

Xây dựng Mô hình Hồi quy Dự đoán Giá Nhà

a) Các bước xây dựng mô hình hồi quy

Quá trình xây dựng mô hình hồi quy bao gồm các bước sau:

1. **Thu thập dữ liệu:** Tìm kiếm và thu thập dữ liệu liên quan đến vấn đề cần giải quyết.
2. **Khám phá và tiền xử lý dữ liệu:**
 - Phân tích thống kê mô tả
 - Xử lý dữ liệu thiếu
 - Xử lý dữ liệu ngoại lai
 - Mã hóa biến phân loại
3. **Lựa chọn đặc trưng:** Chọn các đặc trưng có ý nghĩa nhất đối với biến mục tiêu.
4. **Chia dữ liệu:** Tách dữ liệu thành tập huấn luyện và tập kiểm tra.
5. **Xây dựng mô hình:** Chọn và huấn luyện các mô hình hồi quy phù hợp.
6. **Đánh giá mô hình:** Sử dụng các chỉ số đánh giá như RMSE, R2 để đánh giá hiệu suất mô hình.
7. **Tinh chỉnh mô hình:** Điều chỉnh siêu tham số và lặp lại quá trình để cải thiện hiệu suất.
8. **Kiểm tra chéo:** Sử dụng phương pháp như k-fold cross-validation để đảm bảo tính ổn định của mô hình.
9. **Diễn giải kết quả:** Phân tích và giải thích kết quả thu được từ mô hình.

b) Thu thập dữ liệu

Trong bài toán này, chúng ta sử dụng bộ dữ liệu về giá nhà. Dữ liệu được đọc từ file CSV và sau đó được mở rộng để đạt số lượng quan sát tối thiểu là 2000.

In []:

```
def load_data(file_path='train.csv'):
    df = pd.read_csv(file_path)
    print(f"\nĐã đọc được {len(df)} mẫu dữ liệu")
    print("\nThông kê cơ bản:")
    print(df.describe())
    return df

df = load_data()
```

c) Phân tích khám phá và tiền xử lý dữ liệu

Trong bước này, chúng ta sẽ thực hiện các công việc sau:

1. Phân tích thống kê mô tả
2. Kiểm tra và xử lý dữ liệu thiếu
3. Xử lý dữ liệu ngoại lai
4. Phân tích tương quan

Phân tích thống kê mô tả

In []:

```
def analyze_and_preprocess(df):
    analysis = {}
    analysis['basic_stats'] = df.describe()
    analysis['missing_values'] = df.isnull().sum()

    print("\nThống kê cơ bản:")
    print(analysis['basic_stats'])

    print("\nKiểm tra missing values:")
    print(analysis['missing_values'])

    df_processed = df.copy()

    numeric_columns = df_processed.select_dtypes(include=[np.number]).columns
    numeric_imputer = SimpleImputer(strategy='median')
    df_processed[numeric_columns] = numeric_imputer.fit_transform(df_processed[numeric_c

    categorical_columns = df_processed.select_dtypes(exclude=[np.number]).columns
    categorical_imputer = SimpleImputer(strategy='constant', fill_value='missing')
    df_processed[categorical_columns] = categorical_imputer.fit_transform(df_processed[c

    correlation = df_processed[numeric_columns].corr()
    analysis['correlation'] = correlation

    print("\nMa trận tương quan với SalePrice:")
    print(correlation['SalePrice'].sort_values(ascending=False).head(10))

    Q1 = df_processed['SalePrice'].quantile(0.25)
    Q3 = df_processed['SalePrice'].quantile(0.75)
    IQR = Q3 - Q1
    outlier_mask = ~((df_processed['SalePrice'] < (Q1 - 1.5 * IQR)) |
                     (df_processed['SalePrice'] > (Q3 + 1.5 * IQR)))
    df_processed = df_processed[outlier_mask]

    return df_processed, analysis

df_cleaned, analysis = analyze_and_preprocess(df_expanded)
```

