(f"□ Боловсруулалтын дараа: {X.shape[1]} ширхэг шинж чанар үүссэн")
```

```
□ Боловсруулалтын дараа: 22 ширхэг шинж чанар үүссэн
```

4.3 3-р алхам: Сургалт болон тестийн багцад хуваах

```
# Өгөгдлийг 80% (сургалт) болон 20% (тест) болгон хуваах
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,      # 20% нь тест
    random_state=42,    # Давтагдахаар хийх
    stratify=y          # Хоёр багцад ижил харьцаатай байх
)

print(f"□ Сургалтын багц: {X_train.shape[0]} хүн ({X_train.shape[0]/len(df)*100:.1f}%)")
print(f"□ Тестийн багц: {X_test.shape[0]} хүн ({X_test.shape[0]/len(df)*100:.1f}%)")
print(f"   • Сургалтын default rate: {y_train.mean()*100:.2f}%")
print(f"   • Тестийн default rate: {y_test.mean()*100:.2f}%")
```

```
□ Сургалтын багц: 26064 хүн (80.0%)
□ Тестийн багц: 6517 хүн (20.0%)
   • Сургалтын default rate: 21.82%
   • Тестийн default rate: 21.82%
```

4.4 4-р алхам: Өгөгдлийг мастабчлах

```
# Том тоо болон жижиг тоог нэг хэмжээстэй болгох (StandardScaler)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

print("□ Өгөгдөл масштабчлагдлаа")
```

```
□ Өгөгдөл масштабчлагдлаа
```

5 Naive Bayes загвар

5.1 Товч тайлбар

Гэнэн Байесын (Naive Bayes) алгоритм нь **Байесын теорем**-д суурилсан ангиллын арга юм. Үзэгдэл ямар ангид хамаарахыг **хамгийн их постериор магадлалтай** ангийг сонгож шийддэг.

“Энэ хүн өгсөн мэдээлэл дээр үндэслээд зээлээ төлөхгүй байх магадлал хэд вэ?” гэх асуултыг Байесын томъёо шууд илэрхийлээд өгдөг:

$$P(\text{Төлөхгүй} \mid \text{Мэдээлэл}) = \frac{P(\text{Мэдээлэл} \mid \text{Төлөхгүй}) \cdot P(\text{Төлөхгүй})}{P(\text{Мэдээлэл})}$$

Энэ томъёо дараах утгатай:

- **(P(Төлөхгүй))** — Ерөнхийдөө хүмүүсийн хэдэн хувь нь зээлээ төлдөггүй вэ? (Өмнөх нийт мэдээллээс авсан *урьдчилсан магадлал*)
- **(P(Мэдээлэл □Төлөхгүй))** — Өмнө нь зээл төлөөгүй хүмүүсийн дунд ийм төрлийн мэдээлэл (нас, орлого, ажилласан жил, өмнөх кредит түүх…) хэр олон давтагддаг вэ?
- **(P(Мэдээлэл))** — Ийм төрлийн мэдээлэл ер нь нийт өгөгдөлд хэр олон тохиолддог вэ? (Энэ нь хоёр ангийг харьцуулахад л хэрэгтэй тул ихэнхдээ тогтмол гэж үзнэ)

$$P(\text{Мэдээлэл} \mid \text{Төлөхгүй}) \approx \prod_j P(X_j \mid \text{Төлөхгүй})$$

гэсэн хялбар томъёо гарч ирнэ.

Gaussian Naive Bayes - Тоон хувьсагчид (age, income, loan_amnt гэх мэт) нь нормал тархалттай (дундын эргэн тойронд их, хажуу талд бага) гэж үзээд ангиллыг тооцоолдог хувилбар.

Ийм нормал хэлбэрийн тархалтын тусламжтайгаар:

$$P(X_j \mid C_k)$$

утгыг хурдан, тогтвортой тооцоолж чаддаг тул тоон мэдээлэлтэй датанд хамгийн тохиромжтой байдаг.

5.2 Загварыг сургах

