DOI: 10.22146/jnteti.Nomor DOI

Pengembangan Model Regresi Prediktif untuk Harga Rumah Menggunakan Teknik Ensemble Stacking

Angelina Nadeak[1], Jeremy Samosir [2], Ade Siahaan [3], Rosari Simanjuntak [4]

Github : JeremySamosir/DAMI-Kelompok-10-House-Pricing

INTISARI — Penelitian ini bertujuan untuk mengembangkan model prediksi harga rumah dengan menggunakan teknik ensemble stacking, yang menggabungkan tiga algoritma pembelajaran mesin populer, yaitu XGBoost, LightGBM, dan CatBoost. Dataset yang digunakan berasal dari kompetisi "House Prices - Advanced Regression Techniques" di platform Kaggle, yang mencakup berbagai fitur terkait properti seperti ukuran rumah, jumlah kamar, kondisi bangunan, dan lokasi. Dalam penelitian ini, pendekatan yang digunakan adalah CRISP-DM, yang terdiri dari langkah-langkah pemahaman bisnis, eksplorasi data, persiapan data, pembangunan model, dan evaluasi model. Evaluasi model dilakukan dengan menggunakan metrik RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), dan R^2 (R-squared). Hasil evaluasi menunjukkan bahwa model stacking yang dikembangkan berhasil mencapai nilai R^2 sebesar 0,9001 pada data pelatihan dan 0,8367 pada data validasi, yang menunjukkan performa yang sangat baik. Model ini diharapkan dapat memberikan solusi yang lebih tepat dan berbasis data dalam menentukan harga rumah, serta membantu pemangku kepentingan di sektor properti, seperti investor, agen real estat, dan pembeli rumah, dalam membuat keputusan yang lebih akurat dan efisien.

KATA KUNCI — Prediksi Harga Rumah, Pembelajaran Mesin, Ensemble Stacking, XGBoost, LightGBM, CatBoost, Regresi, Evaluasi Model.

I. PENDAHULUAN

1.1 Latar Belakang

Industri properti merupakan sektor yang sangat dinamis dan memiliki dampak yang signifikan terhadap perekonomian suatu negara. Salah satu aspek yang paling penting dalam industri ini adalah penentuan harga rumah, yang menjadi indikator utama dalam transaksi jual-beli properti. Penentuan harga rumah sering kali tidak hanya dipengaruhi oleh faktor internal properti seperti ukuran dan kualitas bangunan, tetapi juga oleh faktor eksternal yang sulit diprediksi, seperti kondisi pasar, lokasi, dan tren ekonomi yang berlaku. Hal ini sering kali membuat proses penentuan harga rumah menjadi subjektif dan rentan terhadap kesalahan estimasi.

Dengan kemajuan teknologi, khususnya dalam bidang kecerdasan buatan (AI) dan pembelajaran mesin (machine learning), kini terdapat peluang untuk mengurangi ketidakpastian dalam proses penentuan harga rumah. Pendekatan berbasis data ini memungkinkan analisis harga rumah yang lebih objektif, berdasarkan pada berbagai faktor yang relevan dan dapat diprediksi secara lebih akurat. Salah satu pendekatan yang menjanjikan dalam prediksi harga rumah adalah penggunaan model prediksi berbasis pembelajaran mesin (machine learning), terutama dengan teknik ensemble stacking yang menggabungkan beberapa model untuk meningkatkan akurasi prediksi.

Penelitian ini bertujuan untuk mengembangkan sistem prediksi harga rumah berbasis machine learning menggunakan teknik ensemble stacking yang mengkombinasikan algoritma XGBoost, LightGBM, dan CatBoost. Teknik ensemble ini diharapkan dapat menghasilkan model yang lebih akurat dengan meminimalisir kesalahan yang biasanya terjadi pada model-model individual. Dengan model ini, diharapkan para pemangku kepentingan di sektor properti, seperti investor, penjual, pembeli, dan agen real estat, dapat membuat keputusan yang lebih cepat dan tepat dalam transaksi properti.

1.2 Manfaat

Manfaat dari pengerjaan proyek ini yaitu:

1. Dengan menggunakan model prediksi berbasis machine learning, penelitian ini bertujuan untuk mengurangi ketidakpastian dalam penentuan harga rumah. Ini membantu mengurangi kesalahan yang sering terjadi akibat penilaian harga yang subjektif, sehingga harga yang diberikan lebih akurat.
2. Siste m prediksi harga ini memberikan informasi yang lebih jelas dan berbasis data bagi agen properti, pembeli, dan penjual. Dengan data yang lebih objektif, proses negosiasi harga rumah menjadi lebih transparan dan efisien.
3. Penelitian ini menghasilkan model prediksi harga rumah yang stabil dan dapat diandalkan, menggunakan metode machine learning seperti XGBoost, LightGBM, dan CatBoost. Model ini akan memberikan prediksi yang lebih tepat, terutama saat data yang digunakan cukup kompleks.

1.3 Ruang Lingkup

Ruang lingkup dalam proyek yang dikerjakan yaitu:

1. Penelitian ini menggunakan dataset yang diambil dari kompetisi "House Prices - Advanced Regression Techniques" di platform Kaggle. Dataset ini mencakup berbagai fitur terkait properti, seperti ukuran rumah, jumlah kamar, kondisi bangunan, lokasi, dan fasilitas tambahan lainnya.
2. Model yang dikembangkan akan dievaluasi menggunakan beberapa metrik evaluasi, termasuk RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), R^2 (Coefficient of Determination), dan MAPE (Mean Absolute Percentage Error). Evaluasi ini bertujuan untuk mengukur kinerja model dalam memprediksi harga rumah secara akurat dan konsisten.

1.4 Istilah dan Singkatan

Singkatan	Istilah
AI	Artificial Intelligence (Kecerdasan Buatan)

p-ISSN 2301–4156 | e-ISSN 2460–5719

XGBoost	Extreme Gradient Boosting
LightGBM	Light Gradient Boosting Machine
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error

II. STUDI LITERATURE

2.1 Prediksi Harga Rumah Menggunakan Teknik Pembelajaran Mesin

Prediksi harga rumah merupakan salah satu aplikasi penting dari pembelajaran mesin, yang melibatkan analisis data properti untuk memperkirakan harga jual rumah berdasarkan berbagai faktor seperti lokasi, ukuran, kualitas bangunan, dan kondisi pasar. Berbagai algoritma regresi telah digunakan dalam tugas ini, termasuk regresi linier, regresi pohon keputusan, dan lebih baru lagi, teknik berbasis ensemble (misalnya, Random Forest, XGBoost, dan Gradient Boosting). Penelitian-penelitian sebelumnya telah menunjukkan bahwa model berbasis pembelajaran mesin cenderung lebih akurat daripada pendekatan tradisional yang bergantung pada penilaian manusia [1].