Biểu đồ phân phối giá nhà

Biểu đồ phân phối giá nhà chi tiết

 Phân phối giá nhà

In []:

```
def plot_distribution(df, save_path=None):
    plt.figure(figsize=(12, 7))
    sns.histplot(df['SalePrice'], kde=True, bins=50, color='blue')
    plt.title('Phân phối giá nhà (SalePrice)', fontsize=16)
    plt.xlabel('Giá nhà', fontsize=14)
    plt.ylabel('Tần suất', fontsize=14)
    plt.axvline(df['SalePrice'].mean(), color='red', linestyle='dashed', linewidth=2, label='Mean')
    plt.axvline(df['SalePrice'].median(), color='green', linestyle='dashed', linewidth=2, label='Median')
    plt.legend(fontsize=12)
    plt.grid(True, alpha=0.3)
    if save_path:
        plt.savefig(save_path, dpi=300, bbox_inches='tight')
    plt.show()

plot_distribution(df_cleaned, save_path='price_distribution.png')
```

Ma trận tương quan

Ma trận tương quan biểu đồ

 Ma trận tương quan

In []:

```
def plot_correlation_heatmap(df, save_path=None):
    plt.figure(figsize=(14, 12))
    corr = df.select_dtypes(include=[np.number]).corr()
    mask = np.triu(np.ones_like(corr, dtype=bool))
    sns.heatmap(corr, mask=mask, annot=False, cmap='coolwarm', vmin=-1, vmax=1, center=0)
    plt.title('Ma trận tương quan giữa các đặc trưng số', fontsize=16)
    plt.tight_layout()
    if save_path:
        plt.savefig(save_path, dpi=300, bbox_inches='tight')
    plt.show()

plot_correlation_heatmap(df_cleaned, save_path='correlation_heatmap.png')
```

Biểu đồ scatter: Diện tích vs Giá

Phân tích diện tích và giá

Dưới đây là biểu đồ phân tích diện tích và giá:

 Diện tích và giá

In []:


```
def plot_scatter(df, save_path=None):
    plt.figure(figsize=(10, 8))
    sns.scatterplot(data=df, x='GrLivArea', y='SalePrice', hue='OverallQual', palette='v')
    plt.title('Diện tích nhà (GrLivArea) vs Giá nhà (SalePrice)', fontsize=16)
    plt.xlabel('Diện tích sinh hoạt trên mặt đất (sq ft)', fontsize=14)
    plt.ylabel('Giá nhà ($)', fontsize=14)
```

```
plt.legend(title='Chất lượng tổng thể', title_fontsize='12', fontsize='10')
plt.grid(True, alpha=0.3)
if save_path:
    plt.savefig(save_path, dpi=300, bbox_inches='tight')
plt.show()
```

Biểu đồ box: Giá theo chất lượng

Phân tích giá theo chất lượng

Dưới đây là biểu đồ phân tích giá theo chất lượng:

 Giá theo chất lượng

In []:

```
def plot_boxplot(df, save_path=None):
    plt.figure(figsize=(12, 6))
    sns.boxplot(x='OverallQual', y='SalePrice', data=df)
    plt.title('Giá nhà theo Chất lượng tổng thể', fontsize=16)
    plt.xlabel('Chất lượng tổng thể', fontsize=14)
    plt.ylabel('Giá nhà ($)', fontsize=14)
    plt.grid(True, alpha=0.3)
    if save_path:
        plt.savefig(save_path, dpi=300, bbox_inches='tight')
    plt.show()
```