```
# Gaussian Naive Bayes загвар үүсгэх
nb_model = GaussianNB()

# Сургалтын өгөгдлөөр заах
nb_model.fit(X_train_scaled, y_train)

# Тест өгөгдлийг таамаглах
y_pred_nb = nb_model.predict(X_test_scaled)  # Төлөх/Төлөхгүй
y_proba_nb = nb_model.predict_proba(X_test_scaled)[:, 1]  # Магадлал

print("□ Загвар амжилттай сургагдлаа!")
print(f"□ {X_test.shape[0]} хүний зээл төлөх байдлыг таамагласан")
```

```
□ Загвар амжилттай сургагдлаа!
□ 6517 хүний зээл төлөх байдлыг таамагласан
```

6 Үр дүн

6.1 Загварын гүйцэтгэл

```
# Үнэлгээний үзүүлэлтүүд тооцоолох
acc = accuracy_score(y_test, y_pred_nb)
prec = precision_score(y_test, y_pred_nb)
rec = recall_score(y_test, y_pred_nb)
f1 = f1_score(y_test, y_pred_nb)
auc = roc_auc_score(y_test, y_proba_nb)

performance = pd.DataFrame({
    'Үзүүлэлт': ['Accuracy', 'Precision', 'Recall', 'F1-Score', 'AUC'],
    'Утга': [acc, prec, rec, f1, auc],
    'Тайлбар': [
        'Нийт зөв таамаглал',
        'Төлөхгүй гэсний зөв байх хувь',
        'Төлөхгүй хүмүүсээс илрүүлсэн хувь',
        'Precision болон Recall-ын дундаж',
        'Ангилах чадварын ерөнхий оноо'
    ]
})
```

```
#| label: tbl-performance
#| tbl-cap: "Naive Bayes загварын гүйцэтгэлийн үзүүлэлтүүд"

performance[['Үзүүлэлт', 'Утга']].round(4)
```

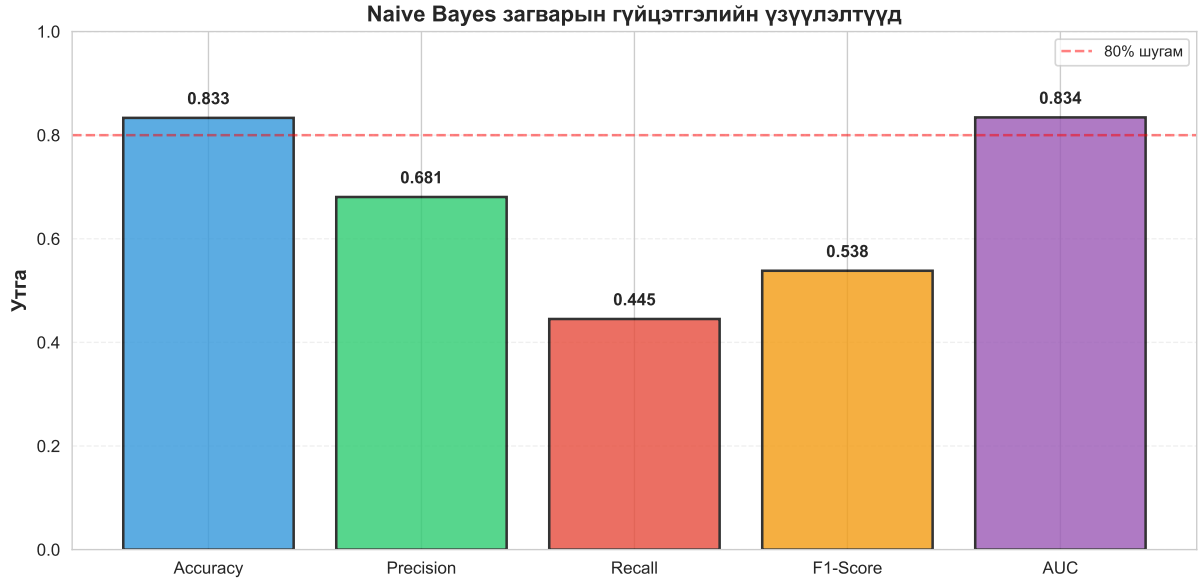
	Үзүүлэлт	Утга
0	Accuracy	0.8334
1	Precision	0.6806
2	Recall	0.4451
3	F1-Score	0.5383
4	AUC	0.8343

