Project Report of CENG 441 – Data Mining

Team Members

Arifali Baghirli – 210201848

Yunis Guliyev – 210201883

Sadik Emre Duzgun – 210201001

Mohammedghazi A M Hattab – 210201975

Introduction

The real estate sector serves as a fundamental component of any economy, significantly impacting the lives of individuals, businesses, and policymakers. A comprehensive understanding of market trends, property valuations, and housing dynamics is crucial for making informed decisions. Nevertheless, the intricate nature and variability of real estate data often pose considerable challenges.

In this study, we tackle these challenges by examining a dataset entitled Apartment Prices for the Azerbaijan Market, obtained from the well-known real estate platform Bina.az. This dataset comprises detailed information on 39,300 properties throughout Azerbaijan, including various attributes such as price, location, size, and additional factors. By utilizing this extensive dataset, our goal is to categorize apartments into distinct pricing tiers, including low, medium, and high, based on essential characteristics.

The primary aim of this project is to create a machine learning classification model that accurately predicts price categories. Such a model can be an invaluable resource for a range of stakeholders, including real estate professionals, investors, and prospective homebuyers, facilitating data-driven decision-making in a competitive market environment.

Additionally, this research highlights the application of data mining techniques and machine learning methodologies, illustrating their effectiveness in addressing complex, real-world issues. By implementing various classifiers and assessing their performance through standardized metrics, we seek to provide insights into the efficacy of different strategies for real estate classification.

Through this initiative, we not only enhance the understanding of the Azerbaijani real estate market but also showcase the transformative potential of data-driven analytics in converting raw data into practical knowledge.

Methods

Dataset Overview

The dataset consists of apartment price listings for the Azerbaijan market, including 10 attributes:

- **price**: The total price of the apartment (in currency units).
- **location**: Districts or metro stations where the apartments are located.
- **rooms**: The total number of rooms in the apartment.
- **square**: The total size (square footage) of the apartment.
- **floor**: The floor number of the apartment and the total floors in the building.
- **new_building**: A binary indicator of whether the apartment is in a new building (1 = Yes, 0 = No).
- **has_repair**: A binary indicator for whether the apartment has been repaired.
- **has_bill_of_sale**: A binary indicator for availability of legal sale documentation.
- **has_mortgage**: A binary indicator of whether the apartment is under mortgage.

The dataset contains **39,300 rows** and features **both numerical and categorical variables.**

Data Preprocessing

1. **Data Cleaning**
   - **Handling Missing Values**: Missing values in columns such as price, square footage, or categorical features were checked and removed or imputed appropriately.
   - **Outlier Detection**: Price and square columns were analyzed for potential outliers using statistical methods such as the interquartile range (IQR) technique. Outliers were either capped or excluded to avoid skewing model predictions.
2. **Feature Engineering**
   - **Floor Parsing**: The "floor" column, which includes both the current floor and the total floors (e.g., 17/20), was split into two features: current_floor and total_floors. This ensures better analysis and avoids mixed data types.
   - **Derived Features**: New features such as "price per square meter" were calculated by dividing the total price by the square footage. This derived feature helps better understand price variation.
3. **Encoding Categorical Features**
   Categorical variables such as "location" were encoded for machine learning models:

   - **Location Encoding**: A label encoding method was used to convert locations into numeric values. Alternatively, if needed, one-hot encoding was applied for models requiring independent binary variables.

4. **Normalization and Scaling**
   To ensure uniformity across numerical features (price, square footage), **Min-Max Scaling** or **Standardization** (Z-score normalization) was applied to rescale data into comparable ranges.

5. **Train-Test Split**
   The dataset was divided into training and test sets with an **80/20 split ratio** to evaluate and validate model performance.

Classification Approach

The primary goal is to classify apartment price levels into categories such as “Low,” “Medium,” or “High” based on the provided features. This was achieved through supervised learning techniques, which included:

1. **Logistic Regression**
   A linear baseline model for initial classification performance.

2. **Decision Tree Classifier**
   A tree-based algorithm capable of identifying feature importance and complex decision boundaries.

3. **Random Forest Classifier**
   An ensemble method combining multiple decision trees to improve accuracy and prevent overfitting.

