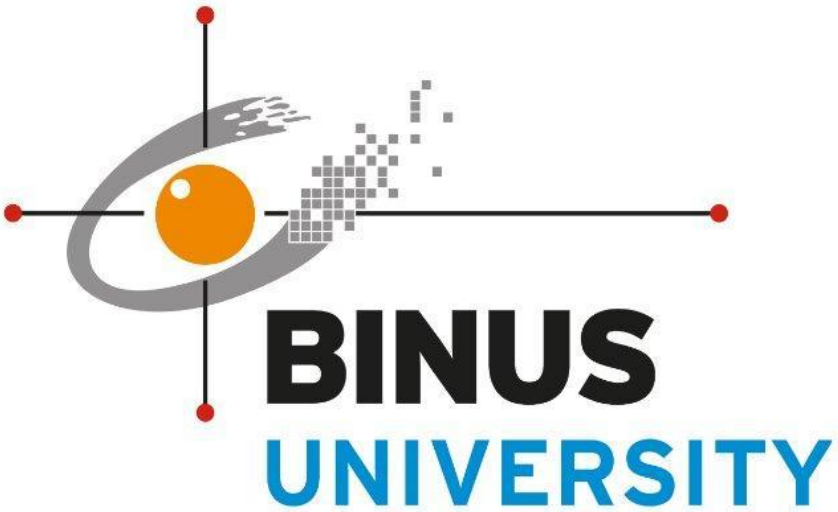
Dokumentasi AOL Machine Learning



Anggota kelompok:
Agnes Calista - 2501980690
Olivia Intan Permata Dewi - 2540128650
Snow White Putri Safa - 2502020970

Jl. Kebon Jeruk Raya No. 27, Kebon Jeruk
Jakarta Barat 11530, Indonesia
2022

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era ini, teknologi telah berkembang dengan pesat. Hampir semua masyarakat bergantung pada teknologi dalam kehidupan sehari-harinya. Teknologi memberikan kemudahan bagi penggunanya untuk mengakses informasi dan berkomunikasi. Banyak dampak positif yang bisa diberikan teknologi kepada penggunanya. Tetapi, selain dampak positif, teknologi juga memberikan dampak negatif, mudahnya pengguna untuk mengakses informasi membuat pengguna dapat dengan mudah menyebarkan informasi palsu (hoax). Pada mesin pencarian, dapat kita temui informasi mengenai syarat-syarat penting agar pengajuan pinjaman kepada bank dapat disetujui. Akan tetapi, tidak semua informasi bersifat akurat, karena informasi yang diberikan hanya sebatas berdasarkan pengalaman. Pada proyek ini, kami memberikan analisis untuk mengetahui variabel-variabel yang penting untuk persetujuan pinjaman berdasarkan data sehingga hasil analisis yang kita miliki dapat memberikan akurasi lebih dan bisa dipercaya.

Perusahaan ingin mengotomatiskan proses kelayakan pinjaman (real time) berdasarkan detail pelanggan yang diberikan saat mengisi formulir pada aplikasi online. Dengan diadakannya pinjaman otomatis ini akan lebih mempermudah perusahaan dalam menganalisa tanpa membutuhkan waktu yang lama, selain itu untuk proses kemudahan dengan cepat. Agar proses ini dapat bersifat otomatis, mereka telah memberikan masalah ini untuk mengidentifikasi segmen pelanggan, yang berhak atas jumlah pinjaman sehingga mereka secara khusus dapat menargetkan pelanggan.

1.2 Rumusan Masalah

1. proses peminjaman membutuhkan waktu yang lama untuk memprosesnya
2. Faktor apa saja yang mempengaruhi bank untuk memberikan pinjaman kepada nasabah?
3. sulitnya untuk melakukan pinjaman ke bank
4. Metode yang paling cocok untuk klasifikasi “Loan-Prediction”

1.3 Tujuan Penelitian

1. Mengetahui faktor apa saja yang mempengaruhi bank untuk memberikan pinjaman uang kepada nasabah.
2. Memprediksi kelas atau kategori dari data baru berdasarkan karakteristik data yang ada.
3. Mengukur akurasi dengan beberapa algoritma, seperti K-Nearest Neighbors, Logistic Regression, Decision Tree, Naive Bayes, dan Random Forest.