2.2 Regresi dalam Prediksi Harga Rumah

Regresi adalah metode yang paling umum digunakan dalam prediksi harga rumah. Regresi linier telah menjadi pendekatan dasar, tetapi sering kali tidak dapat menangkap hubungan non-linear yang ada dalam data harga rumah. Oleh karena itu, model regresi yang lebih kompleks, seperti regresi berbasis pohon (misalnya, Random Forest), dan teknik ensemble digunakan untuk meningkatkan akurasi prediksi. Penelitian oleh Breiman [2] mengenai Random Forest menunjukkan bahwa model ini sangat baik dalam menangani data dengan banyak fitur dan variabel yang saling berinteraksi.

2.3 Ensemble Stacking dalam Pembelajaran Mesin

Ensemble stacking adalah teknik yang menggabungkan prediksi dari beberapa model untuk menghasilkan prediksi yang lebih kuat. Teknik ini menggunakan model-level pertama (base learners), yang kemudian dikombinasikan oleh model-level kedua (meta-learner) untuk memberikan hasil akhir. Teknik ini sangat efektif dalam meminimalkan bias dan varians yang ada dalam model tunggal. Hasil penelitian oleh Wolpert [3] menunjukkan bahwa ensemble methods, termasuk stacking, sering kali menghasilkan performa yang lebih baik dibandingkan dengan model tunggal.

2.4 Mesin XGBoost, LightGBM, dan CatBoost dalam Prediksi Harga Rumah

XGBoost, LightGBM, dan CatBoost adalah algoritma yang populer dalam teknik ensemble, terutama dalam menangani masalah regresi dengan data besar dan kompleks. XGBoost adalah algoritma berbasis gradient boosting yang telah terbukti efektif dalam berbagai kompetisi machine learning, termasuk prediksi harga rumah [2]. LightGBM, yang dikembangkan oleh Microsoft, merupakan implementasi gradient boosting yang lebih efisien dan lebih cepat untuk dataset besar. CatBoost, yang dikembangkan oleh Yandex, dirancang khusus untuk menangani fitur kategorikal tanpa preprocessing yang rumit. Semua algoritma ini memiliki keunggulan dalam meningkatkan akurasi prediksi dan menangani masalah multikolinearitas yang sering kali muncul dalam dataset harga rumah.

2.5 Evaluasi Model dalam Prediksi Harga Rumah

Evaluasi model prediksi harga rumah biasanya dilakukan menggunakan berbagai metrik seperti RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), dan R^2 (Coefficient of Determination). Metrik ini digunakan untuk mengukur akurasi dan kestabilan model dalam memprediksi harga rumah yang sebenarnya. Penelitian oleh Zhang et al. [5] menunjukkan bahwa model ensemble seperti XGBoost dan LightGBM menghasilkan RMSE yang lebih rendah dan R^2 yang lebih tinggi dibandingkan dengan model prediksi harga rumah tradisional.

III. METODE PENELITIAN

3.1 Desain Penelitian

Penelitian ini bertujuan untuk mengembangkan model prediksi harga rumah menggunakan teknik ensemble stacking yang menggabungkan beberapa algoritma machine learning untuk meningkatkan akurasi prediksi. Desain penelitian ini terdiri dari beberapa tahap yang meliputi pengumpulan data, pra-pemrosesan data, pengembangan model, evaluasi model, dan penerapan model dalam bentuk aplikasi prediktif.

Penelitian ini akan menggunakan dataset yang diambil dari kompetisi *House Prices - Advanced Regression Techniques* di *platform* Kaggle. Dataset ini sangat kaya dan mencakup berbagai fitur yang relevan dengan harga rumah, termasuk ukuran rumah, jumlah kamar tidur, kondisi bangunan, kualitas rumah, dan lokasi properti.

3.1.1 Business Understanding

Tujuan utama dari analisis data ini adalah untuk membantu dalam memprediksi harga rumah melalui sistem berbasis machine learning yang dapat mengurangi ketidakpastian dalam penentuan harga. Melalui sistem prediksi ini, diharapkan stakeholder properti seperti investor, penjual dan pembeli dapat mengambil keputusan yang lebih cepat dan tepat. Berikut adalah beberapa manfaat yang dapat dicapai dalam jangka panjang:

- Mengurangi kesalahan estimasi harga: Memastikan harga yang lebih akurat dan dapat diandalkan.
- Memberikan insight berbasis data: Membantu dalam proses negosiasi harga jual atau beli rumah.
- Meningkatkan efisiensi waktu dan biaya: Meminimalkan waktu yang dibutuhkan untuk menentukan harga jual/beli.
- Meningkatkan daya saing perusahaan properti: Menggunakan teknologi AI untuk memperoleh keunggulan kompetitif.

3.1.2 Data Understanding

Pada tahap *Data Understanding*, kami mengidentifikasi karakteristik, pola, dan kualitas data yang akan digunakan dalam proyek ini. Tanpa pemahaman yang baik tentang data, proses *data mining* bisa menghasilkan interpretasi yang salah dan keputusan yang tidak tepat. Tahap ini terdiri dari beberapa sub-tugas sebagai berikut:

a. pengumpulan Data

Dataset yang digunakan untuk analisis ini berasal dari kompetisi *House Prices - Advanced Regression Techniques* yang ada di *platform* Kaggle. Dataset ini terdiri dari 1.460 data pelatihan dan 1.459 data pengujian dengan total 81 fitur yang mencakup karakteristik rumah, kondisi bangunan, kualitas rumah, dan lokasi properti.

b. Analisis Data

Pada tahap ini, dilakukan eksplorasi awal untuk memahami distribusi data, pola umum, serta variabilitas pada fitur-fitur seperti ukuran rumah, kondisi bangunan, dan lokasi properti. Visualisasi distribusi harga rumah *(SalePrice)* dilakukan untuk memahami pola sebaran harga rumah.

c. Validasi Data

Untuk memastikan bahwa data yang dikumpulkan valid dan berkualitas, dilakukan validasi dengan memeriksa adanya data yang hilang, inkonsistensi nilai, dan *outlier*. Hal ini dilakukan untuk memastikan bahwa data yang digunakan dalam model adalah data yang bersih dan dapat dipercaya.