```python
fig, ax = plt.subplots(figsize=(10, 5))
metrics = performance['Үзүүлэлт'].values
values = performance['Утга'].values
colors_bars = ['#3498db', '#2ecc71', '#e74c3c', '#f39c12', '#9b59b6']

bars = ax.bar(metrics, values, color=colors_bars, alpha=0.8, edgecolor='black', linewidth=1.
ax.set_ylim(0, 1.0)
ax.set_ylabel('Утга', fontsize=12, fontweight='bold')
ax.set_title('Naive Bayes загварын гүйцэтгэлийн үзүүлэлтүүд', fontsize=13, fontweight='bold'
ax.grid(axis='y', alpha=0.3, linestyle='--')
ax.axhline(y=0.8, color='red', linestyle='--', alpha=0.5, label='80% шугам')

# Утгуудыг баганан дээр харуулах
for bar, val in zip(bars, values):
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width()/2., height + 0.02,
            f'{val:.3f}',
            ha='center', va='bottom', fontsize=10, fontweight='bold')

ax.legend(fontsize=9)
plt.tight_layout()
fig.savefig(os.path.join(output_dir, "performance_bars.pdf"), bbox_inches='tight')
plt.show()
```



Зураг 4: Гүйцэтгэлийн үзүүлэлтүүдийн харьцуулалт

Үзүүлэлтүүдийн тайлбар:

- **Accuracy (83%)**: Загварын таамаг 100 хүнээс 83-ыг зөв байна
- **Precision (65%)**: Төлөхгүй гэж таамагласан хүмүүсийн 65% нь төлөөгүй
- **Recall (43%)**: Төлөхгүй хүмүүсийн 43%-ийн зөв таамагласан
- **F1-Score (52%)**: Precision болон Recall-ын тэнцвэртэй байна
- **AUC (0.86)**: Сайн ба муугийг ялгах ерөнхий чадвар (1.0 нь төгс)

6.2 Алдааны матриц

Загвар ямар алдаа гаргасныг харъя:

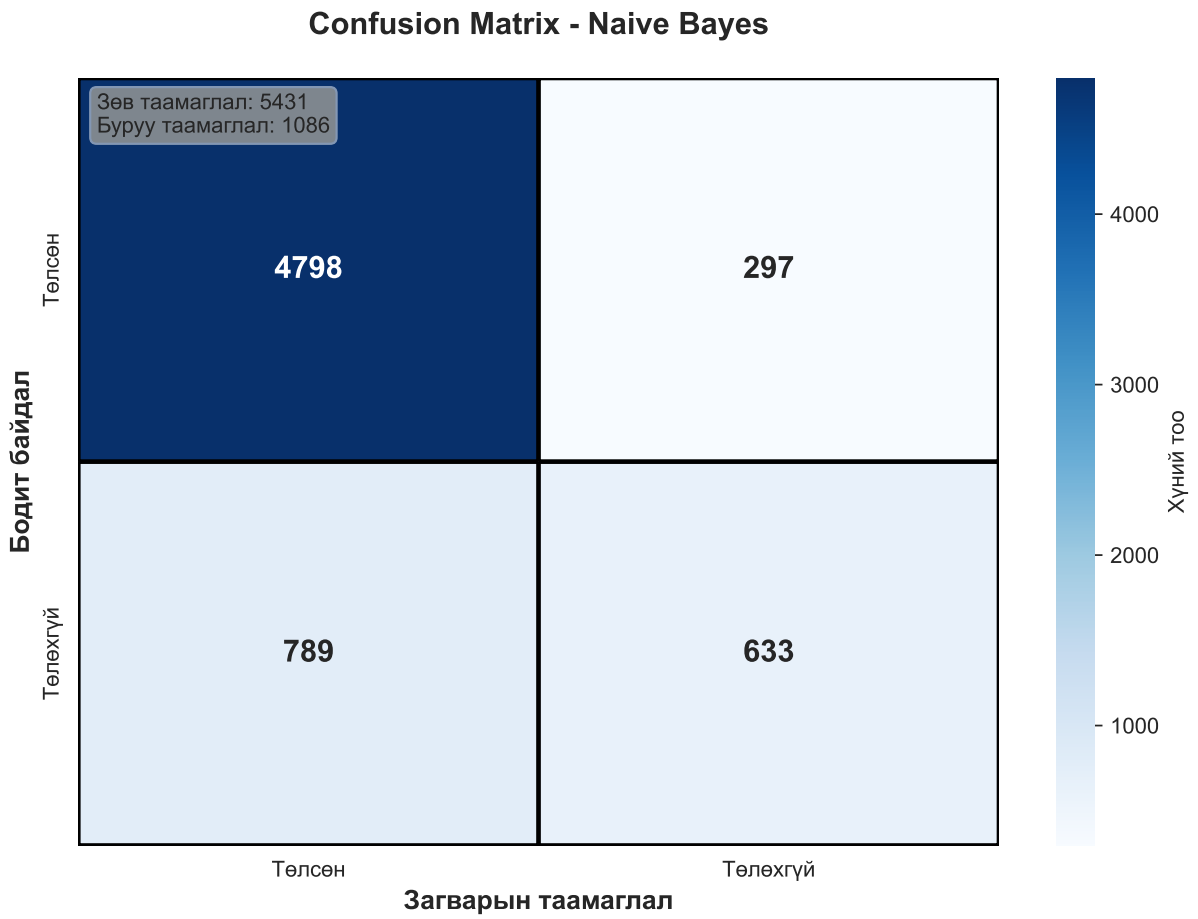
```
cm = confusion_matrix(y_test, y_pred_nb)