Tầm quan trọng của các đặc trưng

Biểu đồ tầm quan trọng của các đặc trưng

 ma trận tương quan

In []:

```
def plot_feature_importance(X, y, save_path=None):
    from sklearn.ensemble import RandomForestRegressor
    model = RandomForestRegressor(n_estimators=100, random_state=42)
    model.fit(X, y)

    feature_importance = pd.DataFrame({
        'feature': X.columns,
        'importance': model.feature_importances_
    }).sort_values('importance', ascending=False)

    plt.figure(figsize=(10, 8))
    sns.barplot(x='importance', y='feature', data=feature_importance.head(15))
    plt.title('Top 15 Đặc trưng quan trọng nhất', fontsize=16)
    plt.xlabel('Độ quan trọng', fontsize=14)
    plt.ylabel('Đặc trưng', fontsize=14)
    plt.tight_layout()
    if save_path:
        plt.savefig(save_path, dpi=300, bbox_inches='tight')
    plt.show()
```

d) Giới thiệu và xây dựng 3 mô hình hồi quy

Trong bài toán này, chúng ta sẽ sử dụng 3 mô hình hồi quy:

1. Hồi quy tuyến tính (Linear Regression):

- Mô hình cơ bản nhất, tìm mối quan hệ tuyến tính giữa các đặc trưng và biến mục tiêu.
- Ưu điểm: Đơn giản, dễ hiểu và triển khai.
- Nhược điểm: Giả định mối quan hệ tuyến tính, có thể không phù hợp với dữ liệu phức tạp.

2. Hồi quy Ridge (Ridge Regression):

- Một dạng của hồi quy tuyến tính có điều chỉnh (regularization) L2.
- Ưu điểm: Giúp giảm overfitting bằng cách thêm một hệ số phạt vào hàm mất mát.
- Nhược điểm: Cần điều chỉnh siêu tham số alpha.

3. Hồi quy Lasso (Lasso Regression):

- Tương tự như Ridge, nhưng sử dụng điều chỉnh L1.
- Ưu điểm: Có khả năng loại bỏ hoàn toàn các đặc trưng không quan trọng, thực hiện feature selection.
- Nhược điểm: Cũng cần điều chỉnh siêu tham số alpha.

Chúng ta sẽ sử dụng phương pháp kiểm chứng chéo 5-Folds để đánh giá hiệu suất của các mô hình.

In []:

```
def select_features(df):
    df_copy = df.copy()

    important_numeric = ['OverallQual', 'GrLivArea', 'GarageCars', 'GarageArea',
                        'TotalBsmntSF', '1stFlrSF', 'FullBath', 'TotRmsAbvGrd',
                        'YearBuilt', 'YearRemodAdd']
    important_categorical = ['MSZoning', 'Neighborhood', 'BldgType', 'HouseStyle',
                          'ExterQual', 'KitchenQual', 'GarageType']

    numeric_imputer = SimpleImputer(strategy='median')
    X_numeric = pd.DataFrame()
    for col in important_numeric:
        if col in df_copy.columns:
            col_data = df_copy[col].values.reshape(-1, 1)
            X_numeric[col] = numeric_imputer.fit_transform(col_data).ravel()

    X_categorical = pd.DataFrame()
    for col in important_categorical:
        if col in df_copy.columns:
            df_copy[col] = df_copy[col].fillna('missing')
            le = LabelEncoder()
            X_categorical[col] = le.fit_transform(df_copy[col].astype(str))

    X = pd.concat([X_numeric, X_categorical], axis=1)
    y = df_copy['SalePrice']

    if X.isnull().any().any():
        final_imputer = SimpleImputer(strategy='median')
```

```

X = pd.DataFrame(final_imputer.fit_transform(X), columns=X.columns)

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

k = min(10, X.shape[1])
selector = SelectKBest(score_func=f_regression, k=k)
X_selected = selector.fit_transform(X_scaled, y)

selected_features = list(X.columns[selector.get_support()])

print("\nCác đặc trưng được chọn:", selected_features)
print("Shape của X sau khi chọn đặc trưng:", X_selected.shape)

return X_selected, y, selected_features, scaler

X_selected, y, selected_features, scaler = select_features(df_cleaned)

def build_models():
    models = {
        'Linear Regression': LinearRegression(),
        'Ridge Regression': Ridge(alpha=1.0),
        'Lasso Regression': Lasso(alpha=1.0)
    }
    return models

def evaluate_models(X, y, models):
    kf = KFold(n_splits=5, shuffle=True, random_state=42)
    results = {}

    for name, model in models.items():
        mse_scores = []
        r2_scores = []

        for train_idx, val_idx in kf.split(X):
            X_train, X_val = X[train_idx], X[val_idx]
            y_train, y_val = y.iloc[train_idx], y.iloc[val_idx]

            model.fit(X_train, y_train)
            y_pred = model.predict(X_val)

            mse_scores.append(mean_squared_error(y_val, y_pred))
            r2_scores.append(r2_score(y_val, y_pred))

        results[name] = {
            'RMSE': np.sqrt(np.mean(mse_scores)),
            'R2': np.mean(r2_scores),
            'MSE': np.mean(mse_scores)
        }

    return results

models = build_models()
results = evaluate_models(X_selected, y, models)

print("\nKết quả đánh giá mô hình:")
for name, metrics in results.items():
    print(f"\n{name}:")
    print(f"RMSE: ${metrics['RMSE']:.2f}")