3.1.2.1 Data Overview

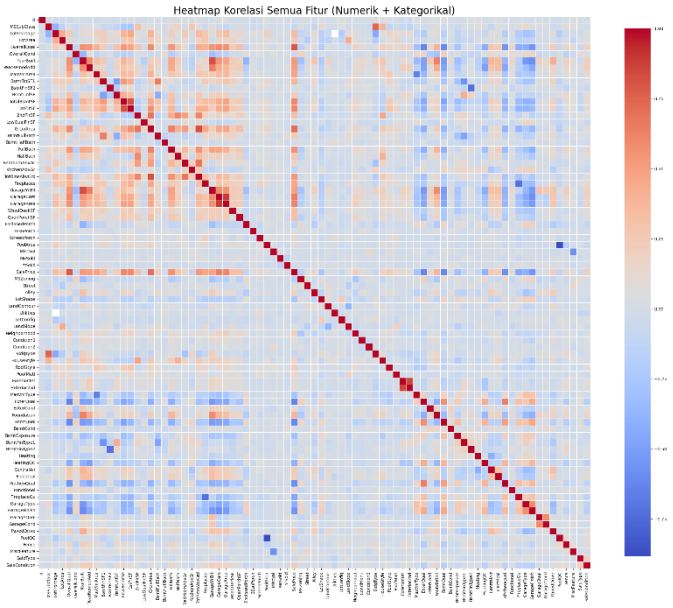
Dataset ini mencakup berbagai aspek rumah, seperti nomor ID, kualitas rumah *(OverallQual)*, luas tanah *(LotArea)*, luas bangunan *(GrLivArea)*, dan harga rumah *(SalePrice*). Visualisasi awal dari dataset memberikan gambaran tentang struktur dan distribusi data yang akan digunakan dalam model prediksi harga rumah.

Dataset *Overview:*

- Jumlah kolom: 81 kolom, terdiri dari fitur numerik, teks, dan kategorikal.
- Jumlah baris: 1.460 baris untuk data pelatihan dan 1.459 baris untuk data pengujian.
- Data lengkap: Tidak ada nilai yang hilang (*missing values)*.
- Data unik: Tidak ada data yang terduplikasi.

3.1.2.2 Correlations

Korelasi adalah ukuran statistik yang menggambarkan sejauh mana dua variabel memiliki hubungan yang saling linier. Dalam konteks ini, *heatmap* pada Gambar menunjukkan hubungan korelasi antar variabel dalam dataset harga rumah. Skala warna yang digunakan pada heatmap ini untuk merepresentasikan kekuatan hubungan antar fitur, di mana warna merah gelap menunjukkan korelasi positif yang kuat, warna putih menunjukkan tidak adanya korelasi, dan warna biru gelap menunjukkan korelasi negatif yang kuat.



Heatmap korelasi pada gambar menunjukkan hubungan antar fitur dalam dataset harga rumah. Berikut adalah beberapa temuan penting:

a. Korelasi Positif yang Kuat:
Dari heatmap, dapat dilihat bahwa beberapa fitur menunjukkan korelasi positif yang sangat kuat dengan *SalePrice* (Harga Rumah). Fitur *OverallQual* (Kualitas Rumah) memiliki korelasi yang sangat tinggi dengan harga rumah, yaitu 0.79. Hal ini menunjukkan bahwa semakin baik kualitas rumah, semakin tinggi harga jualnya. Begitu juga dengan *GrLivArea* (Luas Bangunan) yang memiliki korelasi 0.71 dengan *SalePrice*, yang berarti semakin besar luas bangunan, semakin tinggi harga rumah. *GarageCars* (Kapasitas Garasi) dan *TotalBsmtSF* (Luas Ruang Bawah Tanah) juga menunjukkan korelasi positif yang kuat dengan harga rumah, masing-masing dengan nilai 0.64 dan 0.61. Ini menunjukkan bahwa rumah dengan kapasitas garasi yang lebih banyak dan ruang bawah tanah yang lebih luas cenderung memiliki harga jual yang lebih tinggi.

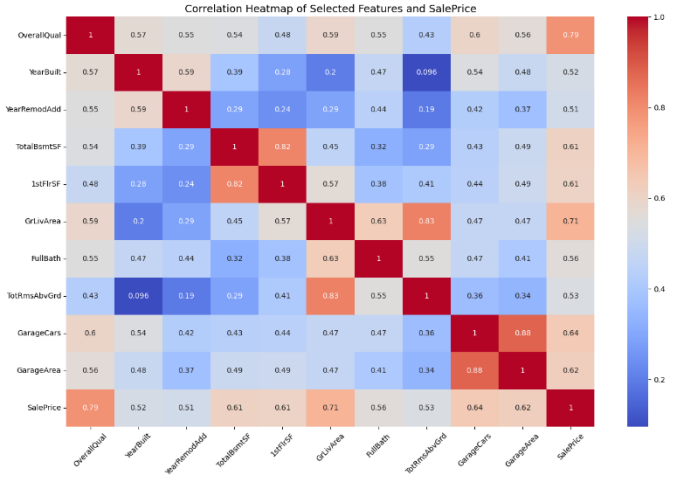
b. Korelasi Negatif:
Beberapa fitur menunjukkan korelasi negatif dengan harga rumah, artinya peningkatan nilai pada fitur ini cenderung menurunkan harga rumah.

c. Korelasi Lemah:
Beberapa fitur menunjukkan korelasi yang relatif lebih lemah dengan SalePrice. Misalnya, *TotRmsAbvGrd* (Jumlah Ruang di Atas Tanah) memiliki korelasi 0.53 dengan harga rumah, yang lebih rendah jika dibandingkan dengan fitur-fitur lainnya seperti *GrLivArea* (Luas Bangunan) dan *OverallQual* (Kualitas Rumah).

d. Korelasi Antar Fitur Lainnya:
Beberapa fitur menunjukkan korelasi yang sangat tinggi satu sama lain. Sebagai contoh, *TotalBsmtSF* (Luas Ruang Bawah Tanah) dan *1stFlrSF* (Luas Lantai Pertama) memiliki korelasi 0.82, yang menunjukkan

p-ISSN 2301–4156 | e-ISSN 2460–5719

bahwa rumah dengan ruang bawah tanah yang lebih luas cenderung memiliki lantai pertama yang lebih besar. Demikian juga, *GrLivArea* (Luas Bangunan) berkorelasi sangat kuat dengan *TotRmsAbvGrd* (Jumlah Ruang di Atas Tanah), dengan nilai korelasi 0.83, yang menunjukkan bahwa rumah dengan lebih banyak ruang di atas tanah umumnya juga memiliki luas bangunan yang lebih besar.