BAB 2
LANDASAN TEORI

1. Algoritma Machine Learning

Machine Learning merupakan mesin yang dikembangkan agar dapat belajar sendiri tanpa arahan. Algoritma machine learning didasari dengan ilmu matematika, statistika, dan lainnya. Selain itu Machine learning itu sendiri adalah salah satu cabang dari artificial intelligence.

Pada tugas ini kami menggunakan beberapa algoritma Machine Learning untuk mencari model terbaik dengan akurasi yang tepat. berikut beberapa algoritma yang kami pakai untuk modeling dari dataset Loan Prediction tersebut.

a. K-Nearest Neighbors

   Algoritma Nearest Neighbor merupakan algoritma klasifikasi paling sederhana, metode ini cukup mudah dipahami karena mengklasifikasikannya berdasarkan jarak terdekat dengan objek lain.

b. Logistic Regression

   Logistic Regression adalah teknik menganalisa menggunakan ilmu matematika agar dapat menemukan hubungan antara kedua faktor data. selain itu, untuk memprediksi nilai dari salah satu faktor tersebut berdasarkan faktor lainnya.

c. Decision Tree

   Decision tree adalah salah satu cara untuk memperoleh masa depan dengan cara membangun regresi ataupun klasifikasi modeling dalam struktur pohon.

d. Naive Bayes

   Naive bayes adalah metode klasifikasi yang paling efektif dan efisien untuk machine learning dan data mining. Dalam perhitungannya, naive bayes menggunakan probabilitas dan statistik, sebagaimana dikemukakan oleh Inggris Thomas Bayes.

e. Random Forest

   Random forest adalah metode klasifikasi yang berisi kumpulan dari pohon keputusan atau decision tree yang dijadikan ke dalam satu model. Random forest digunakan untuk kasus klasifikasi dengan dataset dalam jumlah besar.

BAB III
METODOLOGI PENELITIAN

1. *Exploratory Data Analysis*
    - .shape() : mencari dimensi
    - .info() : menampilkan informasi tentang dataset
    - .describe() : menampilkan informasi deskriptif statistik
    - .duplicated().sum() : menampilkan jumlah data yang terduplikasi
    - .nunique() : menampilkan jumlah data yang bernilai unik.
    - .drop() : menghapus column yang tidak terpakai
    - .isna().sum() : menampilkan jumlah data yang memiliki nilai null.

2. *Feature Engineering*

    2.1. *Imputation*

    Mengganti missing value dengan nilai modus (nilai dengan frekuensi terbanyak)

    2.2. *Label Encoding*

    mengubah kategorik menjadi numerik sehingga dapat dibaca oleh mesin.

    2.3. *Find Value Count*

    melihat jumlah dari masing-masing variabel

    2.4 Creating Feature

    2.5 *Find Outliers*

    mengecek apakah terdapat outliers atau tidak dan apakah outliers tersebut harus dihilangkan atau tidak.

    2.6 *Check Label*

    membuat klasifikasi menggunakan variabel target.

    2.7 *Scaling*

    membuat data numerik pada dataset memiliki rentang nilai yang sama.

3. *Train Machine Learning Models & Evaluate Machine Learning Models*

    3.1 K-Nearest Neighbors

    3.2 Logistic Regression

    3.3 Decision Tree

    3.4 Naive Bayes

    3.5 Random Forest

BAB IV
PEMBAHASAN

1. K-nearest Neighbor

```
[ ] #from sklearn.neighbors import KNeighborsClassifier
    knn = KNeighborsClassifier()

    knn.fit(x_train, y_train)
```

```
KNeighborsClassifier()
```

```
[ ] #Prediction
    prediksi_knn = knn.predict(x_test)
    prediksi_knn
```

```
array([1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1])
```

```
[ ] # Print the accuracy
    print(knn.score(x_test, y_test))
```

```
0.6178861788617886
```

```
[ ] print(classification_report(y_test, prediksi_knn))
```

	precision	recall	f1-score	support
0	0.26	0.13	0.18	38
1	0.68	0.84	0.75	85
accuracy			0.62	123
macro avg	0.47	0.48	0.46	123
weighted avg	0.55	0.62	0.57	123

Pada Algoritma K-Nearest neighbor ini didapat akurasi yang belum cukup baik yaitu di 62%, karena itu kami mencoba menggunakan algoritma lain.