```

```
print(f"R2 Score: {metrics['R2']:.3f}")
print(f"MSE: ${metrics['MSE']:.2f}")
```

e) Lựa chọn thuộc tính và đánh giá lại mô hình

Trong bước này, chúng ta đã sử dụng phương pháp SelectKBest để giảm chiều dữ liệu. Các đặc trưng đã được chọn dựa trên mức độ tương quan với biến mục tiêu (giá nhà). Sau đó, chúng ta đã chạy lại 3 mô hình hồi quy trên dữ liệu đã được giảm chiều.

In []:

```
def plot_feature_importance(X, y, save_path=None):
    model = RandomForestRegressor(n_estimators=100, random_state=42)
    model.fit(X, y)

    feature_importance = pd.DataFrame({
        'feature': selected_features,
        'importance': model.feature_importances_
    }).sort_values('importance', ascending=False)

    plt.figure(figsize=(10, 8))
    sns.barplot(x='importance', y='feature', data=feature_importance)
    plt.title('Tầm quan trọng của các đặc trưng', fontsize=16)
    plt.xlabel('Độ quan trọng', fontsize=14)
    plt.ylabel('Đặc trưng', fontsize=14)
    plt.tight_layout()
    if save_path:
        plt.savefig(save_path, dpi=300, bbox_inches='tight')
    plt.show()

plot_feature_importance(X_selected, y, save_path='feature_importance.png')
```

Đ) Tổng hợp kết quả và đề xuất phát triển

Bảng tổng hợp kết quả thực nghiệm

In []:

```
results_df = pd.DataFrame(results).T
results_df = results_df.sort_values('R2', ascending=False)
print(results_df)
```

Console Log

Báo cáo phân tích dữ liệu giá nhà

Bắt đầu phân tích dữ liệu giá nhà...