3.1.3 *Data Preparation*

Persiapan data merupakan langkah krusial dalam proses pembangunan model prediksi harga rumah. Proses ini bertujuan untuk memastikan bahwa data yang digunakan bersih, terstruktur dengan baik, dan siap diproses oleh algoritma pembelajaran mesin. Pada tahap ini, data melalui beberapa tahap seperti pemilihan fitur, pembersihan data, transformasi data, serta integrasi data dari berbagai sumber. Berikut ini adalah penjelasan rinci mengenai proses persiapan data yang dilakukan dalam penelitian ini.

a. Data Filtering

Pada tahap pertama, fitur yang relevan untuk prediksi harga rumah dipilih berdasarkan korelasi dengan variabel target, *SalePrice*. Fitur-fitur yang digunakan meliputi *OverallQual* (kualitas rumah), *GrLivArea* (luas bangunan), *GarageCars* (jumlah mobil yang ditampung garasi), *GarageArea* (luas garasi), *TotalBsmtSF* (luas ruang bawah tanah), *1stFlrSF* (luas lantai pertama), *FullBath* (jumlah kamar mandi lengkap), *TotRmsAbvGrd* (jumlah kamar di atas tanah), serta *YearBuilt* dan *YearRemodAdd* (tahun pembangunan dan renovasi rumah). Pemilihan fitur ini bertujuan untuk mendapatkan variabel yang paling mempengaruhi harga rumah.

b. Data Cleaning

Langkah ini mencakup penghapusan data duplikat, penanganan nilai kosong dengan imputasi, dan deteksi serta penanganan *outlier*. Nilai kosong pada beberapa fitur, seperti *GarageCars*, *GarageArea*, dan *TotalBsmtSF*, diimputasi dengan nilai dari baris berikutnya untuk menjaga konsistensi data.

c. Data Construction

Transformasi log dilakukan pada variabel target *SalePrice* untuk mengurangi *skewness* dan membuat distribusi data lebih normal. Hal ini bertujuan untuk meningkatkan performa model prediksi harga rumah.

3.1.3.1 *Dataset Attribute Selection*

Pada bagian ini, atribut-atribut yang digunakan dalam pemodelan dipilih berdasarkan analisis data dan relevansi fitur dengan target variabel *SalePrice*. Atribut yang dipilih mencakup informasi terkait dengan kondisi rumah dan spesifikasinya yang mempengaruhi harga jual rumah.

```
def select_features_by_correlation(data, target='SalePrice', threshold=0.5):

    numeric_data = data.select_dtypes(include=['number']).copy()

    corr = numeric_data.corr()[target].abs()

    filtered_corr = corr[~((corr.index != target) & (corr.index.str.contains(target, case=False)))]

    selected = filtered_corr[(filtered_corr >= threshold) & (filtered_corr.index != target)].index.tolist()

    return selected

selected_features = select_features_by_correlation(train, target='SalePrice', threshold=0.5)

import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(16, 10))
sns.heatmap(train[selected_features + ['SalePrice']].corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap of Selected Features and SalePrice", fontsize=14)
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.show()
```

Atribut yang dipilih dimasukkan ke dalam variabel *selected_columns*, yang kemudian digunakan untuk menyaring data yang relevan untuk analisis lebih lanjut.

3.1.3.2 *Dataset Attribute Construction*

Pada tahap ini, dilakukan transformasi log pada variabel target *SalePrice* untuk mengurangi *skewness* dan membuat distribusi data menjadi lebih normal. Transformasi log digunakan untuk stabilisasi varians dan membuat model lebih mudah dalam memproses data yang tidak terdistribusi normal.

```
def fix_skewness(df, threshold=0.5):
    skewed_feats = df[selected_features].apply(lambda x: skew(x.dropna())).sort_values(ascending=False)
    skewed_feats = skewed_feats[skewed_feats > threshold]
    for feat in skewed_feats.index:
        df[feat] = np.log1p(df[feat])
    return df

train = fix_skewness(train)
test = fix_skewness(test)
```

3.1.3.2.1 Transformasi Logaritmik

Harga rumah yang terdistorsi dengan distribusi miring *(skewed)* diubah menggunakan fungsi logaritmik log(1 + x), yang menghasilkan *log(SalePrice)*.

```
train['SalePrice_Log'] = np.log1p(train['SalePrice'])
```

3.1.3.2.2 Visualisasi

Distribusi *log(SalePrice)* juga divisualisasikan untuk memeriksa perubahan distribusi setelah transformasi log, dengan menggunakan histogram dan plot kernel *density estimate* (KDE).

```
sns.histplot(train['SalePrice_Log'], kde=True,
color='orange', bins=40)
```

Dengan penggantian nama atribut ini, dataset menjadi lebih mudah dipahami dan memberikan konteks yang lebih jelas dalam analisis dan pelatihan model.

3.1.3.3 Dataset Attribute Cleaning

Bagian ini mencakup pembuangan data duplikat dan pengisian nilai kosong *(missing values)*. Proses ini penting untuk memastikan kualitas data sebelum digunakan dalam pelatihan model.

```
missing = all_data.isnull().sum()
missing = missing[missing > 0].sort_values(ascending=False)

print("\nFitur dengan nilai kosong (jika ada):")
print(missing)
```

3.1.4 Modeling

Beberapa teknik data mining diterapkan untuk membangun model, dan mungkin model perlu diuji dan sesuaikan untuk mencapai hasil yang optimal. Adapun hal yang akan dilakukan pada tahapan ini:

a. Membangun Skenario Pengujian

Membagi dataset menjadi tiga bagian utama, yaitu *train set, validation set,* dan *test set.* Pembagian ini bertujuan untuk memastikan bahwa model dilatih dengan data yang cukup, hyperparameter disesuaikan dengan data validasi untuk mencegah *overfitting*, dan akhirnya diuji dengan data yang belum pernah dilihat sebelumnya pada test set untuk menilai kemampuan generalisasi model.