2. Logistic Regression

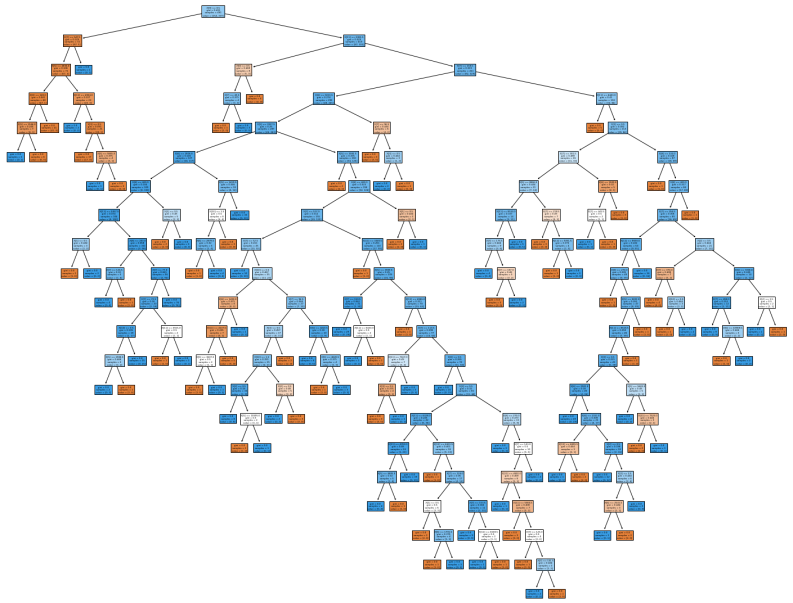
```
[ ] #from sklearn.linear_model import LogisticRegression
    lr = LogisticRegression()
    lr.fit(x_train, y_train)
    lr_pred = lr.predict(x_test)
    lrR = classification_report(lr_pred, y_test)

    print(lrR)
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	0
1	1.00	0.69	0.82	123
accuracy			0.69	123
macro avg	0.50	0.35	0.41	123
weighted avg	1.00	0.69	0.82	123

Pada algoritma Logistic Regression didapatkan yang lebih daripada sebelumnya yaitu sebesar 69%.

3. Decision Tree

Decision tree adalah salah satu cara untuk memperoleh masa depan dengan cara membangun regresi ataupun klasifikasi modeling dalam struktur pohon.



```
print(classification_report(y_test, prediksi_dt))
```

	precision	recall	f1-score	support
0	0.62	0.68	0.65	38
1	0.85	0.81	0.83	85
accuracy			0.77	123
macro avg	0.74	0.75	0.74	123
weighted avg	0.78	0.77	0.78	123

Hasil yang kami dapatkan dari decision tree ini akurasinya sebesar 77%

4. Naive Bayes

Naive bayes adalah metode klasifikasi yang paling efektif dan efisien untuk machine learning dan data mining. Dalam perhitungannya, naive bayes menggunakan probabilitas dan statistik, sebagaimana dikemukakan oleh Inggris Thomas Bayes.