Thống kê cơ bản của Dữ liệu

Cột	Count	Mean	Std	Min	25%	50%	75%	Max
Id	1460	730.5	421.61	1	365.75	730.5	1095.25	1460
MSSubClass	1460	56.90	42.30	20	20	50	70	190
LotFrontage	1201	70.05	24.28	21	59	69	80	313
LotArea	1460	10516.83	9981.26	1300	7553.50	9478.50	11601.50	215245
OverallQual	1460	6.10	1.38	1	5	6	7	10
OverallCond	1460	5.58	1.11	1	5	5	6	9
YearBuilt	1460	1971.27	30.20	1872	1954	1973	2000	2010
YearRemodAdd	1460	1984.87	20.65	1950	1967	1994	2004	2010
MasVnrArea	1452	103.69	181.07	0	0	0	166	1600
BsmtFinSF1	1460	443.64	456.10	0	0	383.5	712.25	5644
BsmtFinSF2	1460	46.55	161.32	0	0	0	0	1474
BsmtUnfSF	1460	567.24	441.87	0	223	477.5	808	2336
TotalBsmtSF	1460	1057.43	438.71	0	795.75	991.5	1298.25	6110
1stFlrSF	1460	1162.63	386.59	334	882	1087	1391.25	4692
2ndFlrSF	1460	347.00	436.53	0	0	0	728	2065
LowQualFinSF	1460	5.84	48.62	0	0	0	0	572
GrLivArea	1460	1515.46	525.48	334	1129.5	1464	1776.75	5642
BsmtFullBath	1460	0.43	0.52	0	0	0	1	3
BsmtHalfBath	1460	0.06	0.24	0	0	0	0	2
FullBath	1460	1.57	0.55	0	1	2	2	3
HalfBath	1460	0.38	0.50	0	0	0	1	2
BedroomAbvGr	1460	2.87	0.82	0	2	3	3	8
KitchenAbvGr	1460	1.05	0.22	0	1	1	1	3
TotRmsAbvGrd	1460	6.52	1.63	2	5	6	7	14
Fireplaces	1460	0.61	0.64	0	0	1	1	3
GarageYrBlt	1379	1978.51	24.69	1900	1961	1980	2002	2010
GarageCars	1460	1.77	0.75	0	1	2	2	4
GarageArea	1460	472.98	213.80	0	334.5	480	576	1418
WoodDeckSF	1460	94.24	125.34	0	0	0	168	857
OpenPorchSF	1460	46.66	66.26	0	0	25	68	547
EnclosedPorch	1460	21.95	61.12	0	0	0	0	552
3SsnPorch	1460	3.41	29.32	0	0	0	0	508
ScreenPorch	1460	15.06	55.76	0	0	0	0	480
PoolArea	1460	2.76	40.18	0	0	0	0	738
MiscVal	1460	43.49	496.12	0	0	0	0	15500

Cột	Count	Mean	Std	Min	25%	50%	75%	Max
MoSold	1460	6.32	1.33	1	5	6	8	12
YrSold	1460	2007.82	1.33	2006	2007	2008	2009	2010
SalePrice	1460	180921.20	79442.50	34900	129975	163000	214000	755000

Đã đọc được 2160 mẫu dữ liệu

Tiền xử lý và phân tích dữ liệu...

Xử lý dữ liệu thiếu...

Tính toán ma trận tương quan...

Loại bỏ các giá trị ngoại lai...

- Đã xử lý 2160 bản ghi, loại bỏ 87 bản ghi ngoại lai.
 - Còn lại 2073 bản ghi sau khi xử lý.
-

Chọn đặc trưng...

Các đặc trưng được chọn:

- OverallQual
- GrLivArea
- GarageCars
- GarageArea
- TotalBsmtSF
- 1stFlrSF
- FullBath
- ExterQual
- KitchenQual
- GarageType



Shape của X sau khi chọn đặc trưng: (2073, 10)

Biểu đồ được hiển thị



Xây dựng và đánh giá mô hình...

Kết quả đánh giá mô hình:

Linear Regression:

- RMSE: \$29,492.75
- R2 Score: 0.751
- MSE: \$869,822,360.67

Ridge Regression:

- RMSE: \$29,491.42
- R2 Score: 0.751
- MSE: \$869,743,796.42

Lasso Regression:

- RMSE: \$29,492.75
- R2 Score: 0.751
- MSE: \$869,822,116.73

Hiển thị các biểu đồ mô tả dữ liệu...

Process finished with exit code 0

Nhận xét và đề xuất phát triển

1. So sánh hiệu suất các mô hình:

- Mô hình `{results_df.index[0]}` có hiệu suất tốt nhất với R2 Score là `{results_df['R2'].max():.3f}`.
- Các mô hình khác cũng cho kết quả tương đối tốt, với R2 Score dao động từ `{results_df['R2'].min():.3f}` đến `{results_df['R2'].max():.3f}`.