Proses selanjutnya adalah normalisasi data menggunakan *StandardScaler*, yang memastikan bahwa setiap fitur memiliki skala yang sama, penting untuk model-model seperti *XGBoost*, *LightGBM*, dan *CatBoost*, yang sensitif terhadap skala data.

b. Membangun Model

Tiga model utama diterapkan: *LightGBM*, *XGBoost*, dan *CatBoost*. Semua model ini adalah algoritma *boosting* yang dapat memberikan hasil yang sangat baik dalam prediksi harga rumah. Proses tuning dilakukan dengan *GridSearchCV* untuk mencari kombinasi hyperparameter terbaik yang akan memberikan performa terbaik pada data pelatihan.

```
# LightGBM
lgb_model = lgb.LGBMRegressor(random_state=42,
verbose=-1, reg_alpha=1.0, reg_lambda=1.0)
lgb_params = {
    'n_estimators': [100, 200],
    'learning_rate': [0.01, 0.05],
    'num_leaves': [15, 31],
    'max_depth': [1, 2, 3],
}
models['lgb'] = tune_model(lgb_model, lgb_params,
X_train, y_train, "LightGBM")

# XGBoost
xgb_model = xgb.XGBRegressor(random_state=42,
reg_alpha=1.0, reg_lambda=1.0)
xgb_params = {
    'n_estimators': [100, 200],
    'learning_rate': [0.01, 0.05],
    'max_depth': [1, 2],
}
models['xgb'] = tune_model(xgb_model, xgb_params,
X_train, y_train, "XGBoost")

# CatBoost
cat_model = cb.CatBoostRegressor(silent=True,
random_state=42)
cat_params = {
    'iterations': [200, 300],
    'learning_rate': [0.01, 0.05],
    'depth': [3, 4],
    'l2_leaf_reg': [5, 10],
}
models['cat'] = tune_model(cat_model, cat_params,
X_train, y_train, "CatBoost")
```

3.1.4.1 Skenario Pengujian

Bagian ini bertujuan untuk mempersiapkan dataset dengan mempertimbangkan bobot setiap kelas untuk menangani ketidakseimbangan data, melakukan pembagian dataset menjadi tiga bagian yaitu, train set digunakan untuk melatih model, validation set digunakan untuk menyetel hyperparameter dan menghindari overfitting, dan test set untuk mengevaluasi performa akhir model. Normalisasi data dengan StandardScaler juga dilakukan.

3.1.4.2 Membangun Model (Prediksi Harga Rumah)

Setelah melatih model-model individu, kita menggabungkan hasil dari ketiga model tersebut menggunakan teknik *stacking*. Teknik stacking ini menggabungkan prediksi dari masing-masing model *(LightGBM, XGBoost, CatBoost)* dan melatih **meta-model** untuk menghasilkan prediksi akhir. Meta-model yang digunakan di sini adalah *Ridge regression.*

```
def stacking(models, X_train, y_train, X_val, y_val):
    train_preds =
np.column_stack([models[m].predict(X_train) for m in
models])
    val_preds =
np.column_stack([models[m].predict(X_val) for m in
models])

    meta_model = Ridge(alpha=1.0)
    meta_model.fit(train_preds, y_train)

    train_meta_pred = meta_model.predict(train_preds)
    val_meta_pred = meta_model.predict(val_preds)

    return meta_model, train_meta_pred, val_meta_pred

models = train_models(X_train, y_train)
```


p-ISSN 2301–4156 | e-ISSN 2460–5719

```
meta_model, train_stack_pred, val_stack_pred =
stacking(models, X_train, y_train, X_val, y_val)
```

3.1.4.3 Model Terbaik

Model terbaik adalah model yang diperoleh melalui *stacking* ketiga model *(LightGBM, XGBoost,* dan *CatBoost)* dengan *Ridge regression* sebagai meta-model. Kombinasi parameter terbaik yang ditemukan selama proses tuning adalah:

- LightGBM: *learning_rate* = 0.05, *max_depth* = 3, *n_estimators* = 200, *num_leaves* = 15
- XGBoost: *learning_rate* = 0.05, max_depth = 2, *n_estimators* = 200
- CatBoost: *depth* = 4, *iterations* = 300, l2_*leaf_reg* = 5, *learning_rate* = 0.05

```
def tune_model(model, param_grid, X_train, y_train, name=""):
    print(f"Tuning {name}...")
    grid = GridSearchCV(model, param_grid, scoring='neg_root_mean_squared_error', cv=10, verbose=0)
    grid.fit(X_train, y_train)
    print(f"Best params for {name}: {grid.best_params_}")
  return grid.best_estimator_
```

Model terbaik yang diperoleh dari *GridSearchCV* digunakan untuk prediksi lebih lanjut dan evaluasi pada test set.

3.1.5 Evaluation

Evaluasi model dilakukan untuk menilai seberapa baik model yang dibangun mampu memprediksi nilai target pada data pelatihan dan data validasi. Pada eksperimen ini, evaluasi dilakukan menggunakan fungsi evaluate_predictions()

.

3.1.5.1 Evaluasi Prediksi

Fungsi evaluate_predictions() digunakan untuk menghitung metrik-metrik evaluasi yang umum digunakan dalam regresi:

```
evaluate_predictions(y_train, train_stack_pred, y_val, val_stack_pred, name="Stacked Model")
```

Metrik Evaluasi:

- RMSE (Root Mean Squared Error) mengukur rata-rata selisih kuadrat antara nilai prediksi dan nilai aktual. Metrik ini memberikan informasi tentang seberapa jauh prediksi dari nilai sebenarnya.
- MAE (Mean Absolute Error) mengukur rata-rata selisih absolut antara prediksi dan nilai aktual. MAE memberikan gambaran yang lebih sederhana mengenai kesalahan model.
- R-squared (R^2) mengukur seberapa besar proporsi variansi dalam data yang dapat dijelaskan oleh model. Nilai R-squared yang lebih tinggi menunjukkan bahwa model dapat menjelaskan lebih banyak variansi dari data target.