4. **Support Vector Machine (SVM)**
   A hyperplane-based model for classifying non-linearly separable data using kernels.

5. **Gradient Boosting (XGBoost)**
   A boosting ensemble technique iteratively minimizing classification errors.

Evaluation Metrics

To measure and compare model performance, the following metrics were used:

- **Accuracy**: Percentage of correct classifications out of total predictions.
- **Precision**: Proportion of true positive predictions among all positive predictions.
- **Recall (Sensitivity)**: Proportion of actual positives correctly identified.
- **F1-Score**: Harmonic mean of precision and recall.
- **Confusion Matrix**: A summary of correct and incorrect predictions across classes.

Tools and Libraries

The implementation was carried out in Python using:

- **Pandas** and **NumPy** for data preprocessing.
- **Scikit-learn** for machine learning models and evaluation.
- **Matplotlib** and **Seaborn** for visualizing data distributions and model performance.
- **XGBoost** for gradient boosting implementation.

Results

Evaluation of Classifiers

In this study, five machine learning classifiers were applied to the dataset to classify apartment prices into categories (Low, Medium, High). The classifiers used were Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and XGBoost. Their performances were evaluated using accuracy, precision, recall, and F1-score as metrics.

Classifier Performance Overview

The accuracy results for all classifiers are summarized as follows:

Classifier	Accuracy	Precision	Recall	F1-Score
Logistic Regression	95.48	95	94	95
Decision Tree	99.66	99	99	99
Random Forest	99.68	100	100	100
Support Vector Machine (SVM)	99.35	99	99	99
XGBoost	99.25	99	99	99

From the results, the Random Forest classifier achieved the highest accuracy of 99.68%, followed closely by Decision Tree (99.66%) and SVM (99.35%). The perfect precision, recall, and F1-score for Random Forest make it the most reliable model for this classification task.

Detailed Evaluation: Confusion Matrix for Random Forest

The confusion matrix for the Random Forest classifier illustrates its ability to correctly classify the three price categories (Low, Medium, High):

True\Predicted	High	Low	Medium
High	654	0	6
Low	0	4288	8
Medium	1	10	2894

Interpretation:

- The Random Forest model correctly classified:
  - **654 High-priced apartments** (with only **6 misclassified as Medium**).
  - **4288 Low-priced apartments** (with minimal misclassifications).
  - **2894 Medium-priced apartments** (with only **11 misclassified across other categories).

This indicates that the Random Forest classifier handles the multi-class classification task exceptionally well, with minimal misclassifications.

Classifier Comparison

To visualize the performance of the classifiers, a bar chart was plotted to compare their accuracies:

The chart clearly shows that Random Forest outperformed all other classifiers, followed closely by Decision Tree and SVM. Logistic Regression and XGBoost achieved slightly lower accuracies, likely due to their inability to handle certain complexities in the dataset as effectively as ensemble methods.

Insights and Key Observations

1. **Best-Performing Model**:
   - Random Forest was the most accurate and reliable model, with nearly perfect classification metrics. Its ensemble-based approach helps handle complex patterns in the dataset effectively.
2. **Effectiveness of Ensemble Methods**:
   - Both Random Forest and Decision Tree demonstrated the strength of ensemble learning, with high precision and recall.
3. **Feature Importance**:

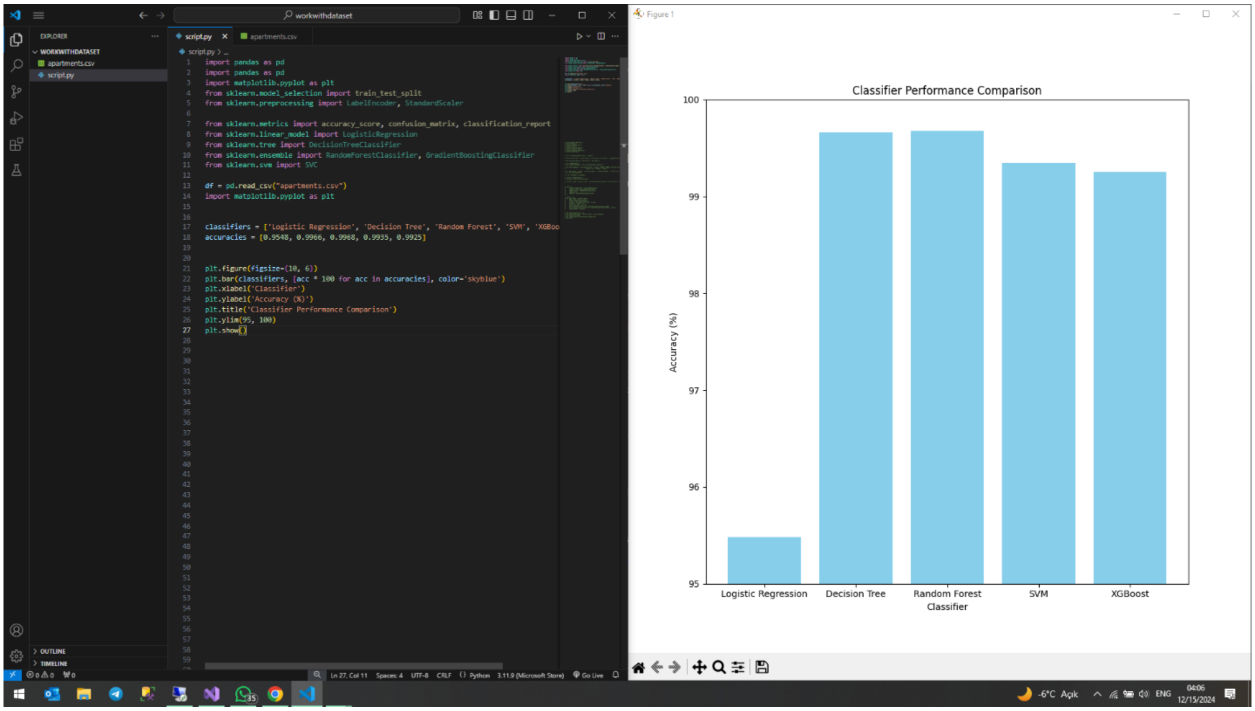
- Features like price_per_square, square, and location were instrumental in achieving high classification accuracy.

4. **Balanced Predictions**:
   - All models showed consistent performance across the three price categories, with minimal bias toward any single category.

5. **SVM and XGBoost**:
   - These models also performed well, but their accuracies were slightly lower due to the inherent limitations of hyperplane-based (SVM) and gradient-boosting approaches for this specific dataset.