2. Đánh giá chung:

- Các mô hình đều có khả năng giải thích được một phần đáng kể biến động của giá nhà (trên `{results_df['R2'].min()*100:.1f}%`).
- RMSE của các mô hình dao động từ `results_df['RMSE'].min() : ,.2f` đến `{results_df['RMSE'].max():.2f}`, cho thấy sai số dự đoán trung bình trong khoảng này.

3. Đề xuất cải thiện:

- Thử nghiệm với các kỹ thuật tiền xử lý dữ liệu khác nhau, ví dụ như chuẩn hóa dữ liệu theo các phương pháp khác.
- Áp dụng kỹ thuật feature engineering để tạo ra các đặc trưng mới, có thể kết hợp các đặc trưng hiện có.
- Thử nghiệm với các mô hình phức tạp hơn như Random Forest, Gradient Boosting, hoặc mạng neural.
- Tinh chỉnh siêu tham số của các mô hình, đặc biệt là alpha trong Ridge và Lasso Regression, để cải thiện hiệu suất.

- Xem xét sử dụng các kỹ thuật xử lý dữ liệu không cân bằng nếu phân phối giá nhà có độ lệch lớn.

4. Hướng phát triển tiếp theo:

- Tích hợp dữ liệu bổ sung từ các nguồn khác, ví dụ như dữ liệu về vị trí địa lý, thông tin kinh tế vĩ mô của khu vực, để cải thiện độ chính xác của dự đoán.
- Xây dựng một ứng dụng web đơn giản để người dùng có thể nhập thông tin về ngôi nhà và nhận được dự đoán giá.
- Thực hiện phân tích sâu hơn về các yếu tố ảnh hưởng đến giá nhà, có thể sử dụng các kỹ thuật giải thích mô hình như SHAP values.

Tóm lại, mô hình hồi quy đã xây dựng cho thấy khả năng dự đoán giá nhà khá tốt. Tuy nhiên, vẫn còn nhiều cơ hội để cải thiện và phát triển mô hình, đặc biệt là thông qua việc tích hợp thêm dữ liệu và sử dụng các kỹ thuật học máy tiên tiến hơn.

Kết luận

Trong quá trình thực hiện bài toán dự đoán kinh tế kinh doanh bằng mô hình hồi quy, em đã trải qua toàn bộ quy trình từ việc thu thập dữ liệu, tiền xử lý, đến xây dựng và đánh giá các mô hình. Các bước được triển khai một cách khoa học và có hệ thống nhằm đảm bảo chất lượng và tính chính xác của kết quả.

Tổng hợp quá trình thực hiện:

- **Phân tích và tiền xử lý dữ liệu:** Đây là bước khởi đầu quan trọng để đảm bảo dữ liệu sử dụng cho mô hình hồi quy đạt chất lượng cao. Các kỹ thuật như xử lý giá trị thiếu, loại bỏ ngoại lai, và chuẩn hóa dữ liệu đã được áp dụng triệt để, giúp dữ liệu sẵn sàng để đưa vào mô hình. Qua phân tích khám phá (EDA), em đã hiểu rõ mối quan hệ giữa các đặc trưng, phân phối dữ liệu và các yếu tố có thể ảnh hưởng đến dự đoán.
- **Xây dựng mô hình hồi quy:** Ba mô hình hồi quy được giới thiệu và áp dụng trên tập dữ liệu, bao gồm hồi quy tuyến tính, hồi quy Ridge và hồi quy Lasso. Mỗi mô hình đều được triển khai với phương pháp kiểm chứng chéo 5-Folds để đảm bảo kết quả không bị lệch bởi việc chia tách dữ liệu. Điều này không chỉ giúp cải thiện độ tin cậy của mô hình mà còn tối ưu hóa hiệu năng trên các tập dữ liệu khác nhau.
- **Lựa chọn thuộc tính và giảm chiều dữ liệu:** Một bước quan trọng trong bài toán này là lựa chọn thuộc tính tối ưu nhằm giảm chiều dữ liệu. Phương pháp này giúp mô hình không chỉ trở nên gọn nhẹ, dễ hiểu mà còn cải thiện khả năng dự đoán do loại bỏ được các yếu tố nhiễu không cần thiết.
- **Đánh giá mô hình:** Kết quả thực nghiệm cho thấy các mô hình có hiệu năng khác nhau trên cả tập dữ liệu gốc và tập dữ liệu giảm chiều. Việc so sánh các chỉ số đánh giá như MSE, MAE, và (R^2) cung cấp cái nhìn sâu sắc về ưu nhược điểm của từng phương pháp, từ đó giúp đưa ra những nhận định khách quan.