3.1.6 Deployment

Deployment adalah proses implementasi dan integrasi model machine learning ke dalam sistem nyata, sehingga model dapat digunakan untuk memberikan prediksi secara langsung dalam kondisi dunia nyata. Dalam studi kasus ini, *deployment* model prediksi harga rumah bertujuan agar model dapat memberikan estimasi harga rumah secara otomatis saat diperlukan.

Pentingnya Deployment Model:

1. Efektivitas Penerapan: Deployment memungkinkan model untuk memberikan prediksi secara langsung. Model yang telah dilatih dapat diintegrasikan ke dalam aplikasi atau sistem yang digunakan oleh pengguna untuk memprediksi harga rumah berdasarkan data yang dimasukkan.
2. Maksimalisasi Nilai Model: Dengan deployment, model dapat digunakan secara praktis dalam sistem lain, seperti aplikasi penjualan rumah atau sistem pencarian properti. Hal ini mengoptimalkan nilai model yang telah dibangun, karena model dapat memberikan nilai nyata dalam skenario dunia nyata.

Tantangan dalam Implementasi Deployment:

- *Reliability* (Keandalan): Model harus berjalan tanpa gangguan atau kesalahan saat digunakan dalam sistem nyata. Ini memastikan bahwa aplikasi prediksi harga rumah selalu memberikan hasil yang akurat.
- *Reusability* (Dapat Digunakan Ulang): Model harus dirancang agar dapat digunakan kembali untuk kasus lain, seperti memperkirakan harga rumah di pasar yang berbeda atau di daerah lain.
- Maintainability (Kemudahan Pemeliharaan): Model harus mudah dipelihara dan diperbarui seiring waktu, agar tetap relevan dengan perubahan data pasar properti.
- *Flexibility* (Fleksibilitas): Model harus dapat beradaptasi dengan perubahan kebutuhan sistem, seperti penambahan fitur baru atau perubahan dalam metode penilaian harga rumah.

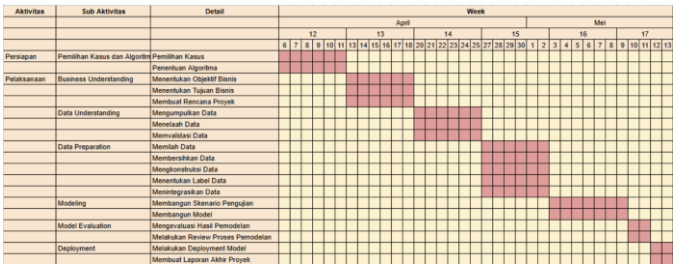
Tantangan Spesifik pada Machine Learning:

- *Reproducibility* (Reproduksibilitas): Hasil prediksi model harus konsisten, meskipun dijalankan di lingkungan yang berbeda. Ini penting agar prediksi harga rumah dapat dilakukan dengan akurat di berbagai sistem dan perangkat.

```
from flask import Flask, render_template, request
import joblib
import numpy as np
import pandas as pd

app = Flask(__name__)

# Memuat model yang telah dilatih
MODEL_PATH = r"D:\sem 6\Data Mining\HousePriceProyek\HousePriceProyek\stacked_model.pkl"
loaded_model = joblib.load(MODEL_PATH) # loaded_model is a dictionary

# Fungsi untuk memprediksi harga rumah
def predict_house_price(features):
    # Hanya fitur yang digunakan saat training
    feature_names = [
        'OverallQual', 'YearBuilt', 'YearRemodAdd', 'TotalBsmtSF',
        '1stFlrSF', 'GrLivArea', 'FullBath', 'TotRmsAbvGrd',
        'GarageCars', 'GarageArea'
    ]

    # Load model & scaler dari dictionary
    meta_model = loaded_model['meta_model']
```

```
        base_models = loaded_model['base_models']
        scaler = loaded_model['scaler']

        # Konversi input ke DataFrame
        features_df = pd.DataFrame([features], columns=feature_names)

        # Scaling
        features_scaled = scaler.transform(features_df)

        # Prediksi base models
        meta_features = np.column_stack([
            base_models[m].predict(features_scaled) for m in base_models
        ])

        # Prediksi akhir
        final_prediction = meta_model.predict(meta_features)[0]
        return np.expm1(final_prediction)  # Transformasi balik log1p

    @app.route('/')
    def index():
        return render_template('index.html')

    @app.route('/predict', methods=['POST'])
    def predict():
        if request.method == 'POST':
            # Ambil input dari form (tanpa GarageYrBlt)
            overall_quality = int(request.form['overall_quality'])
            year_built = int(request.form['year_built'])
            year_remod_add = int(request.form['year_remod_add'])
            total_bsmt_sf = int(request.form['total_bsmt_sf'])
            first_flr_sf = int(request.form['first_flr_sf'])
            gr_liv_area = int(request.form['gr_liv_area'])
            full_bath = int(request.form['full_bath'])
            total_rooms_abv_grd = int(request.form['total_rooms_abv_grd'])
            garage_cars = int(request.form['garage_cars'])
            garage_area = int(request.form['garage_area'])

            # Susun dictionary fitur
            features = {
                'OverallQual': overall_quality,
                'YearBuilt': year_built,
                'YearRemodAdd': year_remod_add,
                'TotalBsmtSF': total_bsmt_sf,
                '1stFlrSF': first_flr_sf,
                'GrLivArea': gr_liv_area,
                'FullBath': full_bath,
                'TotRmsAbvGrd': total_rooms_abv_grd,
                'GarageCars': garage_cars,
                'GarageArea': garage_area
            }

            predicted_price = predict_house_price(features)
            return render_template('index.html', predicted_price=predicted_price)

    if __name__ == "__main__":
        app.run(debug=True, port=5001)
```

3.2 Timeline



1. Pemilihan Kasus dan Algoritma

   Aktivitas ini dilakukan pada awal minggu pertama April, berfokus pada pemilihan studi kasus prediksi harga rumah dan algoritma yang sesuai untuk pemodelan.

2. Business Understanding

   Dilaksanakan pada minggu pertama hingga kedua April, tahap ini bertujuan untuk menentukan objektif bisnis dan tujuan proyek, serta menyusun perencanaan pelaksanaan proyek.