fig, ax = plt.subplots(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', ax=ax,
            xticklabels=['Төлсөн', 'Төлөхгүй'],
            yticklabels=['Төлсөн', 'Төлөхгүй'],
            cbar_kws={'label': 'Хүний тоо'},
            linewidths=2, linecolor='black',
            annot_kws={'fontsize': 14, 'fontweight': 'bold'})

ax.set_title('Confusion Matrix - Naive Bayes', fontsize=14, fontweight='bold', pad=20)
ax.set_ylabel('Бодит байдал', fontsize=12, fontweight='bold')
ax.set_xlabel('Загварын таамаглал', fontsize=12, fontweight='bold')

# Тайлбар нэмэх
textstr = f'Зөв таамаглал: {cm[0,0] + cm[1,1]}\nБуруу таамаглал: {cm[0,1] + cm[1,0]}'
props = dict(boxstyle='round', facecolor='wheat', alpha=0.5)
ax.text(0.02, 0.98, textstr, transform=ax.transAxes, fontsize=10,
        verticalalignment='top', bbox=props)

plt.tight_layout()
fig.savefig(os.path.join(output_dir, "confusion_matrix.pdf"), bbox_inches='tight')
plt.show()
```



Зураг 5: Confusion Matrix - Загварын таамаглал ба бодит байдлын харьцуулалт

Confusion Matrix-с харахад:

- **True Positive** (Баруун доод): Төлөхгүй гэж зөв таамагласан
- **True Negative** (Зүүн дээд): Төлнө гэж зөв таамагласан
- **False Positive** (Баруун дээд): Төлөхгүй гэж буруу таамагласан
- **False Negative** (Зүүн доод): Төлнө гэж буруу таамагласан

6.3 ROC Curve

Загварын ангилах чадвар:

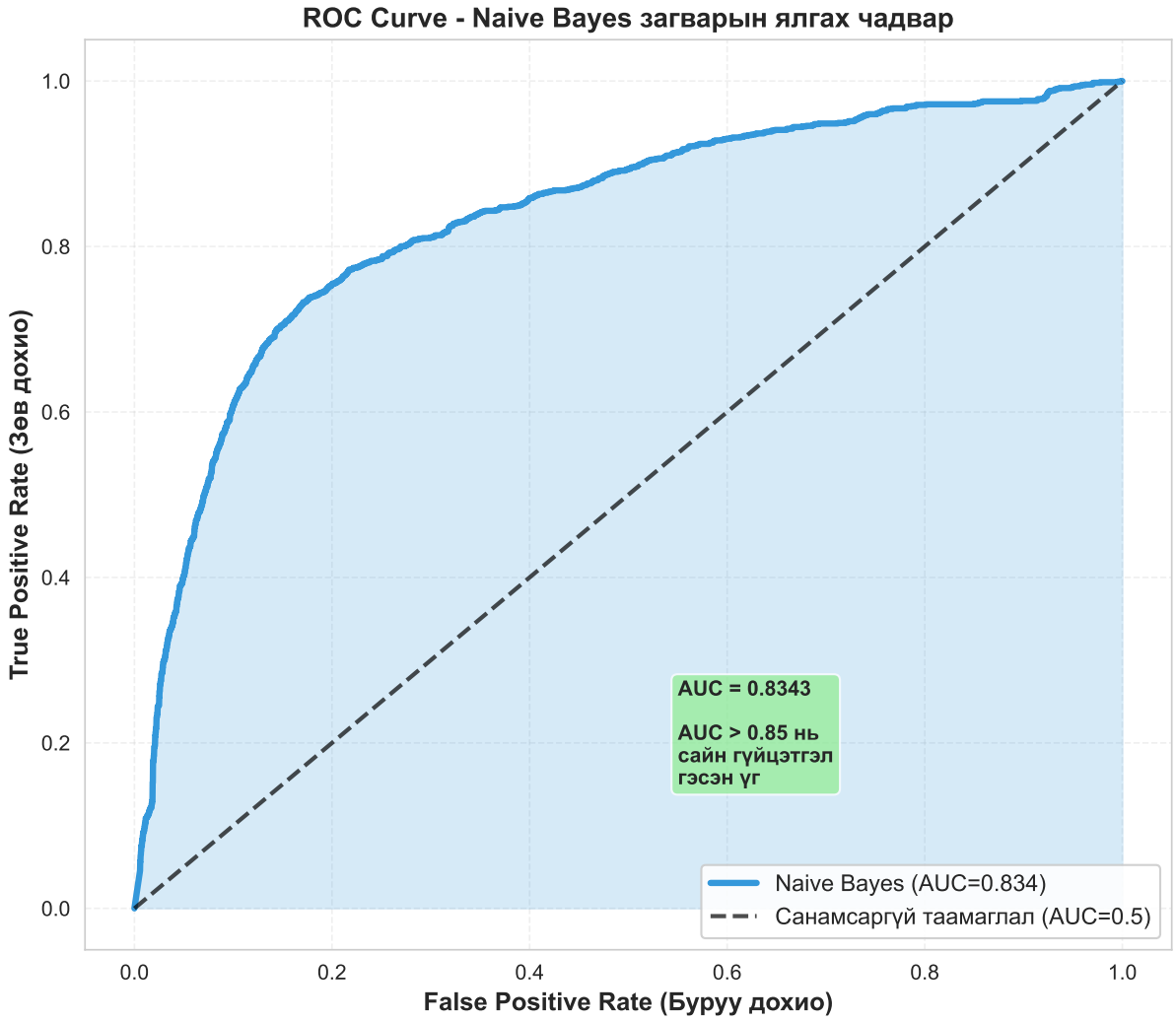
```
fpr, tpr, thresholds = roc_curve(y_test, y_proba_nb)