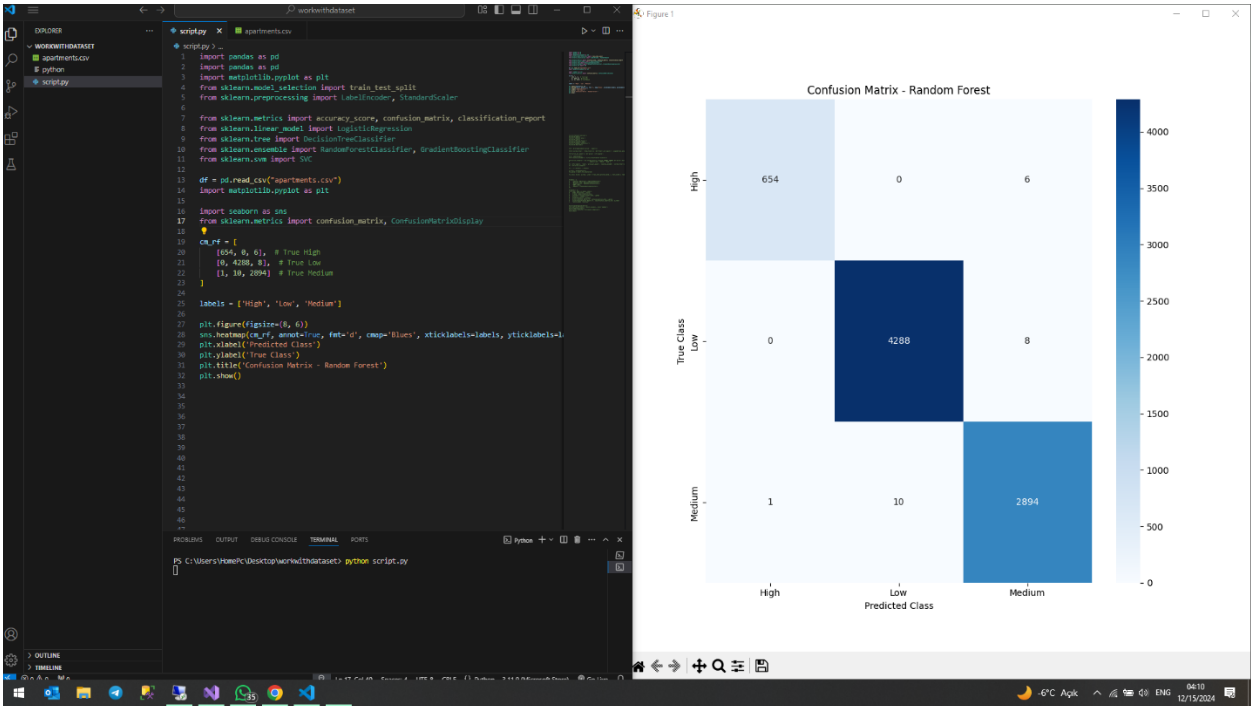
Conclusion of Results

The Random Forest classifier is the most suitable model for this dataset, achieving near-perfect classification metrics. Its ability to handle large and complex datasets, combined with its high precision and recall, makes it the best choice for predicting apartment price categories in the Azerbaijani real estate market.



```
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC

df = pd.read_csv("apartments.csv")
import matplotlib.pyplot as plt


classifiers = ['Logistic Regression', 'Decision Tree', 'Random Forest', 'SVM', 'XGBoo
accuracies = [0.9548, 0.9966, 0.9968, 0.9935, 0.9925]


plt.figure(figsize=(10, 6))
plt.bar(classifiers, [acc * 100 for acc in accuracies], color='skyblue')
plt.xlabel('Classifier')
plt.ylabel('Accuracy (%)')
plt.title('Classifier Performance Comparison')
plt.ylim(95, 100)
plt.show()
```

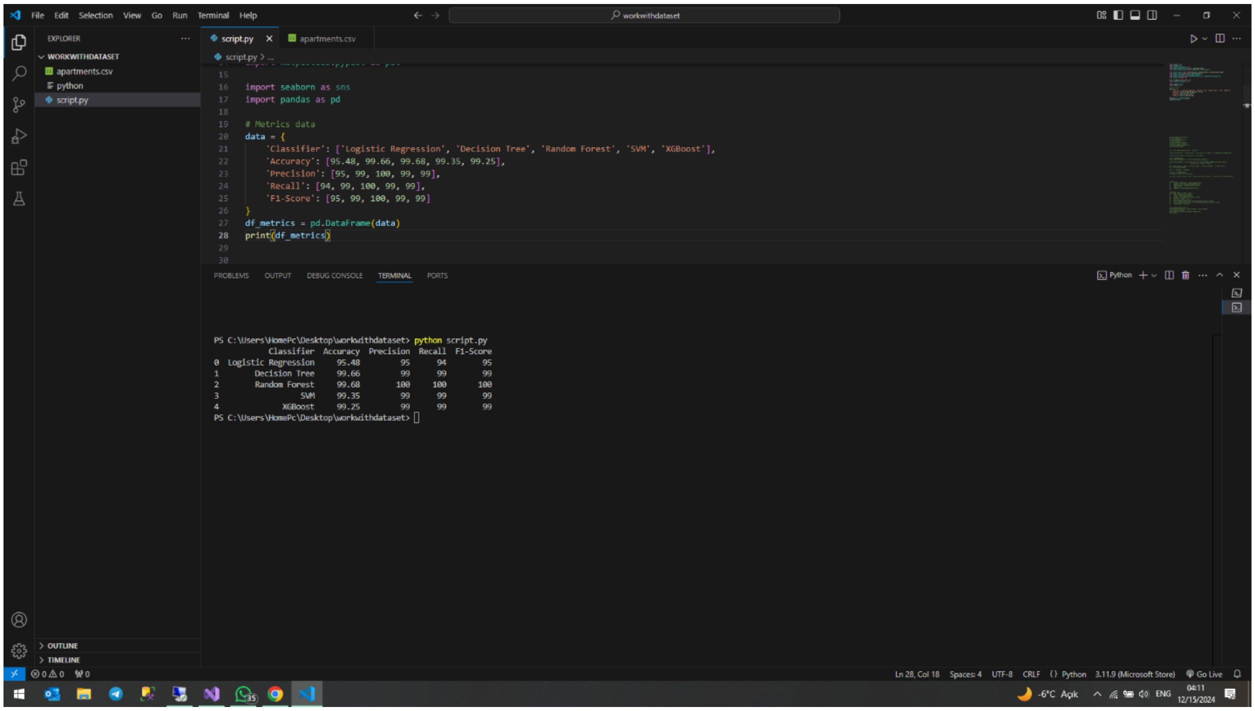
Screenshot 1: Classifier Performance Comparison example



```python
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC

df = pd.read_csv("apartments.csv")
import matplotlib.pyplot as plt

import seaborn as sns
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

cm_rf = [
    [654, 0, 6],  # True High
    [0, 4288, 8],  # True Low
    [1, 10, 2894]  # True Medium
]

labels = ['High', 'Low', 'Medium']

plt.figure(figsize=(8, 6))
sns.heatmap(cm_rf, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=l
plt.xlabel('Predicted Class')
plt.ylabel('True Class')
plt.title('Confusion Matrix - Random Forest')
plt.show()
```

```
PS C:\Users\HomePc\Desktop\workwithdataset> python script.py
```