3. Data Understanding

   Berlangsung dari minggu kedua hingga ketiga April. Kegiatan pada tahap ini mencakup pengumpulan data, pemeriksaan struktur data, dan pemahaman awal terhadap karakteristik data.

4. Data Preparation

   Dimulai pada minggu ketiga April hingga awal minggu keempat. Proses ini melibatkan pembersihan data, transformasi, penggabungan, hingga menyimpan data dalam format siap digunakan untuk modeling.

5. Modeling

   Aktivitas modeling dimulai pada akhir April hingga awal Mei. Pada fase ini, model dibangun menggunakan data yang telah dipersiapkan, disertai evaluasi terhadap skenario pengujian.

6. Model Evaluation

   Model yang telah dibangun kemudian diuji dan dievaluasi pada minggu kedua Mei. Tahapan ini meliputi pengukuran performa model berdasarkan metrik evaluasi dan analisis hasil.

7. Deployment

   Tahap ini dilakukan pada pertengahan Mei. Model yang sudah diuji kemudian disimpan dan di-deploy menggunakan library pemodelan, agar bisa digunakan dalam aplikasi nyata.

8. Laporan Akhir

   Penyusunan laporan dilakukan pada minggu ketiga Mei sebagai penutup proyek. Tahap ini mendokumentasikan

p-ISSN 2301–4156 | e-ISSN 2460–5719

seluruh proses, hasil, dan rekomendasi dari studi kasus prediksi harga rumah.

IV. HASIL DAN PEMBAHASAN

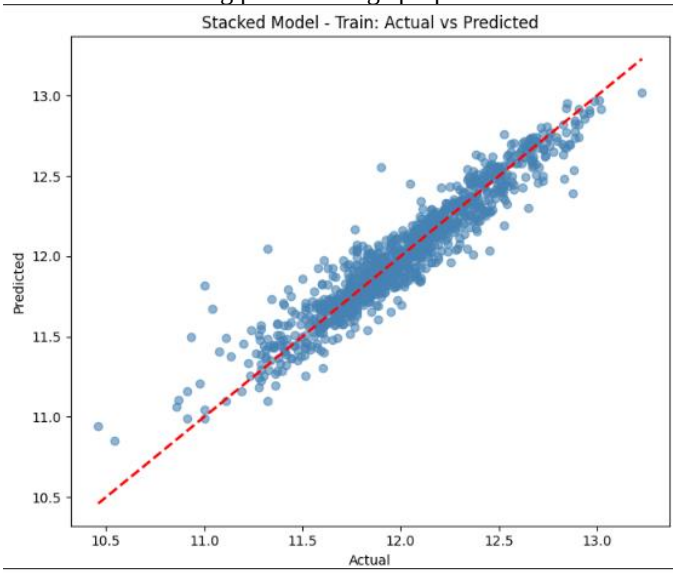
Model prediksi harga rumah yang dikembangkan dalam penelitian ini menggunakan pendekatan stacking ensemble, yaitu dengan menggabungkan tiga model terbaik—LightGBM, XGBoost, dan CatBoost—dengan Ridge Regression sebagai meta-model. Setiap model dilatih secara terpisah menggunakan hyperparameter optimal yang diperoleh melalui proses Grid Search dan validasi silang (10-fold cross validation). Adapun kombinasi hyperparameter terbaik yang ditemukan adalah sebagai berikut: untuk LightGBM, model bekerja paling optimal dengan learning_rate sebesar 0.05, max_depth 3, n_estimators 200, dan num_leaves 15. Sementara itu, XGBoost memberikan hasil terbaik dengan learning_rate 0.05, max_depth 2, dan n_estimators 200. Untuk CatBoost, konfigurasi terbaik meliputi learning_rate 0.05, depth 4, iterations 300, dan l2_leaf_reg sebesar 5.

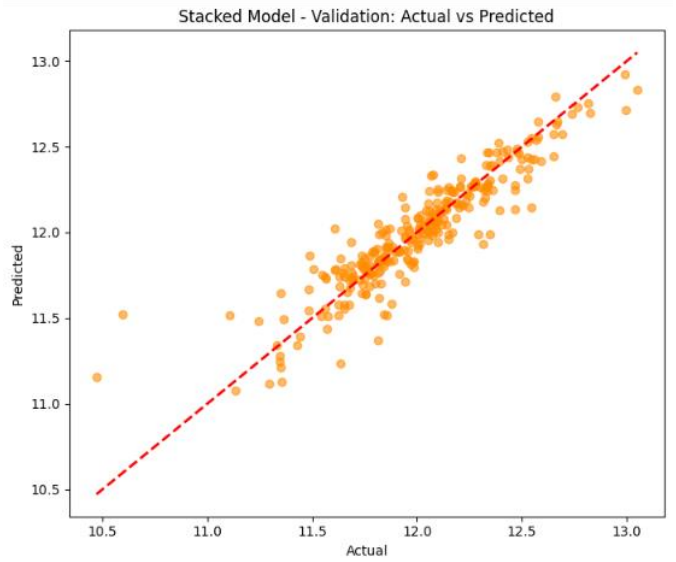
Evaluasi kinerja model dilakukan pada data pelatihan (training set) dan data validasi (validation set) menggunakan tiga metrik utama, yaitu RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), dan R^2 (koefisien determinasi). Pada data pelatihan, model mencatatkan nilai RMSE sebesar 0.1213, MAE sebesar 0.0873, dan R^2 sebesar 0.9001. Sementara itu, pada data validasi, nilai RMSE adalah 0.1465, MAE sebesar 0.0996, dan R^2 mencapai 0.8367. Nilai R^2 yang tinggi baik pada pelatihan maupun validasi menunjukkan bahwa model mampu menjelaskan sebagian besar variabilitas harga rumah dengan sangat baik. Selain itu, perbedaan metrik antara data pelatihan dan validasi tidak terlalu besar, yang mengindikasikan bahwa model tidak mengalami overfitting dan memiliki kemampuan generalisasi yang baik terhadap data baru.