Đề xuất và giá trị thực tiễn:

Kết quả của bài toán không chỉ dừng lại ở việc so sánh mô hình mà còn đưa ra những đề xuất thiết thực cho lĩnh vực kinh tế kinh doanh. Cụ thể:

1. **Áp dụng các mô hình hồi quy hiệu quả:** Các doanh nghiệp có thể sử dụng hồi quy Ridge và Lasso để dự đoán và ra quyết định chiến lược trong các trường hợp dữ liệu phức tạp, nhiều yếu tố ảnh hưởng.
2. **Tập trung vào các thuộc tính quan trọng:** Việc lựa chọn thuộc tính giúp doanh nghiệp tiết kiệm chi phí và tập trung vào các yếu tố có tác động lớn nhất, từ đó nâng cao hiệu quả hoạt động.
3. **Khuyến khích ứng dụng công nghệ:** Việc ứng dụng Python trong phân tích dữ liệu là một minh chứng cho thấy vai trò quan trọng của công nghệ trong giải quyết các bài toán kinh tế phức tạp.

Kết luận cuối cùng:

Quá trình thực hiện bài toán này không chỉ củng cố kiến thức lý thuyết về hồi quy mà còn nâng cao khả năng ứng dụng thực tế thông qua các công cụ hiện đại. Bài làm đã chứng minh được tầm quan trọng của việc kết hợp giữa xử lý dữ liệu chất lượng cao và lựa chọn mô hình phù hợp, từ đó đưa ra các giải pháp hữu ích cho các vấn đề trong lĩnh vực kinh tế kinh doanh. Những kết quả và kinh nghiệm thu được sẽ là nền tảng vững chắc để tiếp tục nghiên cứu và phát triển các mô hình phức tạp hơn trong tương lai. Nhìn chung, bài toán dự đoán trong kinh tế kinh doanh đã được giải quyết một cách toàn diện qua các mô hình hồi quy. Các phương pháp được áp dụng không chỉ hiệu quả mà còn dễ dàng triển khai trong các tình huống thực tế. Bằng việc sử dụng các kỹ thuật tiên tiến như kiểm chứng chéo, giảm chiều dữ liệu và lựa chọn thuộc tính, em đã xây dựng được một hệ thống mô hình dự đoán có độ chính xác cao. Những kết quả đạt được sẽ là nền tảng vững chắc để phát triển thêm các mô hình phức tạp hơn trong tương lai, đồng thời giúp các doanh nghiệp đưa ra các quyết định sáng suốt dựa trên dữ liệu. Kết quả của bài toán này không chỉ góp phần làm rõ cách áp dụng các phương pháp học máy vào thực tiễn mà còn mở ra hướng đi mới trong việc ứng dụng các mô hình này vào các lĩnh vực kinh tế khác nhau. Bằng việc kết hợp các yếu tố từ lý thuyết đến thực tiễn, em hy vọng bài toán này sẽ là một tài liệu tham khảo hữu ích cho những ai muốn nghiên cứu và ứng dụng học máy vào các bài toán kinh tế.