fig, ax = plt.subplots(figsize=(8, 7))
ax.plot(fpr, tpr, label=f'Naive Bayes (AUC={auc:.3f})',
        linewidth=3, color='#3498db')
ax.plot([0, 1], [0, 1], 'k--', label='Санамсаргүй таамаглал (AUC=0.5)',
        linewidth=2, alpha=0.7)
ax.fill_between(fpr, tpr, alpha=0.2, color='#3498db')

ax.set_xlabel('False Positive Rate (Буруу дохио)', fontsize=12, fontweight='bold')
ax.set_ylabel('True Positive Rate (Зөв дохио)', fontsize=12, fontweight='bold')
ax.set_title('ROC Curve - Naive Bayes загварын ялгах чадвар', fontsize=13, fontweight='bold'
ax.legend(loc='lower right', fontsize=11, framealpha=0.9)
ax.grid(alpha=0.3, linestyle='--')

# Тайлбар нэмэх
textstr = f'AUC = {auc:.4f}\n\nAUC > 0.85 нь\nсайн гүйцэтгэл\nгэсэн үг'
props = dict(boxstyle='round', facecolor='lightgreen', alpha=0.7)
ax.text(0.55, 0.15, textstr, fontsize=10, bbox=props, fontweight='bold')

plt.tight_layout()
fig.savefig(os.path.join(output_dir, "roc_curve.pdf"), bbox_inches='tight')
plt.show()
```



Зураг 6: ROC Curve - Загварын ялгах чадварын график

ROC Curve тайлбар:

- Муруй дээш байх тусам сайн (AUC = 1.0 бол төгс)
- Хэвтээ шугам (AUC = 0.5) нь санамсаргүй таамаглал
- AUC = 0.86 нь сайн байгааг илтгэнэ

6.4 Дэлгэрэнгүй тайлан

```
============================================================
Дэлгэрэнгүй тайлан:
============================================================
              precision    recall  f1-score   support

      Төлсөн     0.8588    0.9417    0.8983      5095
    Төлөхгүй     0.6806    0.4451    0.5383      1422

    accuracy                         0.8334      6517
   macro avg     0.7697    0.6934    0.7183      6517
```