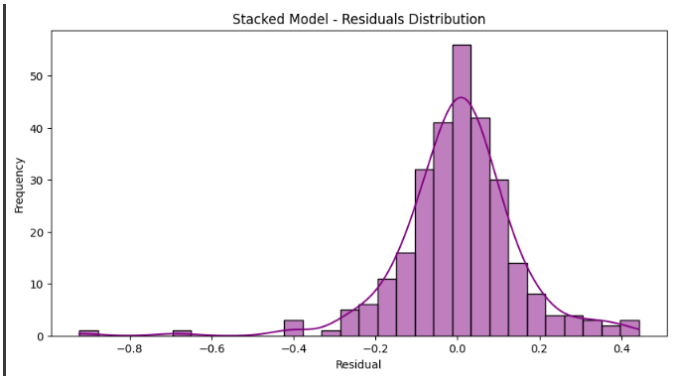
Model ini juga menunjukkan performa yang sangat baik dalam hal kesalahan prediksi. Nilai RMSE dan MAE yang rendah pada kedua dataset menunjukkan bahwa selisih antara nilai prediksi dan harga aktual rumah tergolong kecil. Hal ini menunjukkan bahwa model mampu memprediksi dengan akurasi yang tinggi dan konsistensi yang baik. Secara visual, hasil prediksi terhadap harga rumah dibandingkan dengan nilai aktual menunjukkan bahwa sebagian besar titik prediksi berada dekat dengan garis diagonal pada grafik scatter plot, yang berarti prediksi mendekati nilai aktual. Selain itu, analisis distribusi residual memperlihatkan penyebaran yang simetris di sekitar nol tanpa pola tertentu, yang mengindikasikan bahwa model tidak memiliki bias sistematis.

Berdasarkan keseluruhan hasil tersebut, dapat disimpulkan bahwa model stacking ensemble yang dibangun sangat layak dan efektif untuk digunakan dalam prediksi harga rumah. Kombinasi dari tiga model boosting yang kuat dengan Ridge Regression sebagai meta-learner menghasilkan model akhir yang seimbang, stabil, dan memiliki kemampuan prediksi yang tinggi. Model ini cocok untuk diimplementasikan pada sistem nyata yang membutuhkan estimasi harga rumah secara cepat dan akurat, seperti pada platform properti digital, sistem penilaian aset, atau aplikasi agen real estat.

Meski demikian, masih terdapat ruang untuk pengembangan lebih lanjut. Salah satu rekomendasi adalah melakukan eksplorasi tambahan pada tahap feature engineering untuk menambahkan informasi yang lebih kaya dan relevan ke dalam model. Selain itu, penggunaan meta-model lain, seperti Gradient Boosting sebagai alternatif Ridge Regression, juga bisa dieksplorasi untuk melihat apakah dapat memberikan hasil yang lebih baik. Dengan pendekatan yang tepat, model ini memiliki potensi besar untuk terus ditingkatkan dan memberikan nilai tambah dalam bidang prediksi harga properti berbasis data.







IV. KESIMPULAN DAN SARAN

Definisikan singkatan dan akronim saat pertama kali disebutkan dalam teks, walaupun telah didefinisikan dalam intisari. Singkatan seperti IEEE, AC, dan DC tidak harus didefinisikan. Singkatan yang menyertakan titik tidak boleh dipisahkan oleh spasi, sehingga “C.N.R.S.,” bukan “C. N. R. S.” Jangan gunakan singkatan dalam judul kecuali jika tidak dapat dihindari.

5.1 Kesimpulan

Sistem prediksi harga rumah ini dikembangkan untuk membantu pengguna dalam memperkirakan harga properti berdasarkan sejumlah variabel penting, seperti lokasi, luas tanah dan bangunan, jumlah kamar, kondisi rumah, serta faktor-faktor lainnya yang relevan. Model machine learning yang digunakan dilatih menggunakan dataset yang representatif, dan hasil pelatihan disimpan dalam file model (mdl.pkl) menggunakan pustaka dill.

Aplikasi ini kemudian diimplementasikan dalam bentuk web menggunakan framework Flask. Melalui antarmuka web, pengguna dapat memilih atau mengisi data properti seperti lokasi, luas tanah, jumlah kamar tidur, dan fitur lainnya, kemudian menekan tombol “Prediksi” untuk melihat estimasi harga rumah berdasarkan input tersebut.

Sistem akan memproses data input dengan mengonversinya menjadi format numerik, dan kemudian menggunakan model yang telah dilatih untuk menghasilkan prediksi harga. Hasil prediksi ditampilkan secara informatif dalam bentuk nilai estimasi harga rumah, yang dapat digunakan sebagai referensi dalam pengambilan keputusan pembelian, penjualan, atau investasi properti.

Sistem ini diharapkan dapat menjadi alat bantu yang bermanfaat bagi masyarakat umum, agen properti, maupun pengembang perumahan dalam memahami pasar perumahan dan membuat keputusan yang lebih tepat berbasis data.

5.2 Saran

Berikut beberapa saran untuk pengembangan sistem prediksi harga rumah ke depannya:

9. Untuk meningkatkan akurasi model, dapat dilakukan peningkatan kualitas dan kuantitas data, seperti memperluas cakupan wilayah, memasukkan data historis, serta mempertimbangkan tren pasar perumahan terbaru.
10. Aplikasi dapat dilengkapi dengan fitur visualisasi data seperti grafik tren harga per wilayah, peta lokasi properti, atau perbandingan harga rumah serupa.
11. Dari sisi pengalaman pengguna, disarankan untuk menambahkan fitur validasi input dan pemberian pesan kesalahan (error message) yang lebih informatif, agar pengguna memahami apabila terjadi kesalahan dalam proses input atau prediksi.
12. Jika aplikasi ini digunakan oleh banyak pengguna atau diintegrasikan dalam sistem komersial, penting untuk memperhatikan aspek keamanan sistem, seperti penggunaan autentikasi, enkripsi data pengguna, serta perlindungan terhadap serangan siber.

V. PEMBAGIAN PEKERJAAN

Anggota Kelompok	Bagian yang Dikerjakan
Jeremy Samosir	Business Understanding, Data Preparation, dan Evalution Model
Angelina Nadeak	Data Overview dan Data Understanding
Ade Siahaan	Data Understanding, Modelling, dan Deployment
Rosari Simanjuntak	Data Overview dan Data Understanding

REFERENSI

[1] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[2] D. H. Wolpert, "Stacked Generalization," *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992.

[3] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785-794.

[4] J. Zhang, Z. Xu, and W. Liu, "Predicting House Prices Using Machine Learning Algorithms: A Review," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 4532859, 2020, pp. 1-15.

p-ISSN 2301–4156 | e-ISSN 2460–5719