```
[ ] #from sklearn.naive_bayes import GaussianNB
    gnb = GaussianNB()
    gnb.fit(x_train,y_train)

    GaussianNB()

[ ] #Predict
    prediksi_nb = gnb.predict(x_test)
    print(prediksi_nb)

    [1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 0
     1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
     1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1
     0 1 1 1 1 0 1 1 0 1 1 1]
```

```
print(classification_report(y_test, prediksi_nb))

              precision    recall  f1-score   support

           0       0.89      0.42      0.57        38
           1       0.79      0.98      0.87        85

    accuracy                           0.80       123
   macro avg       0.84      0.70      0.72       123
weighted avg       0.82      0.80      0.78       123
```

Didapatkan model menggunakan Naive-Bayes dengan akurasi sebesar 80%.

5. Random Forest

Random forest adalah metode klasifikasi yang berisi kumpulan dari pohon keputusan atau decision tree yang dijadikan ke dalam satu model. Random forest digunakan untuk kasus klasifikasi dengan dataset dalam jumlah besar.

Dari algoritma yang telah kami buat, model Random Forest memberikan akurasi paling tinggi, sehingga algoritma yang kami pakai adalah Random Forest.

```
[ ] #from sklearn.ensemble import RandomForestRegressor
    rf = RandomForestRegressor(n_estimators = 100, random_state = 42)
    rf.fit(x_train, y_train)

    RandomForestRegressor(random_state=42)
```

Diatas ini merupakan data training yang dimodelkan dan ditetapkan random statenya sebesar 42.

```
[ ] #Predict
    prediksi_rf = rf.predict(x_test)
    print(prediksi_rf)

    [0.97 0.71 0.25 0.88 0.92 0.95 0.44 0.8  0.93 0.68 0.83 0.96 0.99 0.67
     0.79 0.88 0.62 0.89 0.92 0.38 0.78 0.9  0.86 0.28 0.62 0.84 0.13 0.51
     0.03 0.58 0.66 0.9  0.72 0.07 0.98 0.69 0.03 0.88 0.79 0.51 0.99 0.7
     0.55 0.83 0.95 0.74 0.8  0.53 0.89 1.   0.85 0.68 0.77 0.15 0.86 0.61
     0.   0.95 0.77 0.84 0.9  0.8  0.97 0.46 0.89 0.97 0.31 0.95 0.04 0.9
     0.82 1.   0.7  0.99 0.94 0.78 0.81 0.81 0.07 0.44 0.95 1.   0.73 0.99
     0.62 0.82 0.94 0.88 0.65 0.81 0.95 0.08 0.8  0.71 0.93 0.04 0.86 0.96
     0.84 0.73 0.17 0.01 0.99 0.37 0.85 0.49 0.99 0.49 0.45 0.95 0.89 0.2
     0.41 0.47 0.94 0.37 0.   0.36 0.7  0.   0.4  0.69 0.83]
```

```
[ ] yp = (prediksi_rf >= 0.5).astype(int)
    yp

    array([1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
           1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1,
           0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1,
           1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1])
```

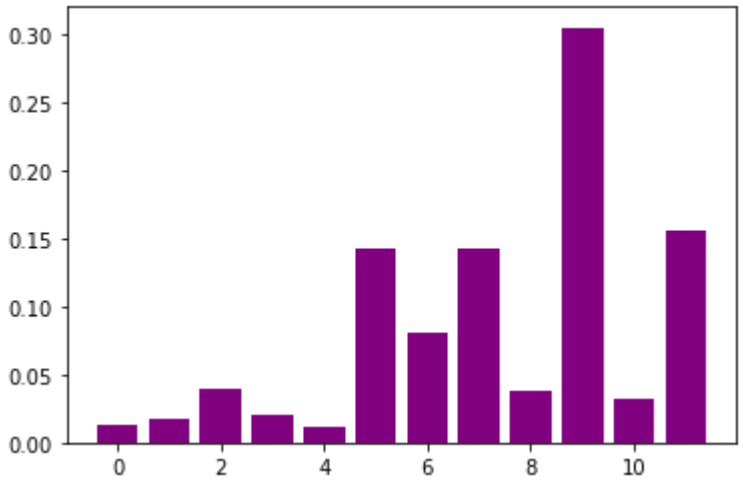
```python
#Random forest for feature importance on a regression problem

importance = rf.feature_importances_
for i,v in enumerate(importance):
    print('Feature: %0d, Score: %.5f' % (i,v))

plt.bar([x for x in range(len(importance))], importance, color = 'purple')
plt.show()
```