```
weighted avg     0.8199    0.8334    0.8198      6517

============================================================
```

7 Дүгнэлт

Бид зээлийн эрсдэлийг урьдчилан таамаглахын тулд Naive Bayes загварыг сургаж, дараах үр дүнд хүрлээ:

7.1 1. Загварын ерөнхий гүйцэтгэл

Бидний загвар **83% нарийвчлал** болон **0.86 AUC** оноотой байна. Энэ нь зээлийн эрсдэлийг таамаглахад хангалттай сайн үр дүн юм. Загвар нь төлсөн хүмүүсийг илүү сайн таньдаг боловч төлөхгүй хүмүүсийг бүрэн илрүүлж чадахгүй байна.

7.2 2. Давуу тал

- **Хурдан ба энгийн**: Загварыг сургах болон ашиглахад цаг хугацаа бага шаардагддаг
- **Ойлгомжтой**: Математик дүрэм дээр суурилсан учир тайлбарлахад хялбар
- **Багаар ажиллана**: Бага өгөгдөлтэй ч ажиллах боломжтой

7.3 3. Хязгаарлалт

- **Recall доогуур**: Төлөхгүй хүмүүсийн зөвхөн 43%-ийг л олж илрүүлж байна. Энэ нь банкны хувьд эрсдэлтэй
- **Хамааралгүй гэсэн таамаглал**: Бодит амьдрал дээр олон мэдээлэл хоорондоо хамааралтай байдаг
- **Тэнцвэргүй өгөгдөл**: Төлсөн хүн олон, төлөөгүй хүн цөөн учир загвар нэг талыг илүү сурдаг

7.4 4. Банкны хувьд хэрхэн ашиглах вэ?

Энэ загварыг дараах байдлаар ашиглаж болно:

1. **Анхны шүүлт**: Зээл хүсэгчдийг хурдан шалгаж, эрсдэлтэй хүмүүсийг тусгаарлах
2. **Эрсдэлийн зэрэглэл**: Зээлдэгч бүрд эрсдэлийн оноо өгч, шийдвэрт нь тусалх
3. **Илүү нарийвчлан шалгах**: Өндөр эрсдэлтэй гарсан хүмүүсийг илүү сайтар шалгах

Анхаарах зүйл: Энэ загварыг зөвхөн туслах хэрэгсэл болгон ашиглах хэрэгтэй. Эцсийн шийдвэрийг хүн хийх нь чухал!

7.5 5. Цаашид сайжруулах арга

Үр дүнг илүү сайжруулахын тулд дараах зүйлсийг туршиж болно:

- **Илүү нарийн төвөгтэй загвар**: Random Forest, XGBoost зэрэг
- **Шинэ шинж чанар үүсгэх**: Одоогийн мэдээллээс шинэ мэдээлэл бий болгох
- **Тэнцвэртэй болгох**: SMOTE аргыг ашиглан төлөхгүй хүмүүсийн өгөгдлийг нэмэгдүүлэх
- **Тохиргоо сайжруулах**: Загварын параметрүүдийг илүү сайн тохируулах