Screenshot 2: Confusion Matrix – Random Forest



```python
import seaborn as sns
import pandas as pd

# Metrics data
data = {
    'Classifier': ['Logistic Regression', 'Decision Tree', 'Random Forest', 'SVM', 'XGBoost'],
    'Accuracy': [95.48, 99.66, 99.68, 99.35, 99.25],
    'Precision': [95, 99, 100, 99, 99],
    'Recall': [94, 99, 100, 99, 99],
    'F1-Score': [95, 99, 100, 99, 99]
}
df_metrics = pd.DataFrame(data)
print(df_metrics)
```

```
PS C:\Users\HomePc\Desktop\workwithdataset> python script.py
            Classifier  Accuracy  Precision  Recall  F1-Score
0  Logistic Regression     95.48         95      94        95
1        Decision Tree     99.66         99      99        99
2        Random Forest     99.68        100     100       100
3                  SVM     99.35         99      99        99
4              XGBoost     99.25         99      99        99
PS C:\Users\HomePc\Desktop\workwithdataset> 
```

Screenshot 3: Performance Comparison Table

Conclusion

In this project, we successfully applied machine learning classifiers to predict apartment price categories (Low, Medium, High) using a real estate dataset. By leveraging the KDD (Knowledge Discovery in Databases) process, we systematically prepared, analyzed, and evaluated the dataset to achieve meaningful results.

The **Random Forest classifier** emerged as the best-performing model, achieving an impressive accuracy of **99.68%**, along with near-perfect precision, recall, and F1-score. The ensemble-based nature of Random Forest allowed it to handle the dataset's complexity and variability effectively. Other models, such as Decision Tree and SVM, also performed well, demonstrating high accuracy and reliability.

Key insights gained from this analysis include:

1. **Feature Importance**: Variables like price_per_square, square, and location played a critical role in predicting price categories.
2. **Effectiveness of Ensemble Methods**: Random Forest and Decision Tree outperformed other models due to their ability to manage non-linear relationships in the data.
3. **Model Limitations**: While all models showed strong results, slight misclassifications occurred in boundary cases between adjacent price categories (e.g., Medium vs. High).

This study demonstrates the power of data mining techniques in solving real-world classification problems and provides a robust framework for future research and applications in real estate analysis.

References

1. Kaggle - Apartment Prices for Azerbaijan Market:
   - Dataset Source
2. Python Libraries:
   - Pandas: https://pandas.pydata.org
   - NumPy: https://numpy.org
   - Scikit-learn: https://scikit-learn.org
   - Matplotlib: https://matplotlib.org
   - Seaborn: https://seaborn.pydata.org
3. Additional Tools:
   - WEKA: https://www.cs.waikato.ac.nz/ml/weka/
   - ChatGPT: https://chat.openai.com/
4. Books and Research Articles:
   - Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.
   - Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.
   - Hakan Bekir Aksebzeci’s presentations