```
Feature: 0, Score: 0.01260
Feature: 1, Score: 0.01784
Feature: 2, Score: 0.03980
Feature: 3, Score: 0.01975
Feature: 4, Score: 0.01212
Feature: 5, Score: 0.14314
Feature: 6, Score: 0.08067
Feature: 7, Score: 0.14262
Feature: 8, Score: 0.03767
Feature: 9, Score: 0.30512
Feature: 10, Score: 0.03245
Feature: 11, Score: 0.15622
```

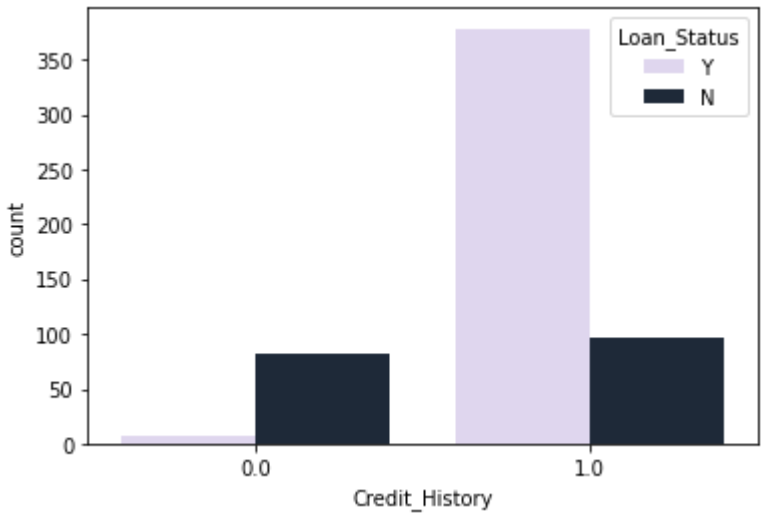
```
[ ] print(classification_report(y_test, yp))
```

	precision	recall	f1-score	support
0	0.74	0.61	0.67	38
1	0.84	0.91	0.87	85
accuracy			0.81	123
macro avg	0.79	0.76	0.77	123
weighted avg	0.81	0.81	0.81	123

BAB V
KESIMPULAN



Pada grafik diatas, dapat dilihat variable ke-9 memiliki nilai yang paling besar, sekitar 30% dari total semuanya. dapat disimpulkan bahwa credit history memiliki hubungan dengan variabel loan_status lebih besar dari variabel lainnya.



Dari semua model yang telah dibuat, kami mendapatkan random forest sebagai model classifier terbaik dalam dataset loan prediction. serta fitur yang paling mempengaruhi loan_status adalah Credit history, dengan korelasi sekitar 30%.

	Method	Accuracy	F1	Precision	Recall
0	KNN	0.617886	0.751323	0.682692	0.835294
1	Logistic Regression	0.691057	0.817308	0.691057	1.000000
2	Decision Tree	0.756098	0.819277	0.839506	0.800000
3	Gaussian Naive Bayes	0.804878	0.873684	0.790476	0.976471
4	Random Forest	0.813008	0.870056	0.836957	0.905882

dari tabel diatas, kita bisa melihat model klasifikasi terbaik yaitu random forest dengan akurasi sebesar 81,3%.

DAFTAR PUSTAKA

Farokhah. (2020, November 25). Implementasi k-nearest neighbor untuk Klasifikasi Bunga Dengan Ekstraksi Fitur Warna RGB | Farokhah | Jurnal Teknologi Informasi Dan Ilmu Komputer. Jurnal Teknologi Informasi dan Ilmu Komputer. https://jtiik.ub.ac.id/index.php/jtiik/article/view/2608

https://aws.amazon.com/id/what-is/logistic-regression/
https://media.neliti.com/media/publications/283828-metode-naive-bayes-untuk-prediksi-kelulu-139fcfea.pdf
https://repository.uinjkt.ac.id/dspace/bitstream/123456789/55034/1/LAILI%20FADILAH-FST.pdf
https://medium.com/machine-learning-id/melakukan-feature-scaling-pada-dataset-229531bb08de