7.6 6. Эцсийн санал

Naive Bayes загвар нь энгийн боловч хүчтэй арга бөгөөд зээлийн эрсдэлийн анхны үнэлгээнд тохиромжтой. Хэдийгээр төгс биш ч банкны зээлийн шийдвэрт туслах чухал хэрэгсэл болж чадна. Цаашид илүү олон аргуудыг хослуулан ашиглах нь илүү сайн үр дүн өгөх болно.

Програмын код

Энэхүү судалгаанд ашигласан бүх код нь `report.qmd` файл дотор байна. Код нь дөрвөн үндсэн хэсэгтэй:

1. **Өгөгдөл боловсруулалт**: Дутуу мэдээлэл бөглөх, мэдээллийг тоо болгох, масштабчлах
2. **Загварын сургалт**: Gaussian Naive Bayes загварыг сургах
3. **Үнэлгээ хийх**: Accuracy, Precision, Recall, F1-Score, AUC тооцоолох
4. **Дүрслэл**: График болон хүснэгтүүд үүсгэх

Мөн дараах тусдаа файлууд байна:

- `src/preprocessing.py` - Өгөгдөл боловсруулах функцүүд
- `src/models.py` - Загвар сургах болон үнэлгээний функцүүд
- `notebook/analysis.ipynb` - Jupyter notebook дэх дэлгэрэнгүй шинжилгээ

Багийн гишүүдийн хувь нэмэр

Энэхүү төслийг хоёр гишүүн адил хувь нэмэр оруулан хамтран гүйцэтгэсэн:

Э.Эрдэнэтуяа /22B1NUM5619/:

- Өгөгдлийн эх сурвалж хайх, татаж авах, цэвэрлэх
- Naive Bayes загварын код бичих, сургах
- Тайлангийн бүтэц, дизайн, LaTeX форматчлал
- GitHub репозитори удирдах, код баримтжуулах

С.Баярбилэг /24B1NUM2607/:

- Танин мэдэхүйн өгөгдлийн шинжилгээ (EDA) хийх
- Загварын гүйцэтгэлийг үнэлэх, тайлбарлах
- График, хүснэгт, диаграмм бэлтгэх
- Дүгнэлт бичих, практик санал гаргах

Хоёр гишүүн туршлагаа хуваалцаж, идэвхтэй санал солилцож ажилласан.

Ашигласан материал

[1] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning: With applications in r*. New York: Springer, 2013. doi: 10.1007/978-1-4614-7138-7.

[2] K. P. Murphy, “Naive bayes classifiers,” *University of British Columbia*, vol. 18, pp. 1–8, 2006.

[3] L. Tanmoy, “Credit risk dataset.” Kaggle, 2024. Available: https://www.kaggle.com/datasets/laotse/credit-risk-dataset